

Índice de contenido

I.INTRODUCCIÓN.....	1
1.1.ÁMBITO DE LA APLICACIÓN.....	1
1.1.1.Telemedicina.....	2
1.1.1.1.Nuevo planteamiento.....	2
1.1.1.2.Cambio de Paradigma en las relaciones médico-paciente.....	3
1.1.1.3.Campos de aplicación.....	5
1.1.1.4.Algunas aplicaciones de Telemedicina.....	7
1.1.2.Internet en Medicina.....	9
1.1.2.1.Problemas técnicos planteados.....	10
1.1.2.2.Principales campos de aplicación.....	11
1.2.PLANTEAMIENTO DEL PROBLEMA.....	12
1.2.1.La transmisión de ECG. Transmisión por Internet.....	14
1.2.2.Anotaciones.....	15
1.2.3.Estándar para la transmisión de ECG.....	18
1.3.PLANTEAMIENTO DEL TRABAJO.....	19
II.EVALUACIÓN DE LOS ESTÁNDARES ACTUALES.....	20
2.1.ESTÁNDARES EN INFORMÁTICA MÉDICA.....	20
2.2.APLICACIÓN DE UN ESTÁNDAR A LA TRANSMISIÓN DE REGISTROS BIOMÉDICOS.	22
2.3.BREVE DESCRIPCIÓN DE LOS ESTÁNDARES CONSIDERADOS.....	24
2.3.1.HL7.....	24
2.3.2.P1073.....	27
2.3.3.MS-HUG.....	31
2.3.4.SCP-ECG.....	31
2.4.CONCLUSIONES DEL ANÁLISIS.....	33
2.5.ESTÁNDAR ESCOGIDO: SCP-ECG.....	34
2.5.1.Objetivo principal de la norma SCP-ECG e Historia reciente.....	34
2.5.2.Necesidad de especificar un mecanismo de compresión.....	37
2.5.2.1.Compresión de datos en SCP-ECG.....	39
2.5.3.Formato de datos que se utiliza.....	41
2.5.4.Protocolo de Transmisión.....	43
2.6.CONCLUSIONES Y PROPUESTA DE APLICACIÓN.....	44
III.CICLO DE VIDA SOFTWARE.....	47
3.1.ANÁLISIS.....	48

3.1.1.Especificaciones y Requerimientos del Sistema.....	48
3.1.2.Arquitectura del Sistema.....	50
3.1.3.Arquitecturas Cliente-Servidor en Internet.....	55
3.1.3.1.Conexión básica: Sockets.....	55
3.1.3.2.Desarrollo de aplicaciones utilizando Objetos Distribuidos.....	56
3.1.3.2.1.RMI.....	58
3.1.3.2.2.CORBA.....	58
3.1.3.2.3.DCOM.....	59
3.2.DISEÑO.....	60
3.2.1.Objetivos de Diseño. Arquitectura empleada.....	60
3.2.2.Diseño de Clases.....	62
3.3.EVALUACIÓN Y MODIFICACIÓN.....	65
3.3.1.Ejemplo de funcionamiento.....	67
3.3.2.Nuevos estándares y métodos de consulta.....	69
IV.CONCLUSIONES Y RECOMENDACIONES.....	72
4.1.CONCLUSIONES.....	72
4.2.FUTURAS VERSIONES	74
V.REFERENCIAS BIBLIOGRÁFICAS.....	76
VI.LISTA DE ABREVIATURAS.....	81
VII.APÉNDICE A.....	83
VII.1.MECANISMOS DE COMPRESIÓN Y DESCOMPRESIÓN.....	83
A.1.1.Especificaciones para la compresión de datos.....	83
A.1.2.Presentación del esquema de compresión-decompresión empleado.....	84
A.1.3.Ejemplos de compresión y explicación del esquema.....	86
A.1.3.1.1.1.Truncamiento.....	87
A.1.3.1.1.2.Señal residuo.....	88
A.1.3.1.1.3.Filtraje y Diezmado.....	88
A.1.3.1.1.4.Algoritmos en Diferencias.....	89
A.1.3.1.1.5.Codificación Huffman.....	90
VII.2.FORMATO DE LOS DATOS	94
A.1.4.Consideraciones Generales.....	94
A.1.5.Datos Demográficos, de Diagnóstico y de Adquisición (No Data).....	98
A.1.6.Anotaciones.....	98
A.1.6.1.Sintaxis y composición del código.....	98

A.1.6.2.Mnemónicos y modificadores empleados.....	99
A.1.6.3.Ejemplos de anotación.....	100

RESUMEN

En el presente trabajo **se ha implementado una aplicación cliente-servidor** que transmite un archivo ECG a través de la Red Internet. La transmisión de este archivo es independiente de la plataforma, es decir, está adaptada al protocolo de transmisión por Internet (TCP/IP). Se han estudiado también diferentes **estándares internacionales de informática médica** y su aplicación para la transmisión de ECG. Después de analizarlos, se escogió uno (SCP-ECG) para su aplicación directa en este trabajo.

Para el diseño del sistema cliente-servidor se ha seguido la Metodología de Diseño Orientado a Objetos (DOO) que permite una descripción del problema en objetos y clases. Estos objetos y clases se han implementado en el **lenguaje de programación Java** y el diseño permite una fácil ampliación a otros estándares.

Para la evaluación se ha integrado la aplicación cliente-servidor con **un visualizador de ECG**. Este visualizador utiliza la aplicación cliente-servidor para la **consulta remota de ECG y para el envío de anotaciones** a través de Internet. Se ha probado la transmisión según el estándar SCP-ECG y de archivos con formato binario. Los resultados sobre la transmisión indican que se puede transmitir ECG a través de Internet utilizando diferentes estándares sobre una misma estructura cliente-servidor. Además se ha evaluado un lector de archivos SCP-ECG y binarios.

Esta aplicación hace posible la interpretación y evaluación de ECG para fines de investigación y clínicos, a través de Internet y se suma a las aplicaciones de Telemedicina.

I. INTRODUCCIÓN

Durante las últimas tres décadas, se ha invertido un gran esfuerzo de investigación en el desarrollo de sistemas para la adquisición y procesamiento automatizado del electrocardiograma (ECG). En esta área se han propuesto distintos métodos para asistir al diagnóstico médico.

Recientemente una nueva disciplina, denominada "telemedicina" ha surgido para facilitar la aplicación de métodos y herramientas médicas entre usuarios ubicados en diferentes localidades. Una de las posibilidades de la Telemedicina es la transmisión de ECG para su estudio a distancia. Aún cuando existen diversas soluciones a este problema, todavía no se ha definido un protocolo de transmisión común a todos los sistemas (de adquisición, de bases de datos, formato de datos, etc.) que forman parte del sistema de transmisión-recepción.

Uno de los canales de comunicación más empleado en la actualidad es la red Internet, por lo que muchas de las aplicaciones de Telemedicina están utilizando esta vía para enviar y recibir información.

1.1. ÁMBITO DE LA APLICACIÓN.

Este trabajo está situado en un marco de aplicación de las nuevas tecnologías a la salud. En concreto se está planteando una aplicación de Telemedicina basada en Internet. En este apartado se introducen brevemente los conceptos básicos que definen el ámbito de la aplicación a implementar.

1.1.1. TELEMEDICINA

Hablar de Telemedicina, y sobre todo en la actualidad es complicado ya que es una especialidad en constante cambio y evolución. No se pretende de ninguna manera ofrecer una visión profunda de este tema sino sólo una explicación general de los cambios que se proponen y de las aplicaciones generales. El objetivo es ubicar este trabajo dentro de los planteamientos generales de la Telemedicina.

1.1.1.1. Nuevo planteamiento

Existen varias definiciones de telemedicina, pero se podría decir que la Telemedicina es la aplicación de métodos y herramientas médicas a distancia. Su objetivo es facilitar las principales tareas de la Medicina, por ejemplo diagnóstico, prescripción, consulta, investigación..., entre personas o instituciones que no están en el mismo entorno físico. Las disciplinas directamente involucradas son la Medicina y las Telecomunicaciones.

En la literatura se pueden encontrar muchos trabajos de telemedicina de los que se pueden citar: teleeducación (clases con profesores y alumnos en diferentes hospitales), telecirugía (entrenamientos de cirugías en lugares Web remotos, por ejemplo), telecooperación (se plantea un caso clínico que es debatido por especialistas en diferentes lugares), telediagnóstico (diagnóstico a individuos o comunidades aisladas), telemonitoreo (seguimiento de la evolución de un paciente en su residencia familiar) [1] [2] [3] [4] [5], etc. Existen muchas aplicaciones y quedan también muchas que desarrollar e implementar.

Este trabajo se sitúa entre las aplicaciones de telediagnóstico y telecooperación. Los problemas asociados a la transmisión y anotación de registros ECG remotos, todavía no han sido resueltos dentro de la telemedicina.

1.1.1.2. Cambio de Paradigma en las relaciones médico-paciente

Otra materia importante a tratar en una aplicación de Telemedicina es la nueva relación (virtual) médico-paciente. Un estudio sobre la evaluación de un proyecto en una zona rural en North Carolina (USA) (teleconsulta por video) [6] da algunas impresiones de pacientes y personal clínico sobre una aplicación de teleconsulta:

- Ahorra tiempo y costo del desplazamiento
- Se acostumbraron fácilmente a la consulta *virtual* (pacientes y personal médico)
- Los pacientes podían “llevarse” la consulta (en video) a su casa
- Disponibilidad total del doctor mientras duraba la teleconsulta
- Echan de menos el contacto físico pero se acostumbran a esta relación con el paciente

Según este estudio, se podría concluir que se puede mejorar la comunicación entre médico y paciente porque existe mayor concentración y disponibilidad en la consulta, aunque no exista un contacto físico directo.

La nueva relación que se crea es virtual, es decir, no existe un contacto directo entre el médico y el paciente. En la figura 1.1 se puede observar como la relación virtual (línea punteada) presta una serie de servicios similar a una consulta real. La comunicación entre las computadoras de la figura es a través de Internet pero se podrían emplear otros canales de comunicación.

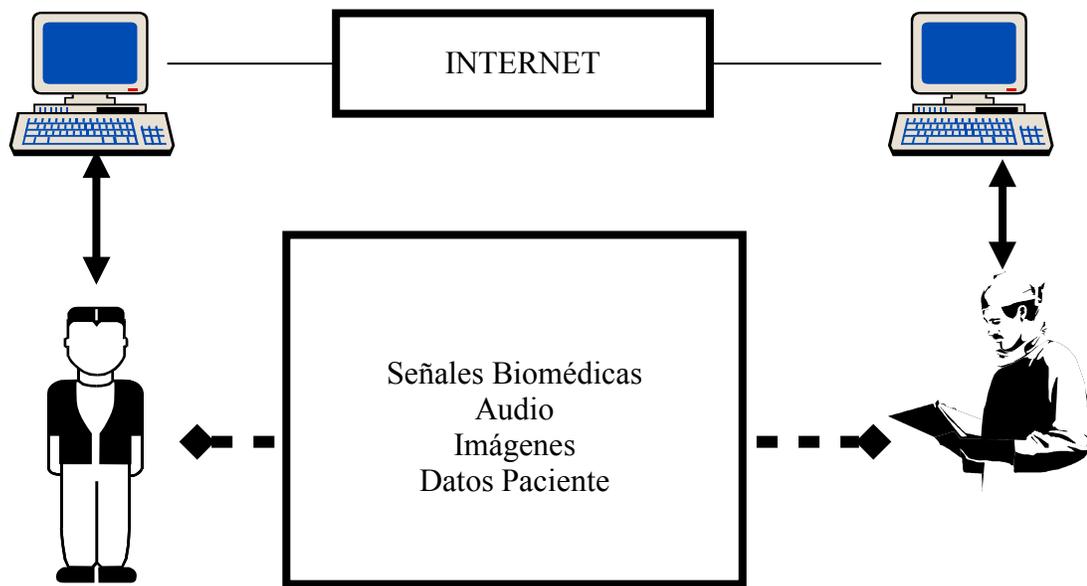


Figura 1. Utilización de un canal de comunicaciones (p.ej. Internet) para crear una relación virtual entre médico y paciente

Por lo tanto, tanto el paciente como el médico se tendrán que adaptar a la nueva relación que se establece y a las nuevas posibilidades [7]. Existen muchos inconvenientes para que la relación virtual modele la relación real. Lo importante es que, de cualquier forma, se producirá un cambio en la forma de entender y pensar las relaciones en el sector salud. Esto es lo que se denomina un cambio de paradigma.

Además, no se puede olvidar que las razones para implementar una aplicación de telemedicina no son exclusivamente tecnológicas, sino que se tiene en cuenta el factor humano y de mercadeo. Algunas preguntas que deben hacerse antes de diseñar una aplicación de Telemedicina serían:

- ¿Cuál es el objetivo de implementar la aplicación?
- ¿Quién la va a utilizar, y en qué circunstancias?
- ¿Qué mejoras ofrecerá, para el médico y el paciente?
- Es viable y rentable la aplicación.

Con todas estas consideraciones en mente, se verán los campos principales de aplicación de la Telemedicina.

1.1.1.3. Campos de aplicación

Un estudio del año 95 [8] clasifica las aplicaciones de la Telemedicina en las siguientes subespecialidades (se muestra también el % de proyectos en cada una):

Especialidad	%
Teleradiología	18.2
Telepatología/Oncología	13.4
Telecardiología	10.9
Telecirugía Endo/Laparoscópica	6
Teledermatología	6
Teleurología	4.8
Aplicaciones Académicas	9.7
Teleinformática Médica	4.8

Tabla 1. 1 Especialidades en Telemedicina

Estos proyectos totalizan más del 70% de las aplicaciones. En cada especialidad hay una disciplina clínica asociada. La designación de la especialidad “Teleinformática” no es muy acertada porque en realidad todas las aplicaciones mezclan informática y telecomunicaciones (telemática) y medicina, pero los autores engloban aquí aplicaciones puramente telemáticas. Lo importante aquí es ver el amplio abanico de proyectos en telemedicina [9]. En Venezuela se han propuesto proyectos en las áreas de telecardiología, telepatología y teleradiología, cuya descripción se puede encontrar en [10].

Las fuentes de información que se manejan generalmente en Telemedicina son[9]:

- Texto via teclado de computadora
- Video Analógico
- Documentos en papel
- Imagen digital desde cualquier dispositivo
- Información en vivo, videoconferencia (audio y video).

También es importante añadir la información que viene de señales biomédicas (ECG, presión arterial, por ejemplo). Esta información está adquirida del paciente a través de dispositivos especializados. Independientemente de la fuente de los datos, los procesos principales en Telemedicina son la adquisición, la transmisión y la visualización de los datos.

Entre las ventajas que se plantean para las aplicaciones de Telemedicina estarían:

- Ampliar la cobertura de los servicios de salud.
- Facilitar la cooperación entre médicos y especialistas
- Fomentar la creación de bases de datos y la cooperación entre centros de investigación
- Brindar apoyo a servicios especiales (emergencias, conflictos bélicos, desastres naturales...)
- Hacer un seguimiento remoto de pacientes (p.ej. pre-postoperatorio)
- Proveer acceso para una mejor formación y educación de especialistas (cooperación entre centros institucionales)
- Posibilitar la Cirugía Remota
- Provocar un cambio en la relación costo-beneficio (descongestión hospitales, fácil acceso a especialistas, disminuye costos y tiempo de visitas médicas, etc...).

A pesar de que estas ventajas se citan con frecuencia, vale la pena preguntarse porqué no es común la utilización de estas aplicaciones. En este sentido se puede ver que todavía existen numerosos inconvenientes para la puesta en marcha de aplicaciones de telemedicina en los servicios de salud. Estos inconvenientes se podrían categorizar en:

- Tecnológicos
- Clínicos

Tecnológicamente hay muchos temas que resolver, tanto a nivel de infraestructura técnica como de software y aplicaciones. Obviamente esto está unido a los costos de la aplicación. Algunos inconvenientes clínicos vienen de la dificultad de “simular” técnicamente la relación necesaria entre médico y paciente [7], por ejemplo, la calidad de las imágenes clínicas, la fiabilidad de los equipos, el contacto humano, la diversidad de pruebas a tener en cuenta para obtener un diagnóstico, etc...

A continuación se presentarán algunos ejemplos reales de aplicación. Se han escogido algunos ejemplos representativos que pueden dar una idea de la amplia gama de proyectos que se están llevando a cabo en Telemedicina.

1.1.1.4. Algunas aplicaciones de Telemedicina

Un primer ejemplo [11] consiste en una unidad móvil que utiliza un satélite para la transmisión de vídeo de alta resolución e imágenes médicas estáticas (p.ej. radiografías). Es un proyecto de alto costo en el que están implicados la NASA, ARPA, GE Medical Systems, JPL (*Jet Propulsion Laboratory*) y la universidad de Washington.

El proyecto trata de implementar un dispositivo que utiliza comunicación por satélite (ACTS, *Advanced Communications Technology Satellite*). Este dispositivo es un terminal con una antena pequeña que transmite video en tiempo real, desde este terminal móvil al departamento de tecnología de la Universidad de Washington. En la figura siguiente (figura 1.2) se muestra un pequeño diagrama con el funcionamiento de este dispositivo. Muchas aplicaciones de telemedicina utilizan comunicaciones por satélite [12] [13].

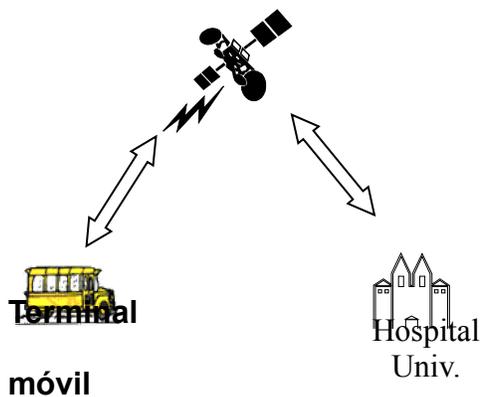


Figura 1. 2. Telemedicina utilizando comunicaciones por Satélite

Otra infraestructura utilizada son las líneas telefónicas o de red (por ejemplo, fibra, coaxial...). En este caso [14] se trata de proveer una aplicación de teleconsulta donde se pueda enviar voz, imágenes (radiografía) y datos. El objetivo es ampliar el acceso a los servicios de salud de la población rural del estado de Oklahoma (USA). Este servicio funciona desde 1996 entre el Hospital Bautista de Oklahoma y el Comanche County Memorial Hospital. Están conectados con una línea T1 (1 Mbps de velocidad de transmisión). La diferencia con la aplicación anterior es que el consultorio virtual (donde está el paciente) y el real (especialista) están fijos. Y que la transmisión se hace por un canal físico (línea dedicada o telefónica). Este es uno de los tipos de comunicación más comunes.

A través de Internet existen muchas aplicaciones de interés a nivel mundial. Dos ejemplos interesantes son el acceso a imágenes del cuerpo humano (teleeducación e investigación) y el paciente interactivo. La filosofía de estas dos aplicaciones es totalmente diferente a las anteriores y por eso se muestran aquí. En este caso son aplicaciones de divulgación e investigación, están situadas entre los campos de teleeducación, teleconsulta y formación continuada (remota).

El proyecto *Human Visible* [15] comenzó en 1991 y es una base de datos de imágenes de un cuerpo de una persona (de ambos sexos). Las instituciones implicadas son la Universidad de Colorado y la Biblioteca Nacional de Medicina, ambas en USA. La consulta de la base de datos se hace a través de Internet, a través de un navegador convencional. Son 15 Gb de información almacenada (persona del sexo masculino) y

la información se ha procesado con estaciones de trabajo y un supercomputador. Se proveen fotografías de cualquier parte del cuerpo y en las tres dimensiones.

Otro proyecto interesante es el paciente interactivo [16]. Se accede también a través de un browser convencional. Es un paciente, con su historia médica, datos demográficos, radiografías, etc. El objetivo es que un médico acceda a esta información para dar un diagnóstico. Este diagnóstico es corroborado automáticamente y se emite una evaluación sobre si el diagnóstico es correcto o no. Esta aplicación está totalmente orientada a médicos o personas especialistas. Está desarrollada en la Universidad de Virginia.

Se han mostrado cuatro ejemplos de aplicaciones, dos con una filosofía de la Telemedicina clásica y estas dos últimas como otra forma de utilizar las tecnologías de la información en el sector salud para investigación y enseñanza de profesionales.

Algunos servidores Web donde se podrá encontrar información general sobre aplicaciones y proyectos de Telemedicina son :[17] [18] [19] [20].

1.1.2. INTERNET EN MEDICINA

La integración de las telecomunicaciones y la telemedicina da lugar a las aplicaciones que se han mostrado anteriormente. En la actualidad Internet, que se podría sintetizar como una integración de las telecomunicaciones y la informática, tiene un papel fundamental en la aplicación de las tecnologías de la información en la vida cotidiana, y por lo tanto en el sector salud.

Un ordenador personal, bajo cualquier plataforma, puede acceder de una manera relativamente sencilla, a través de Internet, a una cantidad de información enorme. Esta infraestructura ofrece una nueva posibilidad de comunicación personal y grupal y está cambiando el modo de trabajo de diferentes sectores profesionales.

Por ello, el sector salud se está viendo influenciado por el impacto de la aplicación de Internet en medicina. Por una parte, la colaboración entre profesionales [21] es fundamental tanto para el trabajo científico como para la práctica clínica, y por otra

parte la relación entre paciente y médico se puede ver mejorada por la posibilidad de establecer una relación virtual a través de Internet.

Por definición, una aplicación de Internet en medicina entra dentro de lo que se ha denominado Telemedicina. De hecho, ya se han presentado algunas aplicaciones que utilizan Internet al tratar de enmarcar el trabajo en el ámbito de la Telemedicina. En este apartado se especificarán algunos problemas de tipo técnico que se plantean y los principales campos de aplicación de Internet en Medicina.

1.1.2.1. Problemas técnicos planteados

La utilización de Internet como esquema de comunicaciones plantea una serie de requisitos para esta aplicación:

1. Independencia de la plataforma
2. Uso de las herramientas comunes de Navegación
3. Utilización del protocolo TCP/IP para la transmisión de datos
4. Acceso directo, o a través de un proveedor comercial a la red Internet
5. Ancho de Banda disponible, según el tipo de aplicación

En la actualidad muchos de estos problemas ya han sido resueltos. Las herramientas comunes de navegación (así se denomina a la búsqueda y recepción de información) son los navegadores, o *browsers*. También se tienen otros servicios como el acceso al correo electrónico, servicios de chat, de transferencia de ficheros (ftp), de grupos de noticias (news), etc... Para una mayor profundización en estos temas se pueden consultar algunos manuales en línea [22].

En una aplicación típica, un servidor HTTP (HyperText Transfer Protocol) provee información por medio de páginas HTML (HyperText Markup Language) a un navegador. Dentro de esta información HTML puede haber llamadas a applets que

son programas que se ejecutan en la máquina que corre el navegador . De esta manera, se pueden ejecutar aplicaciones insertadas en páginas HTML.

Se pueden plantear diferentes arquitecturas para cada aplicación, como se expone en el capítulo III, pero se debe mantener el funcionamiento dentro del navegador. Los applets están escritos en Java que es un lenguaje de programación orientado a Objetos e independiente de la plataforma. Este lenguaje, por sus características, ha tenido un auge muy grande en Internet. Java soporta el protocolo TCP/IP dentro de las clases especializadas en la conexión de red. Mas información se puede encontrar en [23].

Por último, es importante señalar que el acceso a Internet viene dado por un Proveedor de Servicios de Internet (ISP, *Internet Service Provider*) que ofrece el tipo de conexión adecuado para la aplicación. Aunque todavía no existe un acceso masivo, éste se incrementa en una alta proporción cada año [24]. El acceso puede ser a través de un enlace dedicado o a través de la línea telefónica. En cualquier caso el ancho de banda es limitado y se debe tener en cuenta como requisito en la aplicación. Una manera eficiente de aprovechar el ancho de banda es mediante los mecanismos de compresión de datos.

1.1.2.2. Principales campos de aplicación

Internet [25] es una infraestructura que le proporciona al sector salud la posibilidad de implementar muchas de las aplicaciones de Telemedicina que ya se han presentado. Pero no sólo la interconexión de un centro de salud, por ejemplo, con el exterior, sino la interconexión de los diferentes departamentos dentro de un mismo hospital. Esto es lo que se denomina Intranet. La diferencia entre Internet e Intranet está en la globalización de la conexión. Algunos hospitales tienen su propia Intranet (con acceso a Internet) que resuelve muchos de los problemas de comunicación interna y con el exterior.

Las aplicaciones más generales de Internet/Intranet en Medicina son:

- Acceso a información sobre salud (www.homecare.com)
- Cooperación y colaboración clínica[21]

- Administración (Ej: Administración de información de ECGs. Sistema Muse de Marquette Inc.)
- Investigación
- Asistencia y cuidado de pacientes

En [26] [27] están disponibles ponencias sobre las tendencias actuales en las aplicaciones de Internet en el sector salud. Este trabajo se enmarca en las dos últimas especialidades ya que son las líneas de mayor actualidad en el campo de la investigación y académico. En el apartado siguiente se plantea el problema que se pretende resolver en este trabajo.

1.2. PLANTEAMIENTO DEL PROBLEMA

En la figura 1.3 se pueden observar los factores que intervienen dentro del problema que se pretende resolver. El diagrama muestra la problemática de la conexión “virtual” de un paciente con su médico especialista, o con un hospital o centro de salud. Además, se supone que este paciente tiene una computadora, un dispositivo de adquisición de ECG y acceso a Internet.

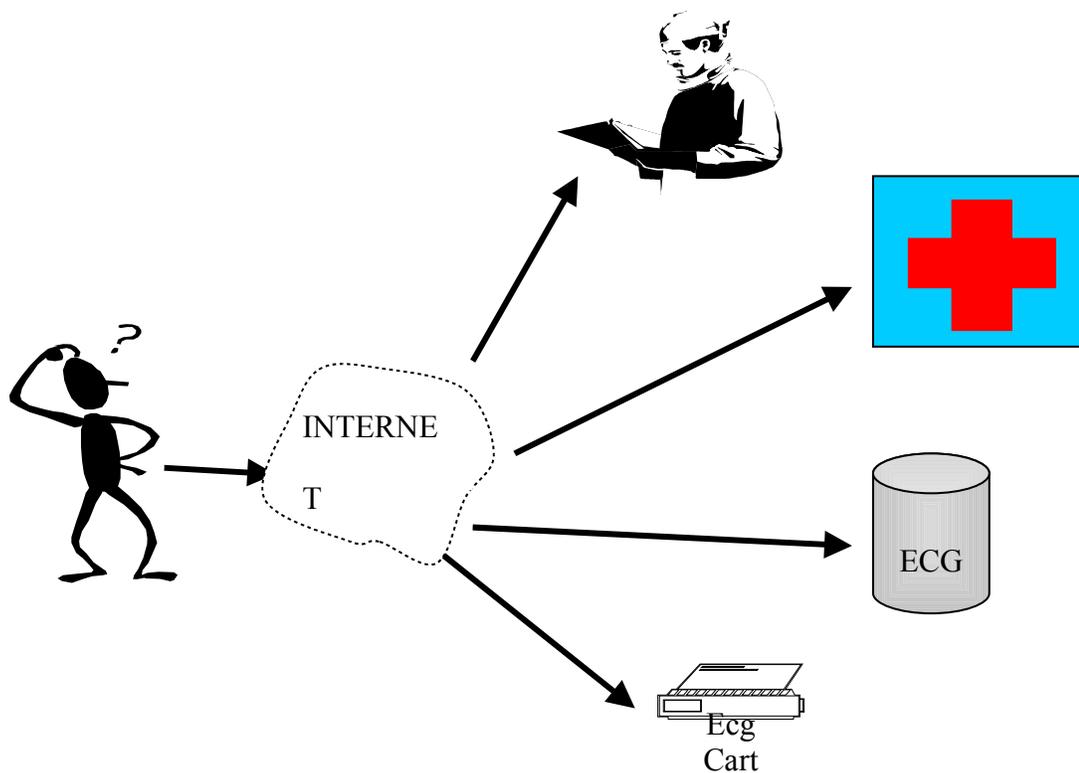


Figura 1. 3 Factores involucrados en la Transmisión de ECG por Internet

La transmisión de ECG ha sido resuelta de varias formas y existen aplicaciones para la mayoría de los canales de comunicación: radio, red local, vía telefónica, canales móviles, etc. La transmisión a través de Internet también se ha planteado ya [5] [28].

Sin embargo, hay que responder si está resuelto el problema de la transmisión de ECG a través de Internet. En primera instancia, la respuesta es “no del todo”. La razón principal es que no hay un estándar mundial para la transmisión de ECG, es decir, todas las soluciones son dependientes del fabricante o de una tecnología específica. En cuanto a Internet, además de no obedecer a un estándar internacional, ocurre que muchas de ellas son comerciales, por lo tanto no están disponibles académicamente.

1.2.1. LA TRANSMISIÓN DE ECG. TRANSMISIÓN POR INTERNET.

La transmisión de un registro ECG consiste en codificar las muestras del ECG y otra información añadida, por ejemplo las interpretaciones o datos demográficos (datos personales, del personal especialista que realiza la adquisición, etc.), y transmitirlos a través de un canal de comunicaciones. Para realizar la transmisión hace falta estar de acuerdo primero en el formato en el que se almacenan los datos y después en cómo se envían. Esto significa que se necesita un estándar para la transmisión-recepción del ECG, como se especificará en el apartado siguiente.

Además hay que adecuarse al canal de comunicaciones utilizado, es decir, no es lo mismo enviar un ECG por un canal de telefonía móvil que por satélite o por Internet. Cada plataforma tiene sus propias especificaciones para el transporte de los datos, por lo tanto, el estándar ha de tener en cuenta cada caso. En el caso de Internet se ha de utilizar el estándar TCP/IP (*Transmission Control Program/Internet Protocol*, Protocolo de control de la transmisión por Internet) para la transmisión de los datos.

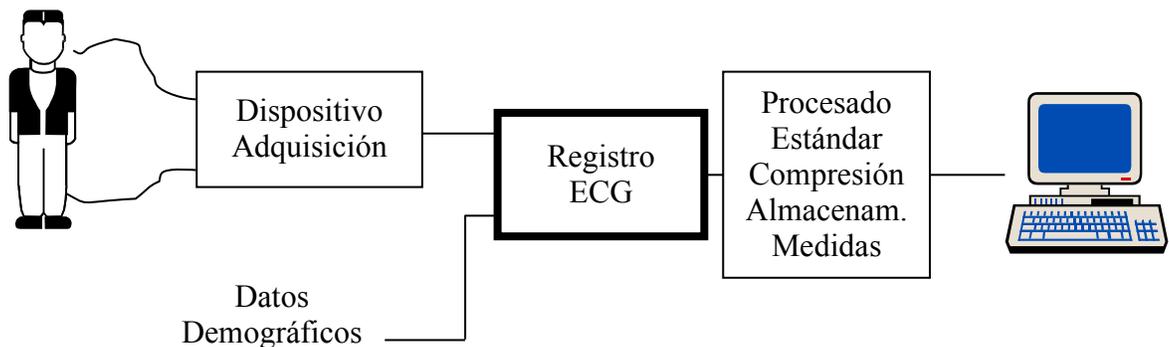


Figura 1. 4 Etapas de Adquisición y Procesado de la señal ECG.

Un esquema general de la aplicación tiene una etapa previa de adquisición y procesado de la señal hasta que se almacena en un computador (figura 1.4). En este trabajo no se analizará esta etapa, sino que se supondrá que se dispone el registro ECG. La computadora de la figura puede representar una base de datos, una

aplicación en tiempo real para la transmisión o cualquier aplicación final de la adquisición de un registro.

Por otra parte, cada canal tiene asociado un ancho de banda disponible y esto define la velocidad de transmisión de los datos. Las aplicaciones en tiempo real son las más exigentes en cuanto a velocidad de transmisión. Para aprovechar mejor esta característica se recurre a los mecanismos de compresión que tratan de eliminar la información redundante y/o la menos significativa. El estándar seleccionado es el SCP-ECG que implementa su propio esquema de compresión y descompresión de datos (Capítulo II).

Por último, para transmitir ECGs se debería tener en cuenta los tipos de registros posibles, como Holter, de reposo, de esfuerzo, de monitoreo, etc... Estos registros se diferencian, a efectos de transmisión, en la cantidad de datos a enviar.

1.2.2. ANOTACIONES

Anotar significa, básicamente, escribir una interpretación, según la codificación empleada, en el registro ECG de un paciente. Desde el punto de vista de la transmisión implica que debe haber una comunicación bidireccional. En algunas aplicaciones es suficiente con enviar los datos. Aquí se pretende que se interpreten remotamente y que se puedan recibir, es decir, que exista una comunicación entre el lugar de la adquisición o almacenamiento y el lugar de la visualización e interpretación.

En el campo de la investigación, se sabe que la interpretación de una señal ECG ayuda a establecer un diagnóstico eficaz de la cardiopatía de un paciente. Además, es un método no invasivo y es una herramienta eficaz para estudiar enfermedades que, si son detectadas a tiempo, se pueden subsanar. Por lo tanto realizar anotaciones sobre un registro ECG es una necesidad clínica importante. Las diferentes anotaciones permiten contrastar investigaciones, diagnósticos y otras apreciaciones cualitativas de la señal ECG. Por ello es un análisis de rutina en la mayoría de los hospitales y

consultorios. Ejemplos típicos de anotaciones serían la duración o amplitud de cada onda del ECG, o de problemas de variabilidad cardíaca, etc...

Estas anotaciones pueden hacer referencia a una parte de la señal o a todo el registro. Este análisis ha de estar en manos de médicos especializados (cardiólogos). En el campo de la investigación, como muchos de estos resultados no están todavía muy claros, se precisa tener bases de datos [29] con varios registros de ECGs que sean consultados por cardiólogos, de cara a hacer estudios exhaustivos de la interpretación realizada según el tipo de registro (que está asociado a un tipo de paciente). Las bases de datos de ECG se crean, en general, con el objeto de reunir información significativa que permita evaluar algún tipo de cardiopatía. La gran cantidad de información reunida hace que se necesiten algoritmos que permitan automatizar la anotación de registros de ECG. Los algoritmos se prueban sobre esta información y se corroboran con la información de un especialista para otorgarles validez.

Cada registro de la base de datos puede ser anotado por uno o varios especialistas (figura 1.5). Se almacenan registros de varios pacientes en la base de datos, y son consultados e interpretados por los investigadores. Estas anotaciones expresan una idea cualitativa sobre la patología o las principales características de ese registro. Con esta información se pueden realizar otras investigaciones, p.ej, clasificación de los registros, relación entre diversas patologías, aplicación de métodos de análisis a registros determinados, etc... También se debe decir que el fin último de un sistema automatizado sería la anotación automática de todos los registros. En cualquier caso, tanto en una anotación manual como automática, es necesario el proceso de anotación y para ello se debe transmitir el ECG.

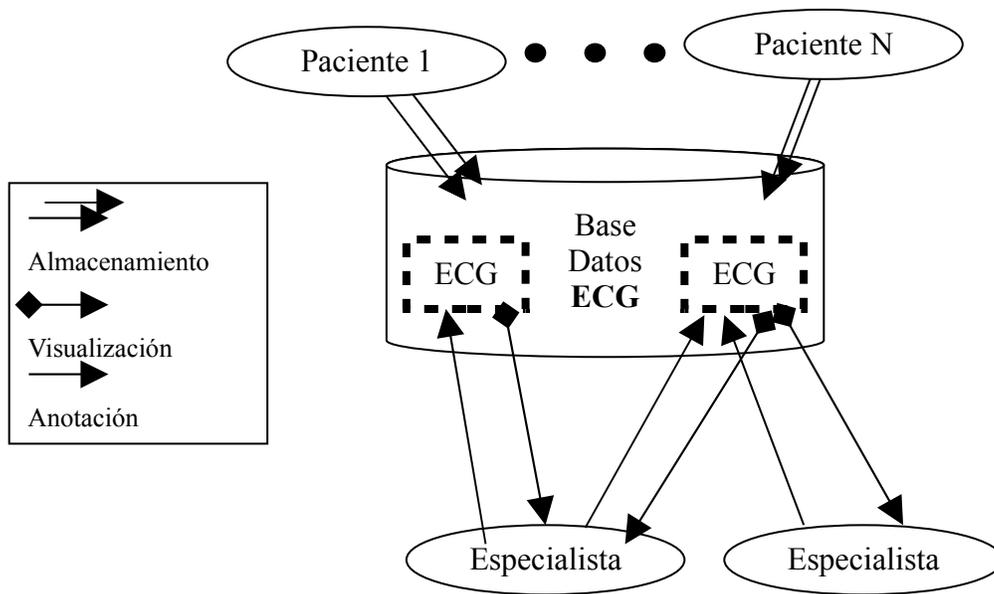


Figura 1. 5 Anotación de una Base de Datos de ECG

Una vez que se determina que existen bases de datos de registros ECG y que la interpretación de éstos es importante en la vida clínica de la mayoría de hospitales y consultorios de cardiología, faltaría determinar cómo se realizan estas anotaciones desde el punto de vista tecnológico.

La anotación de bases de datos remotas no es un problema sencillo de resolver ya que se necesita:

- Determinar el tipo de anotaciones a emplear (diccionario estándar) y el formato.
- Disponer de las bases de datos, en este caso particular, a través de Internet.
- Utilizar un estándar de transmisión de ECG aceptado internacionalmente.
- Gestionar el acceso a los registros ECG.
- Disponer de una herramienta que permita realizar la anotación remota

Las tres últimas características atañen directamente al ámbito de una aplicación a través de Internet. Y se pretenden dar propuestas para su solución.

Para terminar de presentar la importancia que, en la actualidad, está teniendo la anotación de cualquier tipo de información a través de ambientes distribuidos se

puede citar una arquitectura propuesta para la anotación de documentos que están disponibles en la Web. Esta arquitectura cliente-servidor denominada ComMentor [30] provee una plataforma para que se añada información (anotación) a documentos ya existentes. Es importante destacar que la arquitectura utilizada es similar a la que se planteará aquí. Algunas de las necesidades que allí se proponen están relacionadas con el protocolo para la anotación y la autenticación y selección de los usuarios.

1.2.3. ESTÁNDAR PARA LA TRANSMISIÓN DE ECG.

Aunque ya ha sido introducido en apartados anteriores y se discutirá con mayor profundidad en el capítulo siguiente, uno de los problemas planteados es la utilización de un estándar de transmisión de ECG. El análisis de los estándares en informática médica y su implementación necesitaría un trabajo completo, y en la actualidad es objeto de discusión entre diferentes grupos de investigación y también en el ámbito de la industria.

El problema fundamental es que es muy difícil encontrar una solución para todos los casos , de ahí que el tema de los estándares en electrocardiografía [31] [32] y en informática médica (el caso más general) [33] [34] [35] sea un tema todavía no resuelto.

Se han buscado aquellos estándares que son más representativos y se han analizado los factores más importantes para la utilización de cada estándar en la transmisión de ECG por Internet. Algunos estándares son HL7 [36], arquitectura Microsoft [37], sistema MUSE [28], P1073 (IEEE) [38] [39] , norma europea SCP-ECG [40] [41], etc.

1.3. PLANTEAMIENTO DEL TRABAJO.

La idea central de este trabajo está orientada a analizar los principales aspectos de la transmisión de ECG por Internet. Esto ha llevado a profundizar principalmente en la problemática de un estándar, tanto para el formato de los datos, como en el diseño de la aplicación. Una aplicación que se pueda ajustar y ampliar da la posibilidad de aunar y compartir esfuerzos para el desarrollo de un producto final.

El requisito de la transmisión en Internet también obliga a plantear una solución de acuerdo al estándar de facto que plantea Internet, tanto a nivel software como hardware.

Por lo tanto, después de haber definido el marco en el que se sitúa este trabajo, se van a presentar los objetivos que se persiguen dentro de este trabajo:

OBJETIVO GENERAL.

Estudio de los diferentes estándares de transmisión de datos de ECG e implementación de una aplicación cliente-servidor que utilice el más adecuado para la transmisión de ECG por Internet.

OBJETIVOS ESPECÍFICOS.

- Análisis de los diferentes estándares para la transmisión de datos de electrocardiografía y selección del más apropiado para nuestro trabajo.
- Elegir e implementar un protocolo de transmisión de datos de ECG
- Desarrollar el software con un lenguaje de programación orientado a objetos para reutilizar y seguir desarrollando la aplicación.
- Crear una librería de comunicaciones (paquete en Java 1.1) que pueda ser utilizada por diferentes aplicaciones que implementen el protocolo escogido.
- Evaluar la transmisión y anotación de una señal ECG

De este modo, también se han estructurado los capítulos del trabajo. El siguiente capítulo presenta una visión amplia de los estándares que se han evaluado para la aplicación en la transmisión de ECG por Internet y en particular el que se ha escogido. Después se presentan las etapas de desarrollo de la aplicación cliente-servidor en Java y, finalmente, las conclusiones y recomendaciones.

II. EVALUACIÓN DE LOS ESTÁNDARES ACTUALES

Este capítulo corresponde a uno de los objetivos del trabajo: la determinación de qué estándar es el más adecuado para la transmisión de ECG por Internet y el estudio de uno en particular. Para ello se presentarán de manera general algunos de los estándares más importantes en la actualidad y se plantearán las ventajas e inconvenientes de su utilización para la transmisión de ECG. También se escogerá uno para su implementación dentro de la aplicación cliente-servidor para la transmisión de ECG vía Internet, y se analizará en más detalle.

2.1. ESTÁNDARES EN INFORMÁTICA MÉDICA

Se entiende por informática médica al conjunto de estructuras y algoritmos necesarios para el intercambio y manejo de la información de registros biomédicos. El ámbito natural de este trabajo sería la informática médica aplicada a electrocardiografía computarizada.

Un estándar [33] es una colección de especificaciones que permiten a un sistema compartir información y estar de acuerdo en el manejo de ésta. Por sistema se puede entender un grupo de investigación, un grupo de dispositivos, etc... Esta información será cualquier tipo de registro biomédico, i.e., un ECG, una imagen de Rayos X,

medidas de presión arterial, drogas suministradas a un paciente, etc... En definitiva, serían un conjunto de reglas por las que un sistema A puede entender lo que le envía un sistema B. En realidad, se puede hablar de varias categorías de estándares en informática biomédica, que son los relativos a:

- Descripción de los mensajes para el intercambio de información.
- Descripción de conceptos médicos y el formato de datos.
- Descripción de reglas para tomar decisiones.
- Especificaciones de interoperatividad de dispositivos.

Cada uno de estos puntos refleja una parte del problema general. El formato de los datos ha de responder a unas especificaciones clínicas. Un estándar también define los mensajes necesarios para intercambiar los datos, y como objetivo final debería determinar cómo analizar esta información y extraer resultados. Este proceso es muy complejo y es tema de investigación en la actualidad.

Aunque existen diferentes razones y objetivos para la adopción de un estándar en informática médica, muchos de ellos derivan de su objetivo fundamental que es aumentar la productividad y calidad de los servicios de salud.

La mejora de la relación costo/beneficio obedece no sólo a intereses comerciales sino a los campos de la investigación y mejora de la calidad del servicio. Para lograr esto es necesaria la automatización de este proceso, que es lo que lleva a que los sistemas de información ocupen un lugar fundamental, hoy en día, en el sector salud. En la actualidad, los hospitales que tienen computadoras para la admisión, facturación, almacenamiento de imágenes, etc..., se encuentran, por ejemplo, con que cada fabricante tiene un formato de datos para cada tipo de registro biomédico. Algunos de los aspectos a solucionar [39] [42] para obtener una total compatibilidad entre dispositivos de adquisición y tratamiento de la información serían:

Soluciones Hardware:

- Sustitución de un dispositivo de un fabricante por otro sin especificaciones añadidas (*plug&play*)

- Configuración automática (Orientado al personal médico).
- Seguridad del Paciente

Soluciones Software:

- Intercambio de información (protocolos)
- Definición de datos (formato)
- Integración con las plataformas informáticas de la industria (fácil ampliación y actualización).

2.2. APLICACIÓN DE UN ESTÁNDAR A LA TRANSMISIÓN DE REGISTROS BIOMÉDICOS.

Aunque existe una gran variedad de soluciones y de objetivos, y por tanto diferentes estándares [35] [36], existen varias propuestas para la transmisión de registros biomédicos. Existen muchos estándares limitados a tipos de información concretos o a objetivos específicos, pero no existe una solución global al problema. La tendencia actual es la proposición de un estándar global y universal que permita manejar todos los tipos de información biomédica, y por ende, los diferentes estándares ya propuestos. La lista de siglas y acrónimos de estándares en informática médica es extensa y uno de los objetivos de este trabajo fue determinar los estándares con más proyección futura y ver la aplicación para la transmisión de datos de ECG por Internet.

En este trabajo se ha tratado esta cuestión por grupos de trabajo, más que estándares en sí. Los más importantes son HL7, P1073 y MS-HUG. En este trabajo se han estudiado los estándares y arquitecturas propuestas por estos grupos y también la norma europea SCP-ECG, que es un estándar específico de electrocardiografía.

Además existen entidades internacionales que se encargan del estudio y certificación de los diferentes estándares propuestos. Se pueden citar entre ellas, a ISO (*Internacional Standard Organization*), CEN (*Comisión Europea de Normativización*), UN/EDIFACT (*Organismo de la Naciones Unidas*), JAHIS (Organización Japonesa para los SI en el sector salud), etc... Por poner un ejemplo, decir que la ANSI (Instituto Nacional Americano de estándares) tenía en estudio en diciembre de 1996 a 15 organizaciones y sus propuestas en el ámbito de la informática médica [43].

En el caso particular de este trabajo, la aplicación de un estándar para la transmisión de registros biomédicos debe cumplir con los siguientes requisitos:

- Transmisión por Internet. Esto supone que la transmisión ha de ajustarse a los diferentes protocolos y arquitecturas de la red Internet (TCP/IP).
- Definir datos ECG. Aquí se engloban los diferentes tipos de ECG, como ECGs de reposo, ambulatorios, Holter, pruebas de esfuerzo, etc..., y sus características.

El ECG es uno de los registros biomédicos más comunes en informática médica, por ello, el estudio de la transmisión de ECG está dando una visión de lo que puede ocurrir con otros registros biomédicos. También se observará que existe un compromiso entre la definición general de un estándar y las especificaciones particulares para una especialidad, en este caso, electrocardiografía. De hecho, esta es una de las respuestas a porqué no existe un estándar en la transmisión de registros biomédicos.

Los primeros intentos de estándar para la transmisión de ECG son de la década de los 80 [31] promovidos por la *Veterans Administration* de USA y algunas compañías fabricantes de electrocardiógrafos. Estas investigaciones estuvieron centradas en los mecanismos de compresión y en el formato de ECGs. En ningún caso estaban pensados para otro tipo de registros biomédicos.

2.3. BREVE DESCRIPCIÓN DE LOS ESTÁNDARES CONSIDERADOS.

En este apartado se pretende dar una visión general de los estándares que se han considerado. HL7 y P1073 son dos estándares con filosofías diferentes, en el primer caso orientado al software e intercambio de información y en el segundo, a la conexión e interoperabilidad de equipos. El MS-HUG es el grupo de trabajo que propone el estudio de las aplicaciones del sector salud utilizando el entorno Microsoft. SCP-ECG es un estándar particular para la transmisión de ECG, pero que no es válido para otro tipo de registros biomédicos. Por lo tanto, se presentarán los estándares HL7, P1073, MS-HEG y SCP-ECG.

2.3.1. HL7.

El grupo de trabajo de HL7 (High Level 7) está trabajando en las especificaciones de las diferentes versiones del estándar desde 1987. La versión actual es la 2.3 [44] que fue publicada a finales de 1996. El objetivo de este estándar es el intercambio de información entre aplicaciones informáticas en el sector salud. Su funcionamiento se sitúa en el ámbito de la capa 7 de la OSI, que es la denominada de aplicación. De aquí le viene el nombre de HL7. No se hace referencia a la arquitectura que soporta estas aplicaciones, de hecho, HL7 debe utilizar las plataformas más comunes de la industria.

Las aplicaciones a las que está orientado HL7, se pueden denominar ADT (*Admission, Discharge, Transfer*: Admisión, alta y transferencia de pacientes). Siguen el proceso de un paciente desde que entra hasta que sale del hospital o el centro de salud. En este tipo de información consta del registro del paciente, las consultas, facturación, actualización de análisis de laboratorio, historias clínicas, etc...En principio, está orientado a la gerencia de la información desde el punto de vista administrativo. HL7 coopera con otros estándares de la industria del sector salud. De hecho existen grupos

de trabajo con los estándares más importantes. También HL7 trata de utilizar toda la tecnología ya empleada, es decir, es de fácil implementación y actualización.

Es el estándar más importante en la actualidad y las compañías más importantes del sector están involucradas de una manera u otra en el desarrollo de este estándar y la cooperación con otros. En diciembre de 1996 el grupo HL7 estaba conformado por más de 1500 socios y existen más de cien empresas del sector con productos que emplean este protocolo. Muchos países tienen su propio comité, lo que da una idea del amplio desarrollo de este estándar. Además desde finales del año 94 la ANSI lo propone como estándar.

Es importante señalar, que HL7 no especifica cómo es el transporte de la información, por ejemplo, aprovecha las capas más bajas de la OSI para la transmisión de la información. Por lo tanto, a priori, se puede utilizar HL7 en cualquier canal de comunicaciones. En este trabajo se emplearía el protocolo TCP/IP para el transporte de registros que sigan el estándar HL7.

En cuanto al formato de datos propuesto en el estándar se pueden extraer las siguientes características:

- Tamaño Ilimitado. Aunque se supone un tamaño no mayor de 64K, no hay restricciones para la longitud del registro.
- Uso de las capas más bajas de la OSI para la transmisión de la información, control de CRC, inicio y fin del registro, canal utilizado (LAN, Internet, radio, etc...).
- Se define un mismo segmento de datos para todos los tipos de datos clínicos, ya sea una radiografía, un análisis de sangre, un ECG, etc... Este segmento se denomina *Observations/Results* y algunos de los campos son:
 1. *Value Type*. Especifica el tipo de valor, por ejemplo, una cadena de caracteres, un bloque de texto, un número, etc...

2. *Observation Identifier*. Código que identifica a este valor.
 3. *Observation Value*. El valor actual que se envía.
 4. *Units*. Unidad empleada (si el valor es numérico).
 5. *References Range*. Valor normal para este paciente y registro.
- Envío de datos binarios. Se pueden enviar datos binarios en un segmento denominado OBX. Estos datos se envían encapsulados según el tipo de datos *mime-encoded base-64*. La interpretación de estos datos depende de las aplicaciones en sí.
 - HL7 no es *plug&play*, esto quiere decir que se necesitan especificaciones auxiliares para la implementación de la aplicación.
 - Cada registro tiene un número de orden o serie, único para toda la transacción.

Existen algunas aplicaciones que utilizan este formato de datos para transmitir ECG por Internet [28] [45]. Esto indica que es posible, aunque no existen especificaciones públicas de cómo se ha hecho, ya que son investigaciones de tipo comercial, en el primer caso para Marquette® Electronics y en el segundo para HP®.

El sistema MUSE® es un servidor Web que funciona para una Intranet. La cooperación entre HL7 y algunos de los lenguajes estándar de Internet (HTML, XML, etc) permiten, según las especificaciones del fabricante, la transmisión y manejo de cualquier tipo de ECG (según tipo, longitud, almacenamiento). En este trabajo se han visto algunos inconvenientes para el uso de porque:

- No especifica ningún tipo de compresión ni parámetros de adquisición y fiabilidad.
- La interpretación de los datos binarios depende de la aplicación, es decir, no está especificado.
- Está orientado a procesos de tipo administrativo, y el ámbito de este trabajo es para propósitos de investigación.

Por lo tanto, para aplicar HL7 a la transmisión de registros ECG por Internet, habría que especificar un formato concreto y eso significaría diseñar un HL7 específico para electrocardiografía.

2.3.2. P1073.

Esta norma está patrocinada por el IEEE (*Institute of Electrical and Electronic Engineers*, Instituto de Ingenieros electrónicos y eléctricos). Es bien sabido el enorme peso que tiene esta institución en el ámbito tecnológico. Este apoyo hace que existan ya muchas aplicaciones y dispositivos siguiendo esta norma, entre ellas [46] [42], orientadas sobre todo al monitoreo de pacientes en UCIs.

La familia P1073 (tres estándares del IEEE) propone una interconexión e interoperabilidad de equipos médicos, independiente de los proveedores de los equipos. Este estándar pretende especificar la infraestructura para las aplicaciones informáticas en el campo médico. Se denomina también MIB (*Medical Information Bus*, Bus para el intercambio de Información Médica) debido a que se propone una arquitectura que soporte el intercambio de información y de aplicaciones.

La norma está basada en las siete capas de la ISO/OSI (International Standardization Organization/Open Systems Interconnection) que son: física, enlace, red, transporte, sesión, presentación y aplicación. Además utiliza, para cada una de estas capas algunos estándares ya desarrollados, como por ejemplo IEEE 802.x para redes locales, protocolo TCP/IP, etc... Esta característica garantiza interoperabilidad con otros estándares y es uno de los requisitos de diseño.

El comité del IEEE se formó en 1984, y su objetivo fue el desarrollo de un estándar para la comunicación entre dispositivos médicos y la adquisición automática de registros biomédicos. En diciembre de 1994 se aprueban las especificaciones relativas a las capas de transporte y física y en noviembre de 1995 se presenta una propuesta

para las capas superiores. Se debe mencionar que el mayor esfuerzo ha estado en la implementación de la capa física y de enlace, que son las de implementación asociadas al “hardware”. Por lo tanto algunos documentos de la disposición 1073.3.x aún están en revisión y no existen todavía las especificaciones completas. Las especificaciones principales de la familia P1073 son [38]:

- 1073.1 Semántica y sintaxis de MDDL (Medical Device Data Language).
- 1073.2 Controladores DCC y BCC. Red de dispositivos para cada cama. Intra Bed Network. Capas superiores: Aplicación, Presentación y Sesión.
- 1073.3 Controladores MCC y BCC. Red de todas las camas. Inter-Bed Network. Capas inferiores: física, enlace, red y transporte.

El lenguaje MDDL [47] tiene como objetivo fundamental el proveer flexibilidad y control para monitorear los diferentes dispositivos.

Estas especificaciones tratan de cumplir unos requisitos de diseño que están orientados a definir un estándar para la adquisición automática de la data y la comunicación entre dispositivos. Se orientó principalmente a unidades de cuidados intensivos (ICUs) ya que este es un ambiente donde hay que cambiar y reconfigurar equipos y donde se maneja un volumen importante de señales vitales (ECG, presión arterial, respiración, presión parcial de O₂, etc.). Este estándar está orientado a especificaciones “hardware” a diferencia de HL7.

Entre los requisitos que debe cumplir el MIB, [48] se propusieron:

- *Tecnología Apropiada.* Aprovechar la tecnología existente siempre que sea posible. Esto favorecerá la transición de los sistemas más antiguos y abaratará costos.
- *Orientado a personal médico (no técnicos).* Tratar de independizar el equipo del usuario que lo utiliza (conexión, instalación) ya que no suelen ser personas especializadas en el diseño de los equipos.

- *Autoconfigurable*. El MIB tiene que tener una gran facilidad para conectar y desconectar equipos, cambiarlos de posición (lugar en la red) e independizar la conexión del usuario.
- *Seguridad Paciente*. Diseño que provea seguridad, eficacia y precisión como es lógico en aplicaciones médicas.
- *Alarmas en tiempo real*. El retardo que ofrece la red está restringido por la necesidad de aviso inmediato de informaciones vitales como por ejemplo alarmas, averías del dispositivo, etc...
- *Independiente del equipo*. La red (flujo de información) ha de ser independiente del equipo (y compañía) que ofrece la información (adaptable a los estándares de la industria).

[49] propone algunos requisitos más, como, redundancia y confiabilidad de la información, posibilidad de expansión (modular), protección contra interferencias, etc...

Para el cumplimiento de estos objetivos se creó el comité P1073 y las especificaciones de la norma 1073.

Dentro del MIB se encuentran tres dispositivos fundamentales. Hay un host que actúa como servidor (se controla el sistema, se toman las decisiones, se recoge información, etc...) y un equipo médico que es el que realiza la adquisición. Las especificaciones de la arquitectura hablan de tres controladores de comunicaciones (DCC, BCC y MCC):

- DCC. Device Communication Controller. Es un controlador específico para el equipo médico. Se encarga de adaptar el equipo (tipo de señal, unidades, tipo alimentación, conexión, etc...) al MIB.
- BCC. Bed Communication Controller. Se encarga de gestionar la información generada, o recibida por la cama (bed) a la que atiende. Ya se mencionó antes que cada cama va a tener varios dispositivos, por lo tanto un BCC va a tener

varios DCCs asociados. Esto forma lo que se llama “Intra-Bed Network” (red de dispositivos de una cama), que es como una subred dentro del MIB.

- MCC. MIB Communication Controller. En algunos artículos también se le denomina “Host Interface”. Es la interfaz entre el Host y el MIB (bus de datos). De un MCC dependen varios BCC, esto forma la denominada “Inter Bed Network” (red de todas las camas).

En la figura 2.1 se muestra un ejemplo con dos camas y tres dispositivos por cama. La topología es de estrella (un host). En el lateral se muestran cada una de las

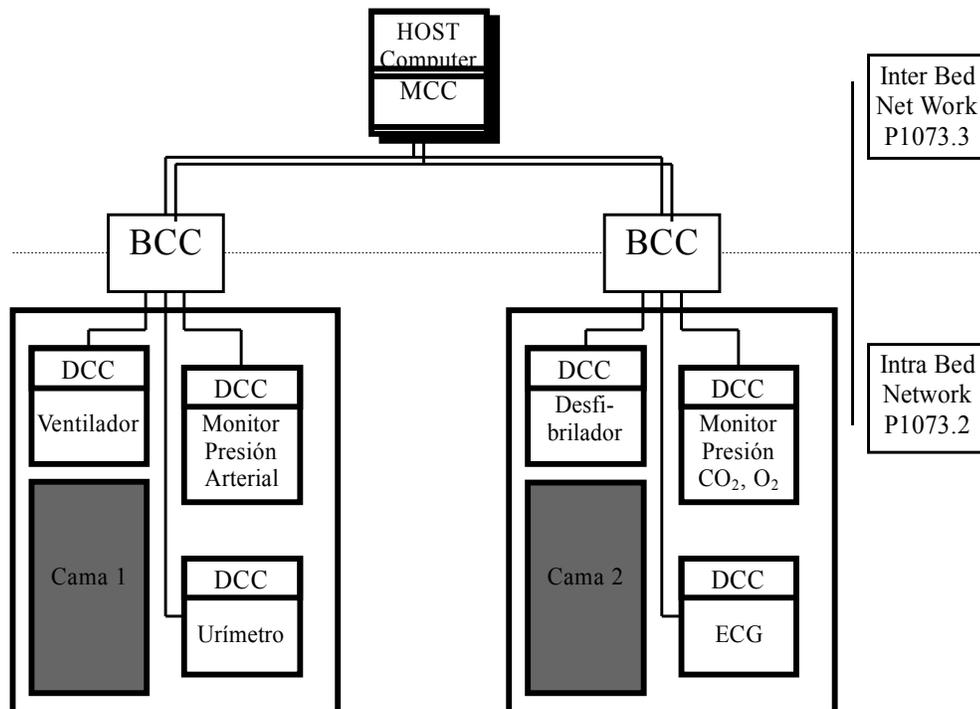


Figura 2. 1. Arquitectura del MIB.

disposiciones de la norma P1073.

Para la aplicación a la transmisión de ECG por Internet, las especificaciones generales de MIB permiten:

- Utilización del protocolo TCP/IP para el intercambio de información.

- Conexión de un dispositivo que adquiere ECG a una LAN (que utiliza TCP/IP), y por tanto a Internet. Esta facilidad prevee por lo tanto, la conexión a través de MIB de un dispositivo que adquiere ECG a Internet.

Pero en contrapartida, al igual que HL7 no define un formato de datos específico para ECG. Por lo tanto, al igual que para HL7, habría que definir un formato de datos propio, que no se ha definido todavía porque es una propuesta general para todos los tipos de registros biomédicos.

2.3.3. MS-HUG

El grupo de trabajo MicroSoft Healthcare Users Group (MS-HUG) [37] son un conjunto de empresas que han tratado de aglutinar las aplicaciones del sector salud bajo los sistemas operativos Windows 95 y Windows NT. Por tanto no es específicamente un estándar de transmisión aunque se ha nombrado debido al peso de Microsoft en el mercado.

La propuesta no deja de presentar una de las arquitecturas que se han propuesto para implementar aplicaciones cliente-servidor. Esta propuesta se denomina DCOM y se verá en el contexto del análisis y diseño de la aplicación, en el capítulo III.

2.3.4. SCP-ECG

Las especificaciones de la norma SCP-ECG (*Standard Communications Protocol for ECG*, Protocolo estándar de comunicaciones para ECG) están dentro de un proyecto más general denominado OEDIPE [50]. Las siglas OEDIPE corresponden a un estándar europeo de intercambio y procesamiento de datos de electrocardiografía computarizada (*Open European Data Interchange and Processing for Computerized Electrocardiography*).

En esencia es un intento de estandarización del intercambio de datos de electrocardiografía de la Comisión Europea de Normalización (CEN). Todo este

conglomerado de siglas no pretende más que subrayar la importancia oficial dada a este proyecto, y la necesidad que se ha visto desde la Unión Europea para la implementación de un estándar para la adquisición, almacenamiento, transmisión y análisis de datos de electrocardiografía.

El ámbito de esta norma [40] se restringe al procesado de datos de electrocardiografía. Las especificaciones que se dan no son válidas para otro tipo de registros biomédicos porque son específicas a la información de adquisición de ECGs, y en este caso a registros ECG de reposo con frecuencias de muestreo típicas entre 500 y 1000 Hz y de 10 segundos de duración cada canal (típicamente 8 a 12 canales). Esta es la razón fundamental para su escogencia para el intercambio de ECG por Internet, que es el objetivo de este trabajo, y también una de sus limitaciones como estándar.

El grupo de trabajo que desarrolló estas especificaciones está formado por grupos de investigación en informática médica de varias universidades europeas y la colaboración con empresas distribuidoras, médicos y especialistas clínicos de todo el mundo. Los primeras reuniones de discusión comenzaron en Noviembre de 1989. Asistieron representantes de más del 80% de las compañías que manufacturan dispositivos de adquisición de ECG (electrocardiógrafos) a nivel mundial.

Los objetivos fundamentales son tres:

1. Definir el formato de datos de ECG.
2. Definir los requerimientos mínimos para la codificación y compresión de ECGs.
3. Definir el control del flujo de la información entre dispositivos.

Los puntos débiles de esta norma es que está centrada en ECGs de reposo y no plantea soluciones para otros tipos de ECG. También especifica un protocolo para el intercambio de datos, pero no bajo el protocolo de transporte de Internet (TCP/IP).

2.4. CONCLUSIONES DEL ANÁLISIS

Una primera aproximación a cada uno de estos estándares muestra como cada uno trata de ofrecer una solución definitiva al problema de interoperabilidad de equipos e intercambio de información independiente del proveedor. En la práctica ninguno de ellos soluciona el problema general, y mucho menos da una solución coherente y concreta a la transmisión de ECG por Internet.

Los estándares HL7 y P1073 son demasiado generales y se necesitaría definir un formato de datos propio para su aplicación aquí. Por otra parte, el grupo MS-HUG tampoco propone un estándar como tal sino, simplemente la aplicación de herramientas Microsoft en el sector salud (Active X, objetos COM/DCOM, Sistemas Operativos Windows, etc.). Son tres maneras diferentes de atacar un problema complejo:

- HL7. Propone especificaciones orientadas al software, como registros de datos
- P1073. Orientada a equipos médicos hardware
- MS-HUG. Aplicaciones de la industria (Microsoft) al campo salud, pero sin especificaciones concretas.

Ninguno de ellos especifica por ejemplo, ningún formato de datos ni mecanismos de compresión para enviar un registro ECG a través de un canal de comunicaciones. Al contrario, la norma SCP-ECG especifica muy bien el formato de datos de un registro ECG y está totalmente orientada a registros de electrocardiografía. Sin embargo, pierde generalidad y no puede ser considerado como un estándar en informática médica.

La conclusión general que se extrae es que un estándar general tendrá que englobar otros estándares de la industria para que se consiga una integración de las diferentes soluciones. En este trabajo se utilizará la norma SCP-ECG que es el estándar que mejor se adapta a los requerimientos planteados.

2.5. ESTÁNDAR ESCOGIDO: SCP-ECG.

En este apartado se va a hacer una descripción de los aspectos más importantes de la norma SCP-ECG tal como está especificada en la última versión de la Comisión Europea para la Normalización (CEN) del año 1993 [41]. En el apéndice A se encuentra información más detallada con ejemplos de cómo se realiza la compresión, del diccionario de anotaciones y otras especificaciones. Se quiere dar una visión amplia de las principales características del estándar SCP-ECG.

Una vez seleccionado el protocolo para la transmisión de ECG vía Internet hay que estudiar en profundidad el formato de datos utilizado, el protocolo para el intercambio de información y todos los mecanismos necesarios para enviar la información contenida en un archivo que siga este estándar.

2.5.1. OBJETIVO PRINCIPAL DE LA NORMA SCP-ECG E HISTORIA RECIENTE.

Básicamente, los problemas que se quieren solucionar dentro de la propuesta de la norma SCP, enmarcada en el proyecto OEDIPE, son:

- Económicos. La adopción de un estándar común, disminuye los gastos de compra y renovación de equipos, y provee la facilidad de automatización de la adquisición de la información.
- Acceso a Bases de Datos. La transmisión y consulta de la información se podría hacer desde puntos remotos, esto permitiría la colaboración de diferentes centros de investigación y clínicos.

La meta del proyecto SCP es similar a propuestas anteriores y de hecho utiliza varios de los trabajos realizados anteriormente (Trabajos de la VA y Marquette). Se trata de diseñar unas especificaciones que permitan el intercambio de datos ECG entre diferentes dispositivos computarizados. Ese trabajo inicial se desglosó en tres

paquetes de trabajo [51]. El primero se ocupó de los tópicos generales en el intercambio de data ECG y se trabajó en centros de investigación de Leuven y Rotterdam (Bélgica y Holanda), el segundo estaba encargado de la codificación de la data, y los centros de investigación implicados fueron los de Hannover y Padova (Alemania e Italia), y por último la problemática del almacenamiento se desarrolló en los centros de Lyon y Glasgow (Francia y Reino Unido).

En definitiva, la norma SCP es fruto de la cooperación de diferentes centros de investigación europeos. Para la elaboración del estándar se celebraron varios encuentros a partir del año 1989. Participaron en el proyecto 45 expertos en electrocardiografía de todo el mundo y 13 compañías fabricantes de más del 80% de los electrocardiógrafos del mercado. Desde los comienzos fue un esfuerzo de colaboración entre el mundo académico y científico y el mundo comercial. Así en el año 1989 ya existía una primera propuesta de estándar. Después de varias propuestas, en el año 1993 se propone como preestándar por la Comisión Europea de Normalización. El proyecto SCP-ECG ha sido un esfuerzo para la creación de un estándar en electrocardiografía computarizada, y no existe ningún estándar más específico que éste en el ámbito del intercambio de datos de ECG. Aunque no sea la solución perfecta, hasta el momento es la más adecuada para el intercambio de información de ECG.

El escenario real sobre el que se enmarca el proyecto SCP [52] es similar al que se ha estado planteando en este trabajo. Ejemplos típicos de intercambio de información sobre ECGs serían:

- Un médico en la sala de emergencias almacena en la base de datos del hospital, un ECG tomado a un paciente
- Un cardiólogo consulta la BD de otro hospital para revisar otros ECGs tomados a su paciente
- Un técnico envía por teléfono el ECG de un paciente que está siendo monitoreado en su casa

- Un médico de familia puede consultar a un cardiólogo de un hospital sobre un caso general

En la figura 2.2 se puede ver un diagrama con las diferentes posibilidades. La interconexión está pensada para trabajar en redes locales y a través de Redes de Área extendida (LAN y WAN). Las evaluaciones realizadas del proyecto OEDIPE y SCP-ECG utilizan esta arquitectura [53].

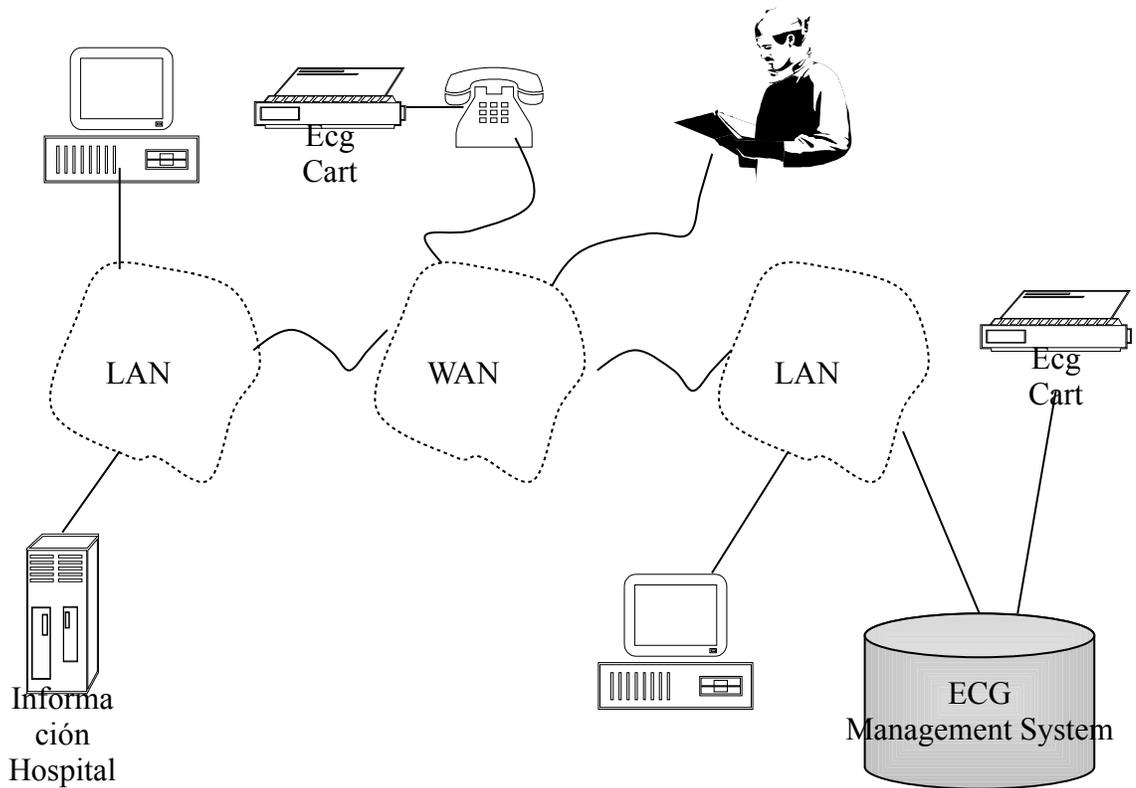


Figura 2. 2 Arquitectura general para el intercambio de ECG

2.5.2. NECESIDAD DE ESPECIFICAR UN MECANISMO DE COMPRESIÓN.

Uno de los puntos de más trabajo para el grupo SCP-ECG fue el de la compresión de los datos. Las razones fundamentales para utilizar un mecanismo de compresión serían:

- Mejorar la eficiencia en la transmisión.
- Aumentar al capacidad de almacenamiento

Si se quieren transmitir datos a través de un canal de comunicación con un ancho de banda limitado, se tiene que comprimir la data para aprovechar mejor sus características [31]. Si, por ejemplo, se envían 90 Kb (el tamaño medio de un ECG de 12 canales de 10 segundos de duración) a través de un módem de 2800 bps, se tardarían 6 minutos aproximadamente. Dependiendo de la aplicación, este tiempo podría ser aceptable o no pero, suponiendo que se quiere realizar una anotación, ¿cuánto tiempo duraría la consulta?. Por ello, se ha insistido tanto en especificar un esquema de compresión estándar.

Otra razón importante es el almacenamiento de la señal. Si un hospital realiza 10 ECGs al día, que ocupan cada uno 90 Kb (tamaño típico de un ECG de reposo), se tendría en un año más de 300 Mb de información. Esto trae dos problemas, uno técnico y otro económico. El problema técnico significa que a mayor cantidad de información se necesitará un sistema hardware más complejo para el almacenamiento, y lo mismo con el software que provea la capacidad de consulta de la información. A mayor complejidad, mayor costo, lógicamente. Se puede pensar en que actualmente no es un problema almacenar hasta gigabytes pero esto no era así hasta hace pocos años, y además la totalidad de información que se maneja hoy en día en un hospital es cada vez mayor.

El tema de la compresión de data también es importante en el procesado de imágenes médicas. Para profundizar sobre el problema de la compresión se puede consultar [54] y para compresión de ECG [55] [56] [57] y [58].

Un estándar de transmisión debería especificar qué mecanismo de compresión se ha de utilizar y/o los parámetros y medidas que se deben cumplir. Estos parámetros de medida tratan de mostrar cuantitativamente la diferencia entre la señal original y la señal recuperada (después de comprimir y descomprimir). A excepción de los métodos de compresión por reducción de redundancia de la información (métodos Huffman, p.ej.), otros algoritmos reducen la resolución de la señal y por lo tanto se pierde información. Es la clásica distinción entre métodos con pérdida de información (*lossy methods*) y los de reducción de redundancia (*lossless*). En el primer caso se hace necesaria una medida cualitativa y cuantitativa del error.

En el campo biomédico juegan un papel definitivo la opinión de los médicos y personal clínico, y ha sido tema de discusión la visualización de la data recuperada. Aquí lo importante son las medidas cualitativas, es decir, cómo “se ve” la señal recuperada. Básicamente ocurre que una reducción fuerte en un área poco significativa puede no representar un error cualitativo grande, mientras que aunque los parámetros cuantitativos arrojen valores pequeños, si éstos ocurren en porciones de la señal con mucha información (el complejo QRS es un ejemplo típico) pues la variación cualitativa es mayor. Por lo tanto, algunas características importantes a la hora de comprimir los datos de un ECG serían:

- Observar si se pierde o no información. Parámetros de medida
- Determinar el efecto cualitativo de la señal. Opinión del personal clínico
- Calcular y comparar el efecto de la compresión sobre las medidas tomadas en la señal original.

Todas estas consideraciones se han concretado, dentro de la norma SCP, en la propuesta de unas especificaciones para la señal original y la recuperada [57]. El mecanismo de compresión utilizado calcula una señal promedio para todo el registro y se obtiene de ella una señal residuo, como se explica en el apartado siguiente.

2.5.2.1. Compresión de datos en SCP-ECG.

La norma SCP permite manejar dos tipos de compresión:

- El esquema denominado “high compression”
- Compresión con eliminación de redundancia

En los dos esquemas, a las muestras obtenidas se le aplica un algoritmo de diferencias para eliminar la correlación entre muestras, y una codificación Huffman según una tabla almacenada en el archivo SCP. Este es un tipo de compresión que no pierde información (*lossless algorithm*).

En el primer esquema, se obtienen dos señales a partir de la original: una promedio o representativa y otra residual, que queda después de sustraerle a la original el promedio (*median signal*). Además, existe una diferencia conceptual importante, en el primer esquema hay pérdida de información y en el segundo no. Por lo tanto en el primer caso aplican las especificaciones (Apéndice A, tabla 1) propias de la norma para la medida de error de la señal reconstruida frente a la señal original. En la figura 2.3 se muestra el procedimiento. Las especificaciones dan valores para los parámetros de error. Un parámetro típico es el valor cuadrático medio (RMS) del error.

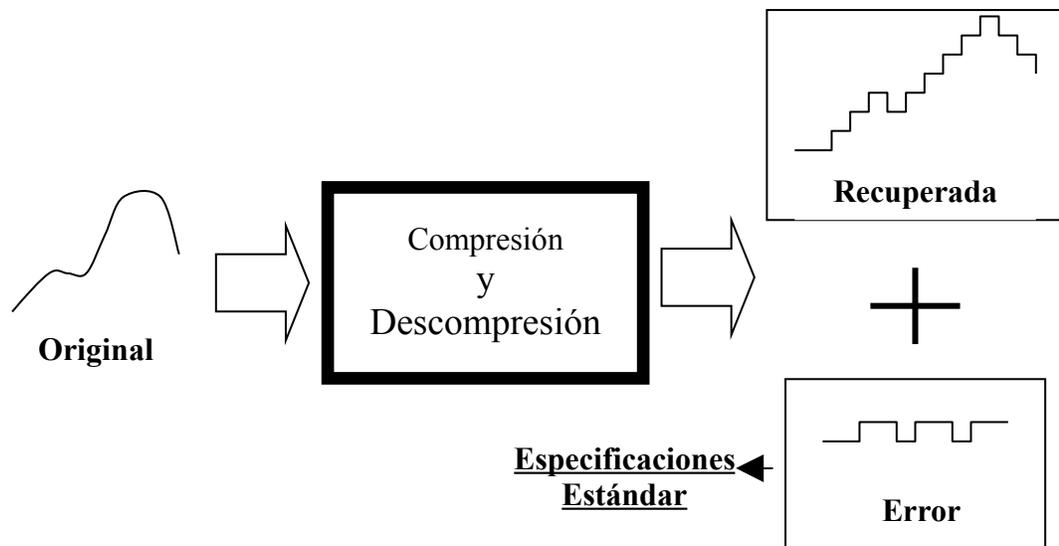


Figura 2. 3 Especificaciones sobre la señal recuperada

Para obtener la señal promedio, primero se clasifican todos los complejos QRS presentes en el canal. Para cada QRS se determina el punto fiducial (máxima amplitud) y las muestras de comienzo y fin de cada complejo QRS. Los punteros que especifican estos valores se almacenan en la sección 4 y 7. Además estos complejos se clasifican, y todos ellos se *protegen* de las etapas de diezmado y filtraje que son las etapas que dan una mayor característica de compresión (de ahí el nombre “alta compresión”, *high compression*).

Esta *protección* significa que la etapa de diezmado y filtraje no se aplicará a toda la señal sino tan solo a las áreas que no están protegidas. Estas áreas son principalmente los complejos QRS, que por otra parte tienen la información más valiosa de la señal ECG.

El cálculo de la señal promedio y de la detección de los QRSs ha sido criticado y se plantea como uno de los inconvenientes de este tipo de compresión [31]. En el apéndice A se encuentra en detalle y con ejemplos el esquema de compresión y descompresión. Simplemente para ver la complejidad del esquema de alta compresión vamos a citar los pasos a seguir para descomprimir una señal ECG que es lo que se ha implementado en la aplicación:

- Decodificación con la Tabla Huffman
- Reconstrucción de la señal en diferencias
- Interpolación de las muestras diezmadas (zonas no protegidas)
- Filtraje Paso Bajo (zonas no protegidas)
- Multiplicación por la constante de amplitud (nV)
- En el caso de alta compresión, se aplican los apartados a, b, e para la señal promedio. Además se ha de alinear la señal promedio con cada uno de los complejos QRS y sumar las señales promedio y residuo.

Por lo tanto, es un esquema de cierta complejidad, ya que se buscan elevadas tasas de compresión (aproximadamente 15 a 1). Todos los parámetros necesarios para realizar la descompresión se encuentran almacenados en las especificaciones del formato de datos.

2.5.3. FORMATO DE DATOS QUE SE UTILIZA.

La norma SCP está orientada a ECGs de reposo, de 10 segundos de duración típica y con 12 canales de adquisición. También se pensó en su transmisión por redes asíncronas y enviando la información completa o en bloques, por ello se hace un hincapié especial en la utilización de códigos de detección de errores (código de redundancia cíclica, CRCs). En el apartado de ventajas y desventajas se analizará en más detalle los inconvenientes de estas consideraciones, por el momento se presentará el formato de datos sin mayor análisis.

El archivo SCP está conformado por secciones. Cada sección es un conjunto de datos especificados dentro de la norma [41]. Están especificadas 12 secciones, aunque se ha dejado la posibilidad de que los fabricantes definan hasta la 1024. Se recomienda el uso de una misma estructura para cada sección. No se pretende explicar todos los detalles del formato de datos sino una explicación general para la mejor comprensión de las especificaciones. Se muestra una mejor documentación en el Apéndice A y en [41].

En la tabla 2.1 se pueden observar las secciones que ya se han definido, y a continuación se da una pequeña explicación del contenido de cada una.

Obligat	2 Bytes. CRC de todo el archivo.
Obligat	4 Bytes. Tamaño de todo el archivo en bytes.
Obligat	(Sección 0) Punteros a las secciones del archivo
Obligat	(Sección 1) Información general Datos sobre el paciente y la adquisición

Opcion	(Sección 2) Tabla/s Huffman SI se utilizan en la compresión
Opcion	(Sección 3) Definición de los canales de adquisición empleados
Opcion	(Sección 4) Localización de los QRSs (si se codifica la señal promedio)
Opcion	(Sección 5) Muestras codificadas de la señal promedio (si se utiliza)
Opcion	(Sección 6) Muestras codificadas de la señal residuo (sustracción señal promedio) O Muestras codificadas de la señal sin sustracción
Opcion	(Sección 7) Medidas sobre el registro ECG. Anotaciones
Opcion	(Sección 8) Diagnóstico del dispositivo que realiza el análisis
Opcion	(Sección 9) Diagnósticos especiales del fabricante y nuevas interpretaciones
Opcion	(Sección 10) Medidas sobre cada canal de adquisición
Opcion	(Sección 11) Interpretaciones. Anotaciones

Tabla 2. 1 Secciones del Formato de datos en la norma SCP.

La longitud no es fija ya que depende del tipo de compresión elegido, el número de canales, datos demográficos, etc...

Los datos almacenados se puede clasificar en:

- Tipo texto. Datos demográficos, anotaciones, medidas, etc...
- Tipo numérico. Datos relativos a las muestras almacenadas, tabla Huffman, Datos de los canales adquiridos, etc...

El proyecto SCP hizo un esfuerzo notable para tratar de ofrecer un léxico adecuado para el almacenamiento de interpretaciones clínicas de la señal ECG. Por ello la propuesta de la norma SCP es un conjunto de mnemónicos y modificadores que pueden reflejar desde interpretaciones simples hasta las más complejas [51] [41]. Por otra parte, se almacenan medidas sobre la señal como por ejemplo duración de la onda P, o detección de los complejos QRS, que son ejemplos típicos de anotaciones.

2.5.4. PROTOCOLO DE TRANSMISIÓN.

La norma SCP proporciona un protocolo de transmisión (Capítulo 8 de [41]) de bajo nivel para el envío de un archivo adquirido en un electrocardiógrafo (*cart*) a una computadora (*host*). Este protocolo especifica el tipo de conexión, las tramas de envío, los mensajes y la máquina de estados para la transmisión-recepción. Las aplicaciones trabajan sobre este protocolo.

En este trabajo, sin embargo se necesitaría un protocolo propio, o una adaptación del propuesto para la aplicación en Internet. Ocurre que el protocolo de transporte en Internet ya existe y se denomina TCP/IP. Y la conexión a Internet (las capas más bajas de la OSI) también están generalmente especificadas según el canal de comunicaciones empleado (vía telefónica, redes locales, enlaces por radio...) por el proveedor (ISP). Por ejemplo, en Internet, es el protocolo TCP/IP el que se encarga de chequear la redundancia, si la información llega bien o no, la comunicación entre el transmisor y el receptor, etc.

Así, el protocolo de transmisión aquí empleado será uno propio del lenguaje Java y que utiliza TCP/IP (Capítulo III). Este punto ha sido importante en la discusión de este trabajo y simplemente se plantea utilizar protocolos ya establecidos, y empleados por un sinnúmero de aplicaciones en Internet. Esta decisión ayudará a crear una aplicación lo más fácilmente ampliable y “estándar” y permite un diseño del sistema independiente del protocolo de transmisión escogido. Esta discusión se planteará en

el capítulo siguiente a la hora de seleccionar la técnica RMI (Invocación de métodos remotos) de Java.

Al utilizar un protocolo más general que no sea específico para esta aplicación, se pierde eficiencia en la comunicación. Si se diseñara un protocolo *ad hoc* se aprovecharían mejor las capacidades del canal, aumentando fundamentalmente la velocidad de transmisión.

Entonces, ¿no es incoherente utilizar compresión de los datos, y por otra parte perder eficiencia en la velocidad aquí? Depende de la aplicación. En nuestro caso, el objetivo es tener una aplicación de fácil ampliación y reutilización, con un diseño consistente que permita albergar diferentes estándares y que funcione sobre una plataforma ya estándar de por sí, como es Internet. Se dispondrá de una aplicación cliente-servidor que se podrá adaptar tanto al acceso de una base de datos como a un electrocardiógrafo que este adquiriendo en tiempo real.

2.6. CONCLUSIONES Y PROPUESTA DE APLICACIÓN.

Como conclusión general del análisis de los estándares se podría decir que las especificaciones de HL7, P1073 y MS-HUG son tan generales que se necesitaría plantear un estándar nuevo. La especificación de un estándar necesita de un gran esfuerzo por parte de la academia, y de la industria, y no ha sido el objetivo en este trabajo. En este trabajo se aprovecharán los trabajos anteriores de investigación en la transmisión de ECG y se utilizará un estándar ya planteado (SCP-ECG). Los futuros estándares se verán influidos por los actuales planteamientos, y mejorarán sus limitaciones. Para ello se necesitan aplicaciones que trabajen con este estándar, y existe muy poca documentación y evaluación pública por investigadores que no pertenezcan a los grupos que desarrollaron el proyecto SCP-ECG.

La gran ventaja de SCP-ECG es que está pensado para la transmisión de registros de electrocardiografía. Dentro del Proyecto OEDIPE se pensó en el SCP-ECG como

herramienta de comunicaciones para una plataforma común en el intercambio de ECGs, lo que se ajusta a los requisitos que se han planteado en este trabajo.

Se especifica un esquema de compresión muy elaborado y con altas tasas de compresión lo que se hace imprescindible para una transmisión eficaz. Un inconveniente de este esquema de compresión es la necesidad de la detección de los complejos QRS y el cálculo (no obligatorio) de una señal promedio. Además, es el único estándar que especifica un formato de datos tan concreto para ECGs y sobre todo, que ha hecho un hincapié especial en la etapa de diagnóstico e interpretación. Tiene un diccionario de anotaciones completo y orientado a la interpretación de ECGs en bases de datos de investigación.

El formato de datos se ha orientado a registros ECG en reposo justificando que el resto (pruebas de esfuerzo, registros Holter, monitoreo de cuidados intensivos...) no son tan comunes, de ahí que, por ejemplo, no sea posible guardar un registro Holter de 24 horas de duración, o un solo canal para monitoreo. Esta consideración es un punto en contra para que un protocolo sirva para varios tipos de ECG y por lo tanto para que se convierta en un estándar.

También es importante decir que está orientado a la transmisión en redes asíncronas y en funcionamiento "batch", es decir que se transmite por bloques o entero. Por ello es una constante la aplicación de códigos de redundancia cíclica (CRC) para el archivo global y para cada sección. Estos requisitos de diseño eran adecuados para la época de desarrollo del estándar (89-93) pero en la actualidad hay un énfasis especial en las redes LAN y el acceso a Internet, mientras que el estándar había ido por las redes de área extendida (WAN) y el acceso a éstas vía telefónica.

Además, ha sido un inconveniente no poder acceder al software que realiza la compresión y descompresión de datos y su almacenamiento en el formato especificado. Es decir no existe el acceso público a los algoritmos de lectura y grabación en formato SCP. Por estos inconvenientes, a pesar de que se ha propuesto en el año 1993, todavía no es aceptado como estándar por la industria y la academia.

En la figura siguiente (figura 2.4) se resumen las principales ventajas e inconvenientes que se han observado.

Pros y Contras del SCP ECG

Ventajas	Inconvenientes
Específico para ECG Mecanismo de Compresión Diagnóstico y Anotaciones Trabajo Académico y de la Industria	Orientado a ECG de reposo Necesita señal promedio y detección QRS Protocolo Redes asíncronas Software No público No implementa TCP/IP

Figura 2. 4 Ventajas e inconvenientes de la Norma SCP

Aunque uno de los inconvenientes ha sido que se ha aplicado sobre todo en redes asíncronas, en este trabajo se realiza una propuesta para la comunicación entre una aplicación cliente-servidor en Internet. Así la implementación de la norma SCP, en este trabajo, ofrece las siguientes características

1. Aplica la norma SCP en Internet
2. Establece una arquitectura cliente-servidor abierta
3. Desarrolla documentación pública para una mayor extensión de la norma
4. Desarrolla una clase que permite acceder a un archivo en formato SCP. Está desarrollada en Java y cualquier programador puede introducirla en su esquema de comunicación particular.

Esta es la propuesta para la aplicación cliente-servidor para la transmisión de ECG vía Internet, que será implementada en este trabajo. Se desarrollará un lector de archivos en formato SCP y una arquitectura cliente-servidor para la transmisión de archivos con este formato y con posibilidades de ampliación a otros estándares. En el capítulo siguiente se muestran las diferentes etapas en el desarrollo de la aplicación.

III. CICLO DE VIDA SOFTWARE

En este capítulo se mostrarán los resultados obtenidos en este trabajo: la realización de una aplicación cliente-servidor para la transmisión de ECG vía Internet. La descripción de la aplicación se hará siguiendo las etapas del ciclo de vida software. Se ha empleado el DOO para la implementación de las clases y objetos. El código público con todas las clases y sus métodos se pueden obtener en <ftp://aponguao.gbba.usb.ve>.

Se ha puesto especial énfasis en la parte de análisis y diseño. Además se muestran las clases implicadas en el sistema y algunos ejemplos de métodos. Las etapas del ciclo de vida del software son: análisis, diseño, evaluación y modificación. Las dos últimas partes son la programación y las diferentes versiones o ampliaciones del sistema.

En el DOO estas etapas se realimentan constantemente, como se ve en la figura 3.1 Es decir que continuamente se está revisando cada parte del ciclo de desarrollo de la aplicación. Aquí se presentarán los resultados finales en cada etapa.

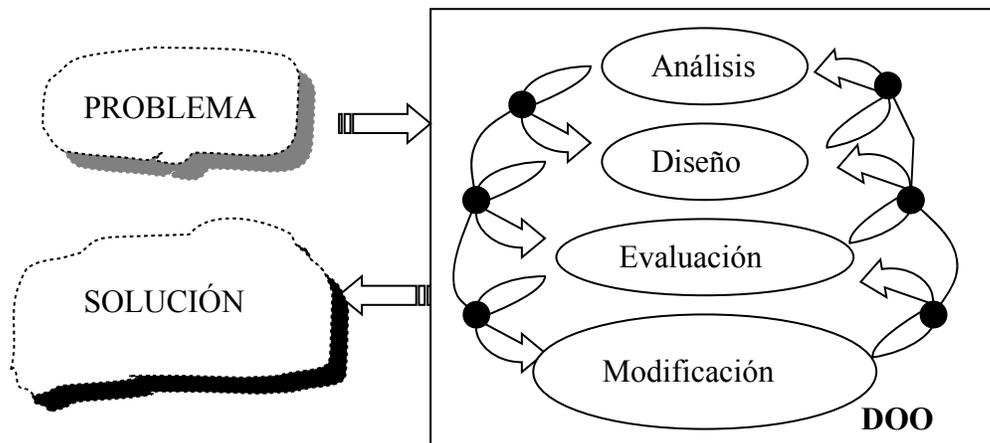


Figura 3. 1 Ciclo de vida Software en la metodología de DOO (Etapas)

3.1. ANÁLISIS

En esta etapa se determinan las especificaciones y requerimientos del sistema, y las posibilidades de solución. También se muestran las diferentes alternativas de la arquitectura del sistema.

3.1.1. ESPECIFICACIONES Y REQUERIMIENTOS DEL SISTEMA

Aunque ya se han planteado las especificaciones de la aplicación, aquí se concretan los detalles de la implementación práctica. El resultado principal será implementar un paquete de clases que permita la transmisión de datos de ECG por Internet. Un paquete en lenguaje Java es, básicamente, un conjunto de clases englobadas bajo un mismo nombre. Java, al ser un lenguaje de programación orientado a objetos, no es más que un conjunto de paquetes, por ejemplo, clases de entrada salida (`java.io`), o para objetos remotos (`java.rmi`), o del lenguaje (`java.lang`), etc... Por lo tanto, el resultado final consistirá en añadir un conjunto de clases dirigidas a la transmisión de ECG por Internet.

Una de las aplicaciones directas es un visualizador-annotador remoto de registros ECG. Este visualizador-annotador remoto consta de un applet que visualiza un registro ECG, enviado a través de la red, y permite la anotación por parte del usuario a través del ratón de la computadora y unas ventanas de selección (figura 3.9). Estas anotaciones no modifican otras partes del archivo y seguirán un diccionario prefijado para el estándar utilizado. Además el estándar especifica el formato de datos utilizado. El diseño ha de permitir el uso de varios estándares, por ejemplo, diferentes formatos de datos, diccionarios de anotaciones, etc...

El archivo ECG puede contener varios canales de adquisición e información anexa a la forma de onda (datos demográficos y de adquisición): datos del paciente, especificaciones de la adquisición, etc... Se desea también que pueda transmitir ECG de diferentes duraciones y tipos (no necesario). Si hay algún tipo de compresión, se

enviará, lógicamente, la información comprimida. El protocolo ha de ser robusto y transparente a la aplicación que lo emplee. Este protocolo se basa en el estándar TCP/IP que es el utilizado por Internet para la transmisión de datos.

El médico puede realizar consultas sobre el registro de ECG, por ejemplo, puede solicitar ver un canal entero, o algunos ciclos, varios canales, etc... Aquí se entenderá por consulta la especificación de un tipo de acceso a la información de un registro ECG. No debe confundirse con una consulta de tipo clínico. Todo el proceso asociado a los diferentes tipos de consulta serán transparentes al tipo de archivo ECG utilizado. Se incluirán métodos de consulta básicos y se deja la interfaz para la implementación de consultas más elaboradas, o el acceso a bases de datos locales o remotas, etc... Se implementará un diseño donde sea fácil agregar diferentes métodos de consulta. El resultado de una consulta será un archivo, o varios, que cumplen con la búsqueda solicitada. En esencia es una operación de búsqueda de información, generalmente sobre una base de datos. Ejemplos típicos en este trabajo serán:

- Búsqueda por nombre, o fecha de adquisición.
- Petición de abrir un archivo con un ID o Nombre de un paciente
- Petición de un número de muestras (en tiempo) de un canal.
- Petición de varios canales completos.

La transmisión se hará a través de Internet. La operación se realizará desde un navegador estándar, por ejemplo, Netscape Communicator, Opera o Microsoft Explorer. Han de tener soporte para la versión de Java 1.1 o superior.

La evaluación y prueba de este esquema de transmisión y anotación consistirá en el uso del paquete de clases implementado, por un anotador remoto de señales ECG. El diseño de este anotador no es el objetivo de este trabajo y se modelará la parte gráfica por medio de un objeto (clase) denominado GraphicManager¹.

¹ Los nombres de las clases y métodos se han escrito en inglés, debido a que se pretende una divulgación del trabajo y pedir colaboración en la ampliación y mantenimiento del código.

Por último salientar, que al ser una aplicación remota, se debe utilizar una arquitectura cliente-servidor (CS). En este caso, varios cardiólogos utilizando una aplicación cliente, pueden anotar un mismo archivo ECG ofrecido por un servidor. Los servicios ofrecidos en Internet siguen, normalmente esta arquitectura, por ejemplo correo electrónico o el acceso a páginas Web (HTML).

Con todo ello, los requerimientos a tener en cuenta para el desarrollo de las clases son:

- Transparente a la aplicación que utilice el paquete desarrollado. Se definirá la interfaz del protocolo, y se modelará la aplicación por un objeto denominado GraphicManager. Por tanto, la librería de clases podrá ser utilizada por diferentes aplicaciones.
- Fácil ampliación a otros estándares, otros tipos de consulta, etc...
- Arquitectura CS sobre TCP/IP.
- Se utiliza un navegador estándar de Internet.

El primer problema a resolver será la arquitectura CS que se va a escoger. Este tema es de mucha actualidad en Internet y existen varias implementaciones. A continuación se mostrará primero las posibilidades de acceso a archivos remotos y después las alternativas para su implementación.

3.1.2. ARQUITECTURA DEL SISTEMA

En esta aplicación un servidor HTTP provee información por medio de páginas HTML a un navegador. La página HTML puede contener una aplicación incrustada (applet) o también un CGI (Common Gateway Interface), que son formas de ejecutar programas desde páginas Web. Por lo tanto, la transmisión de ECG se realiza entre la aplicación servidor y un programa cliente. Suponiendo que se utiliza un applet cliente, éste pide un archivo ECG y el servidor se lo envía, según el protocolo

establecido. Se ha de implementar el applet en Java porque todos los navegadores poseen un interpretador de este lenguaje.

En este punto se plantean varias soluciones [59] y diferentes arquitecturas a la transmisión de un archivo desde una aplicación cliente (applet) a una aplicación servidor. La organización de estos archivos tampoco está especificada en los requerimientos, por lo que vamos a considerar que pueden estar organizados dentro de una base de datos o con otros esquemas que se propongan. Se ha generalizado el esquema en un acceso a base de datos, pero no se especifica un modelo relacional concreto, es decir, no se tendrá en cuenta la organización de la información.

En la figura 3.2 se ve una conexión del applet cliente con una base de datos relacional. Para esta conexión, Java tiene una serie de clases especializadas que implementan el driver especificado para esa base de datos. Existen drivers para Oracle, SyBase, miniSQL, etc... Este tipo de conexión se denomina JDBC (Java DataBase Connection, Conexión Java-Base de Datos) y es un estándar dentro del lenguaje Java.

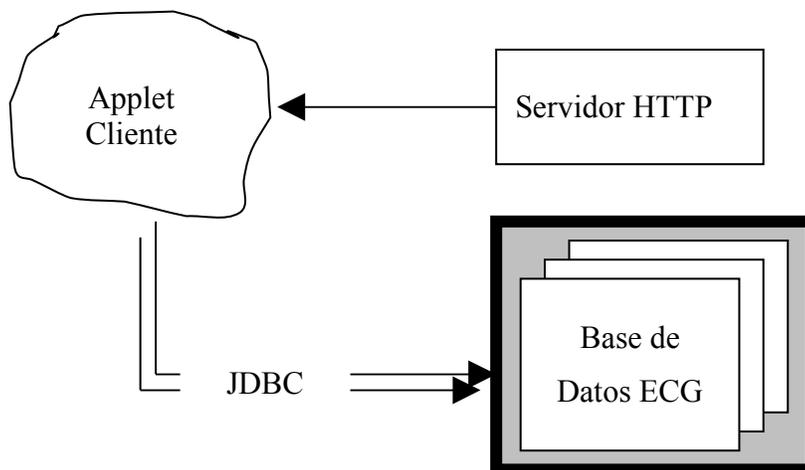


Figura 3. 2 Conexión JDBC

Esta arquitectura es muy utilizada y existen herramientas de desarrollo para este esquema. Para implementar este diseño es necesario conocer exactamente el tipo de base de datos a utilizar. Es decir la conexión es específica para cada tipo de Base de Datos, ya que se necesita un “programa” diferente para cada tipo de BD. Así, se hace

más difícil la ampliación a nuevos tipos de BD y se ha presentado como una desventaja de este esquema. También se ha visto como un problema la falta de estabilidad entre la conexión de la BD y el applet.

Otro esquema posible es hacer la conexión con una base de datos relacional a través de un CGI. Un CGI (Common Gateway Interface, dispositivo de comunicación entre la página HTML y el servidor de información HTTP) permite lanzar programas que están en el servidor y realizar consultas en una base de datos o leer de archivos locales. Estas consultas se realizan, como en el caso anterior, a través de librerías especializadas para cada tipo de Base de Datos. Este sistema es de gran utilidad para la creación de tablas de información, búsquedas temáticas, etc, pero no para visualizar registros de señales digitalizadas, como es el objetivo aquí. La figura 3.3 muestra la arquitectura general de este tipo de acceso.

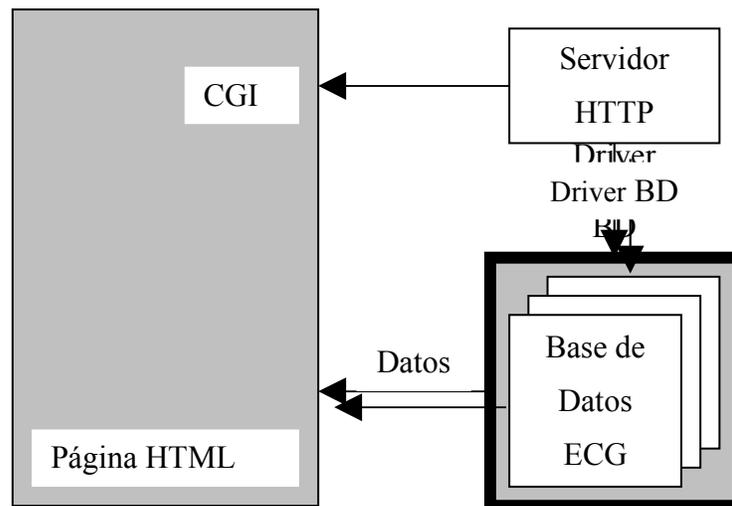


Figura 3. 3 Conexión a través de CGI

El paso de parámetros para el programa que está en el servidor se realiza a través del URL (*Uniform Resource Locator*, que es la dirección de esa página HTML) lo que limita la cantidad de parámetros y variables a utilizar.

Los esquemas que se han presentado trabajan con bases de datos relacionales principalmente. El análisis y diseño de una base de datos relacional es un problema complejo. Además de la definición de la estructura de los datos y la relación entre los miembros de esa estructura (campos), se necesita programar la aplicación y almacenar los datos en algún dispositivo. Como cada estándar especifica un formato de datos diferente, el modelo de almacenamiento de cada uno de ellos será diferente y se necesita un esquema que sea lo más independiente posible de la base de datos para la transmisión al cliente.

La tercera posibilidad (figura 3.4) consta de un applet cliente que se conecta directamente a una aplicación-servidor de archivos. Este servidor de archivos es una aplicación que atiende las peticiones de archivos ECG. En la figura también están en sombreado las posibles conexiones entre esta aplicación y una base de datos o lectura de archivos binarios. Este esquema se ha denominado "3 tier"[60] porque consta de tres partes: configurar una aplicación servidor (del applet), establecer la conexión entre el servidor y la base de datos (o archivos locales) y establecer la comunicación entre el applet y el servidor (apartado siguiente). Estos esquemas son temas de discusión entre desarrolladores de software y esta solución es una de las más empleadas por la robustez y facilidad de ampliación y mantenimiento [59].

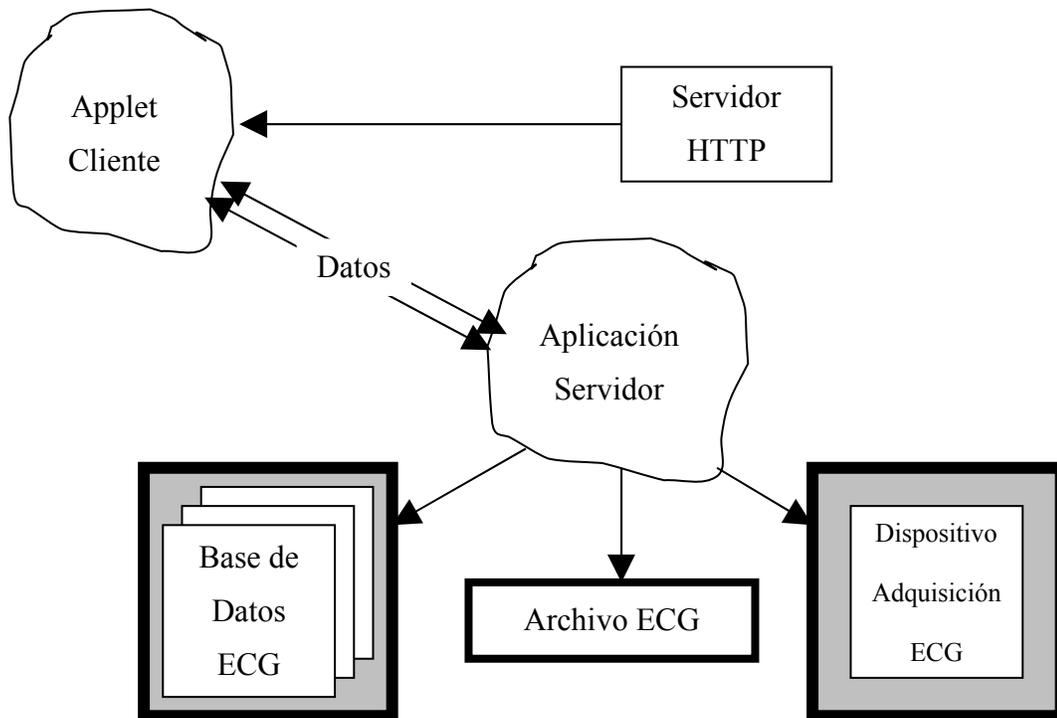


Figura 3. 4 Applet y aplicación servidor

Ya que no hay restricciones en cuanto a la arquitectura a utilizar se implementará una aplicación CS entre un applet y un programa ejecutándose en un servidor. El servidor de archivos atenderá las peticiones de archivos ECG. Las ventajas que presenta este esquema son:

- Es mucho más modular que los anteriores y fácilmente ampliable cambiando solamente el servidor.
- Puede adaptarse a distintas aplicaciones: acceso a una base de datos, conexión con un electrocardiógrafo, lectura de archivos locales, etc...
- Es una arquitectura típicamente cliente-servidor, con las ventajas que esto conlleva.

Estas razones que facilitan la realización y se ajustan mejor a los requerimientos, justifican la escogencia de este esquema.

3.1.3. ARQUITECTURAS CLIENTE-SERVIDOR EN INTERNET.

En el apartado anterior se discutieron las diferentes posibilidades de acceso a archivos remotos en Internet. Pero, ¿cómo se entenderán el applet cliente y el servidor?. ¿Qué protocolo utilizarán para comunicarse?

Se propondrán diferentes alternativas a la conexión cliente servidor, sobre todo aquellas que están basadas en el DOO. En primer lugar se introducirá brevemente qué es un socket y después tres técnicas de mucha actualidad en la implementación de objetos remotos.

3.1.3.1. Conexión básica: Sockets

Un *Socket* [61] es cada punto a través del cual, una aplicación accede a un canal de comunicaciones. Es decir, algo así como la vía de acceso a ese canal. Esta definición es global, no sólo está enmarcada en una aplicación a Internet. En Java se proveen dos clases para crear sockets, que permiten crear comunicaciones entre una aplicación cliente y otra servidor. Además estas clases ya emplean el protocolo TCP/IP que es un requisito de las especificaciones.

En la figura 3.5 se observa la comunicación entre un cliente y un servidor por medio de un socket. Un cliente envía un mensaje A al servidor y este le envía un mensaje B.

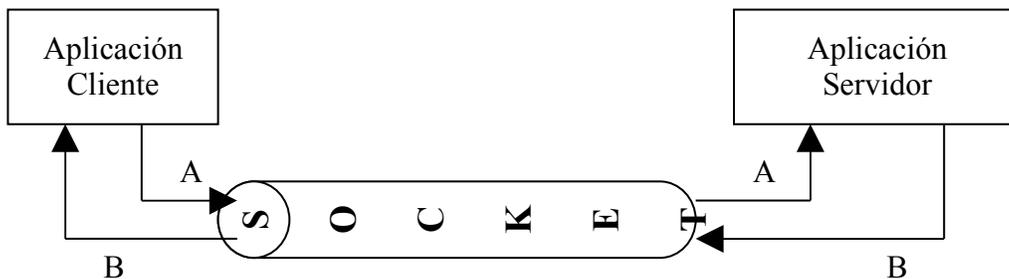


Figura 3. 5 Comunicación con sockets

Las ventajas de este esquema son principalmente:

- Eficiencia: es la implementación más básica y simple, por lo tanto requiere de menos procesado y es más rápida.
- Flexibilidad: se puede hacer un protocolo de comunicación propio. Esto implica interpretar el mensaje A para devolver un mensaje B.

En este caso, a la hora del diseño se utilizan las clases y los métodos (o se heredan las propiedades) de Java para la comunicación a través del socket. Los sockets están asociados a un puerto de la máquina. Un puerto es el lugar de entrada de las peticiones de comunicación al servidor. Por lo tanto el procedimiento de “instalación” de un socket en Java (paquete java.net) sería [62]:

- Servidor: Se crea un socket (asignado a un puerto) y espera mensajes.
- Cliente: Busca la conexión a un host y un puerto.
- Servidor: Acepta la conexión y crea el socket
- Cliente y Servidor: Se envían datos a través del socket.

Entre el servidor y el cliente ha de establecerse un protocolo de comunicación que utilice sockets. Cada mensaje A o B necesita que se interprete por parte del emisor y receptor.

El uso de sockets se integra dentro del diagrama de clases y objetos, pero cada objeto pertenece o al cliente o al servidor y hace que se pierda flexibilidad en el diseño de aplicaciones en red. Un nuevo concepto que ha surgido es el de los objetos distribuidos.

3.1.3.2. Desarrollo de aplicaciones utilizando Objetos Distribuidos

Estas técnicas que se explicarán aquí, implementan un objeto distribuido [63], que tiene métodos locales y métodos remotos. En este tipo de objetos los métodos no están en “el mismo lugar” y existe una arquitectura CS dentro del mismo objeto.

En la figura 3.6 se muestra este nuevo concepto. El objeto distribuido proporciona métodos que pueden accederse localmente o remotamente. Un método es el mecanismo de comunicación entre objetos. Este objeto, y por lo tanto sus métodos, se

puede llamar desde la aplicación cliente o desde la aplicación servidor. Para la creación de objetos distribuidos se necesitan especificaciones para el protocolo de comunicación a través de Internet y esto es lo que ha dado lugar a diferentes estándares en la implementación de objetos remotos.

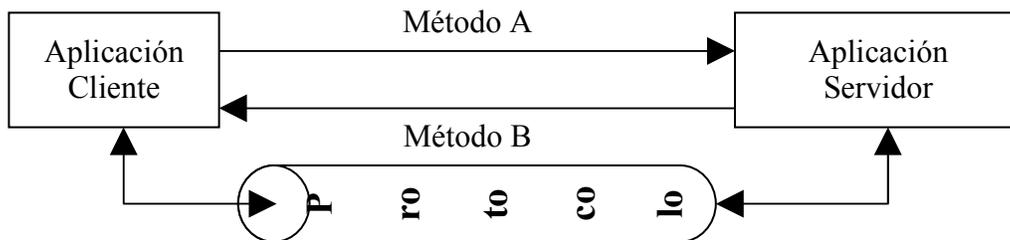


Figura 3. 6 Objetos Distribuidos

Las razones que se esgrimen para la utilización de objetos remotos o distribuidos[63] son:

- Reutilización de código para la implementación más rápida y funcional de aplicaciones (RAD, Rapid Application Development)
- Mantenimiento de la aplicación: ampliación y sistematización. Fácil Actualización
- Distribución de las tareas, dejando las más costosas en el servidor.

Lógicamente muchas de las ventajas provienen de que se utiliza la metodología de DOO, orientado a la red. Las tres tecnologías que existen para crear objetos distribuidos son RMI, CORBA y DCOM. En la etapa de diseño se determinará cual se escoge y porqué, aquí se está haciendo una descripción de las técnicas que se podrían utilizar.

Una vez que se ha diseñado el diagrama de clases, se aplica una de estas técnicas para su implementación práctica. Esto quiere decir que el diseño que aquí se presentará sigue siendo básicamente el mismo, independientemente de que se emplee RMI, CORBA o DCOM. Cada una de estas especificaciones se añadirán al diagrama general. Para una descripción más general de cómo se aplica cada técnica en una aplicación Java se puede consultar [64] [65] [66].

3.1.3.2.1. RMI

RMI [64] [68] (Remote Method Invocation) significa invocación de métodos remotos. Esta técnica está integrada en las especificaciones del JDK 1.1[23]. JDK es la herramienta estándar para el desarrollo de programas Java. RMI da la posibilidad de llamar a métodos a través de la red, estos métodos estarán en una o varias máquinas distribuidas por Internet. Es la técnica más fácil de emplear para la implementación de un objeto remoto. En la etapa de diseño se especificará cómo se ha aplicado esta técnica al diagrama de clases utilizado.

Entre las limitaciones que presenta es que tiene un protocolo de comunicación propio, por lo tanto, un cliente RMI sólo se puede “entender” con un servidor RMI, y también que es una técnica que sólo se aplica en Java. Es decir, la programación de todos los métodos ha de ser en Java, no en otros lenguajes. Estas limitaciones indican que, en el caso de aplicaciones complejas, no es tan robusto y fácilmente ampliable como la arquitectura CORBA y DOM.

Estas limitaciones están fuera de los requisitos de la aplicación, por lo tanto se aprovecharán las características de integración en Java 1.1, su sencillez y rapidez de integración para su utilización en este trabajo.

3.1.3.2.2. CORBA

CORBA [65] (*Common Object Request Broker Architecture*) permite el acceso a métodos remotos como si fueran locales. Ha sido desarrollada por el OMG (*Object Management Group*) en 1990. La última especificación es CORBA 2.0 y especifica la interfaz de comunicación entre objetos distribuidos a lo largo de la red. La comunicación entre un cliente y el objeto (que tiene la implementación del método) se hace a través de la interfaz ORB (*Object Request Broker*).

En la actualidad es la técnica más adecuada para el desarrollo de aplicaciones complejas distribuidas por la red. Los objetos se pueden implementar en diferentes lenguajes de programación y en diferentes plataformas. Está ampliamente aceptado en la industria, y compañías como IBM, Netscape y Oracle han desarrollado este tipo

de especificaciones en sus aplicaciones. Esto quiere decir que el acceso, por ejemplo, a una base de datos Oracle se puede hacer utilizando una interfaz ORB. Esta es la potencialidad de CORBA, que define un estándar para la creación de objetos distribuidos, bajo cualquier plataforma y en cualquier lenguaje. La especificación es, en esencia un protocolo de comunicaciones entre objetos.

Sin embargo, es una técnica más compleja y hay especificaciones que no han sido implementadas.

3.1.3.2.3. DCOM

DCOM [66] significa Modelo de Componentes de Objetos Distribuidos (*Distributed Component Object Model*). En esencia es una copia de la idea de CORBA con la particularidad de que está adaptada al entorno Windows, y con la potencia que implica el mercado de Microsoft. Microsoft planteó esta arquitectura en el año 1996.

Se debe remarcar aquí que esta es la propuesta que MS-HUG propone como estándar para aplicaciones de salud bajo entorno Windows. Es decir que el estándar de informática médica, en realidad es un estándar de objetos distribuidos (aplicaciones cliente-servidor). En realidad no propone ningún formato de registro biomédico sino simplemente la arquitectura de la aplicación, al igual que las técnicas de objetos distribuidas que se han comentado aquí. Por ello no hace un aporte significativo a la informática médica como tal sino unas especificaciones generales.

La propuesta es un protocolo de comunicaciones para el intercambio de mensajes entre objetos. Es decir, la misma propuesta de los dos esquemas anteriores. Como se puede ver, cada técnica en su medida trata de imponer un estándar para desarrollar una aplicación que pueda utilizar objetos distribuidos en la red.

La gran ventaja de este esquema es que está patrocinado por Microsoft y la gran desventaja es que sólo funciona bajo ambientes Microsoft. Tanto CORBA como DCOM son las dos grandes técnicas para aplicaciones complejas con objetos distribuidos en Internet.

3.2. DISEÑO

En esta etapa se han tomado las decisiones sobre la arquitectura del sistema y las clases y objetos involucrados. La aplicación consta de un applet que accede a métodos remotos (RMI) de una aplicación servidor. También se mostrará el esquema de clases que se desarrollará.

3.2.1. OBJETIVOS DE DISEÑO. ARQUITECTURA EMPLEADA.

Para aplicaciones en tiempo real o para un programa firmware en un microcontrolador de un dispositivo de adquisición por ejemplo, la solución con sockets es la más eficiente desde el punto de vista de acceso al canal y programación dedicada (*custom*). Pero, en este trabajo se ha hecho especial hincapié en la coherencia del diseño y en la creación de una aplicación con una estructura lo más general posible, de ahí que los problemas presentados por la utilización de sockets fueron:

- Necesidad de crear un diagrama de estados para modelar la máquina de recepción y transmisión. Es decir es necesario interpretar los datos enviados y recibidos a través del socket. Habría que crear un protocolo de comunicaciones propio.
- Coherencia del Diseño. Se ha considerado que prevalece la coherencia del diseño, para futuras ampliaciones, por ejemplo, más que la velocidad de la transmisión.

La definición de un protocolo propio no es un problema desde el punto de vista de la implementación sino de la estrategia de diseño. Como se ha visto, existen muchos estándares y planteamientos diferentes para la transmisión de ECG, por tanto se está buscando un diseño lo más general posible, para la ampliación a nuevos estándares. La utilización de técnicas estándar para la comunicación CS, frente al uso de una técnica propia, hará más fácil la utilización de la aplicación en futuros desarrollos.

Por lo tanto se optará por una técnica de objetos distribuidos para la comunicación entre cliente y servidor. Se ha escogido la invocación de métodos remotos (RMI) frente al empleo de sockets [68] por las ventajas que aporta al diseño la utilización de objetos remotos. Frente a las alternativas de CORBA y DCOM, RMI en el ámbito de esta aplicación presenta las siguientes ventajas:

- Es la técnica más sencilla de implementar para una aplicación de tamaño medio como la que aquí se propone.
- Es menos costosa en la utilización de recursos (*lightweight*).
- Orientada a la aplicación en *Java*. Todas las clases se desarrollarán en Java 1.1.
- La característica multi-plataforma viene dada por el applet integrado en la página HTML, por ello no se necesita una arquitectura específica como en CORBA y DCOM.

Estas razones justifican el empleo de RMI en el ámbito de esta aplicación. Otro objetivo de diseño ha sido la ampliación a otros estándares de la manera menos costosa posible, es decir que sea rápida y que se adapte de una manera sencilla al diseño. En la figura 3.7 se muestra el diagrama general de clases y cómo es transparente el protocolo en el cliente y específico en el servidor.

Por protocolo transparente se entiende que la comunicación entre esas clases es independiente del estándar empleado para almacenar el archivo ECG. Para cada estándar se desarrollará una clase que lea ese formato, por ello, en el lado del servidor el protocolo será específico a cada uno de ellos. La mayor parte del procesado sobre el formato de datos escogido se realizará en el servidor. En los casos en los que se emplee compresión, el cliente tendrá disponibles las clases que tengan los métodos para descomprimir la señal. Esto se mostrará en el diagrama de clases.

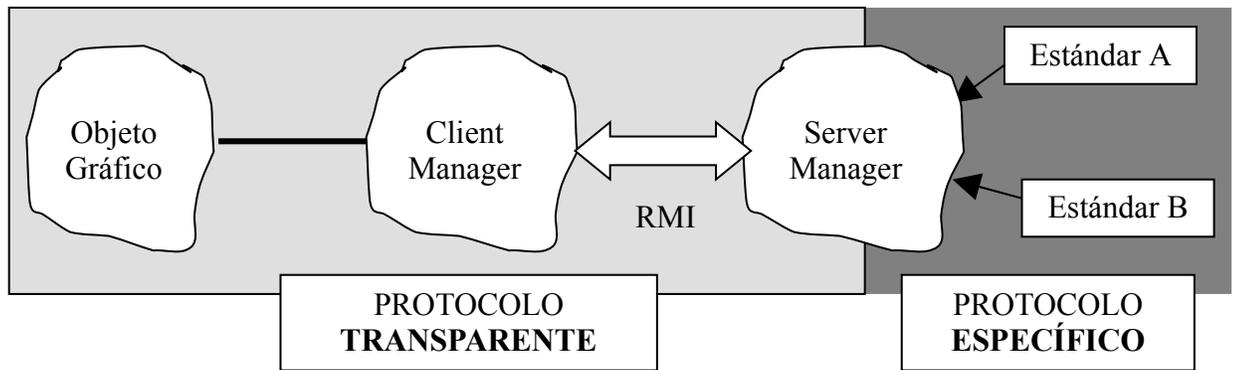


Figura 3. 7 Diagrama general de clases y ámbito del estándar

La importancia de este diseño radica en la necesidad de crear la mayor abstracción posible para permitir una ampliación futura a otros estándares y un fácil mantenimiento y reutilización de las clases. La propuesta se centra en la mayor transparencia posible para la anotación de registros ECG.

3.2.2. DISEÑO DE CLASES

En la página siguiente se muestra el diagrama definitivo de clases (Figura 3.8). Este diagrama es el resultado de una evolución en el diseño y es el que se ha empleado en el desarrollo final. Se utiliza la nomenclatura de Booch [69]. Las líneas punteadas son clases, las flechas son relaciones de herencia y la relación de tipo *uses* indica que se utilizan métodos de esa clase. La comunicación entre objetos de estas clases posibilita todas las operaciones necesarias para la transmisión de ECG por Internet, según los estándares implementados.

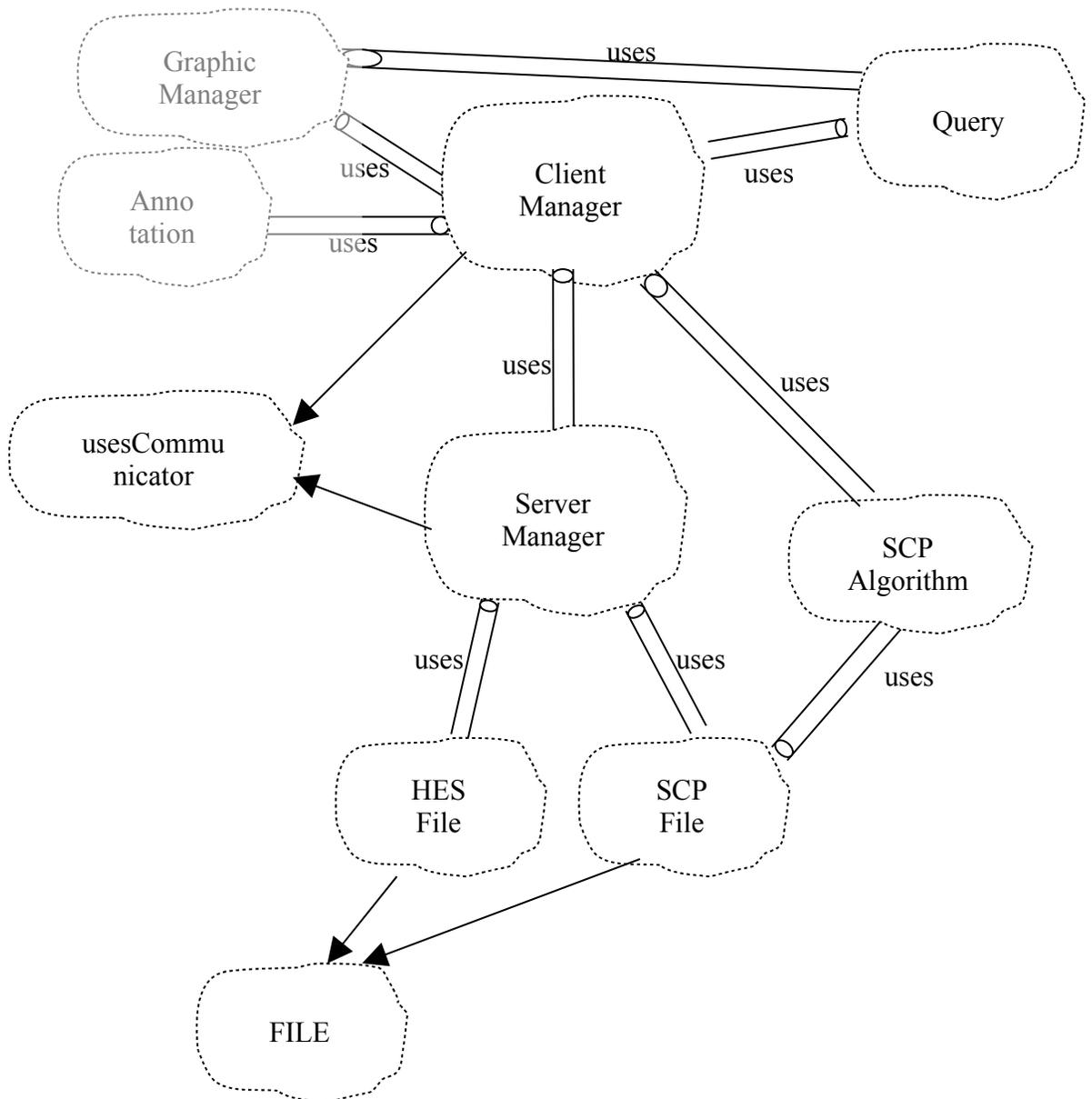


Figura 3. 8 Diagrama general de clases.

La clase GraphicManager modela el interfaz de usuario (visualizador) que utiliza las demás clases, y la clase Annotation representa una anotación según tipo y muestra de la señal en la que se realiza. Estas dos clases no han sido desarrolladas en este trabajo y se han puesto en el diagrama como semi-transparentes para indicar que provienen del anotador. El apartado 3.3 muestra un ejemplo de funcionamiento.

En el código original se encuentra explicado cada método y el ámbito de cada clase. Aquí se presentará el funcionamiento y el objetivo general de cada una:

- **Graphic Manager.** Modela el visualizador y la interfaz con el usuario. Pertenece al applet que utiliza el paquete de clases implementado.
- **ClientManager.** Es el objeto encargado de ofrecer los datos a **GraphicManager**. Implementa los métodos remotos de la interfaz **usesCommunicator**, aquí están definidos los métodos que se pueden invocar remotamente. La inicialización de la conexión, por ejemplo, el ID del archivo y el estándar empleado, se llevan a cabo por un objeto **Query** como respuesta a una consulta.
- **Query.** Un objeto de esta clase procesa las consultas y determina sobre qué archivo se va a trabajar. Es un objeto creado ex profeso para la futura ampliación. Se han creado métodos muy sencillos y se explica más adelante cómo añadir métodos a esta clase.
- **ServerManager.** Es el objeto que implementa los métodos remotos. Está en contacto con los archivos y sirve los datos tanto a **ClientManager** como a **Query**. Utiliza las clases encargadas de leer los estándares utilizados.
- **SCPFile, HESFile, etc...** Son las clases específicas para los estándares utilizados. En este caso existe también una clase **SCPAlgorithm** que tiene los métodos encargados de la descompresión (en el cliente) de los datos enviados. Se puede añadir cualquier formato de datos mientras se implementen los métodos de la interfaz **usesCommunicator**.
- **File** es una clase específica de Java.

Como se puede observar del diagrama, faltaría una clase denominada *usesCommunicator* que es la clase añadida para aplicar RMI. En realidad es una interfaz que especifica los métodos que se pueden invocar remotamente. Estos métodos se implementan en la clase **ServerManager** y son llamados por **ClientManager**.

Una pregunta que surge en esta etapa de diseño sería ¿para qué existe la clase ClientManager si puedo acceder a los métodos de ServerManager desde, por ejemplo, GraphicManager? Las razones son varias:

- Coherencia con el diseño original. Se mantiene la relación entre el visualizador y la comunicación remota a través de ClientManager.
- Procesado Local. En los casos en los que se transmite la data comprimida, ClientManager tiene que encargarse de hacer ese procesado.
- Consultas. Se gestionan las consultas a través de ClientManager, así la clase Query no se comunica con ServerManager, y cualquier modificación en el diseño será independiente de la clase que modela el servidor.

El lector de SCP, que son las clases SCPFile y SCPAlgorithm, proveen métodos para la lectura del formato de datos SCP, así como el procesado de señal necesario para la descompresión de los datos y la anotación. Esta clase será de acceso público y es el primer lector de este estándar que se hace público, y que está programado en lenguaje Java. La mayoría de los métodos implementados no se han utilizado porque el visualizador-annotador no está preparado para utilizar todas las potencialidades de la librería de clases (paquete). Para la evaluación del lector se han utilizado archivos en formato SCP enviados por investigadores asociados al Proyecto SCP, concretamente el Prof. Rubel y el Dr. Zwietz (LISI-INSA, Francia, y la Escuela Médica de Hannover, Alemania). Algunos de estos archivos pertenecen a los registros de prueba de los algoritmos de compresión y codificación del SCP.

3.3. EVALUACIÓN Y MODIFICACIÓN

A la hora de la realización práctica de estas clases se ha utilizado la herramienta JDK (*Java Development Kit*, Kit de Desarrollo en Java) en su versión 1.1. Esta herramienta es suministrada gratuitamente a través de Internet por Sun Microsystems. Otras herramientas comerciales son Microsoft J++, Symantec Café, Code Warrior,

etc, y son más elaboradas porque tienen una interfaz integrada para el desarrollo de la aplicación, compilación, interpretación, creación de proyectos, etc...

La información acerca del lenguaje se ha hecho a través de [62] [70] [23] [71] y [72]. En esta documentación se pueden encontrar las especificaciones oficiales del lenguaje, tutoriales y material de apoyo.

Las características principales de este lenguaje son:

- Es orientado a Objetos.
- Es independiente de la plataforma. El código fuente es el mismo para cualquier plataforma.
- Orientado a Internet y a programación en redes bajo TCP/IP.
- Disponible gratuitamente y con mucha información pública.

Para ver la programación en Java de las clases presentadas se puede acceder al código público, aquí no se ha pretendido hacer una descripción de cómo se ha hecho la aplicación en Java. Sin embargo, algunos métodos de ejemplo serían:

- *public boolean initECGFile(String ID, String Protocol)*. Método de un objeto *ServerManager* que inicializa el archivo sobre el que se realizará la anotación y el protocolo utilizado. Regresa *False* si la operación no tuvo éxito.
- *public int getNumLeads()* . Método de un objeto *SCPFile* que regresa el número de canales del archivo SCP al que hace referencia.
- *public int getSI(String type)* . Método de *ServerManager* que regresa el valor del intervalo de muestreo utilizado por el archivo ECG sobre el que se está trabajando. Se accede remotamente desde *ClientManager*.
- *public int[] getSamples(int lead, String Type)* Definición abstracta (sin implementación) en la interfaz *usesCommunicator*. Es un método que tiene que implementarse en *ServerManager* para que pueda ser accedido desde *ClientManager*.

- `public int [] getSamplesHES(int lead)` Regresa las muestras del canal `lead` en un archivo con formato HES²

3.3.1. EJEMPLO DE FUNCIONAMIENTO

Integrando la aplicación con el visualizador-annotador son 13 clases con más de 3000 líneas de código. En la figura 3.9 se muestra la pantalla (applet) y una pequeña explicación.

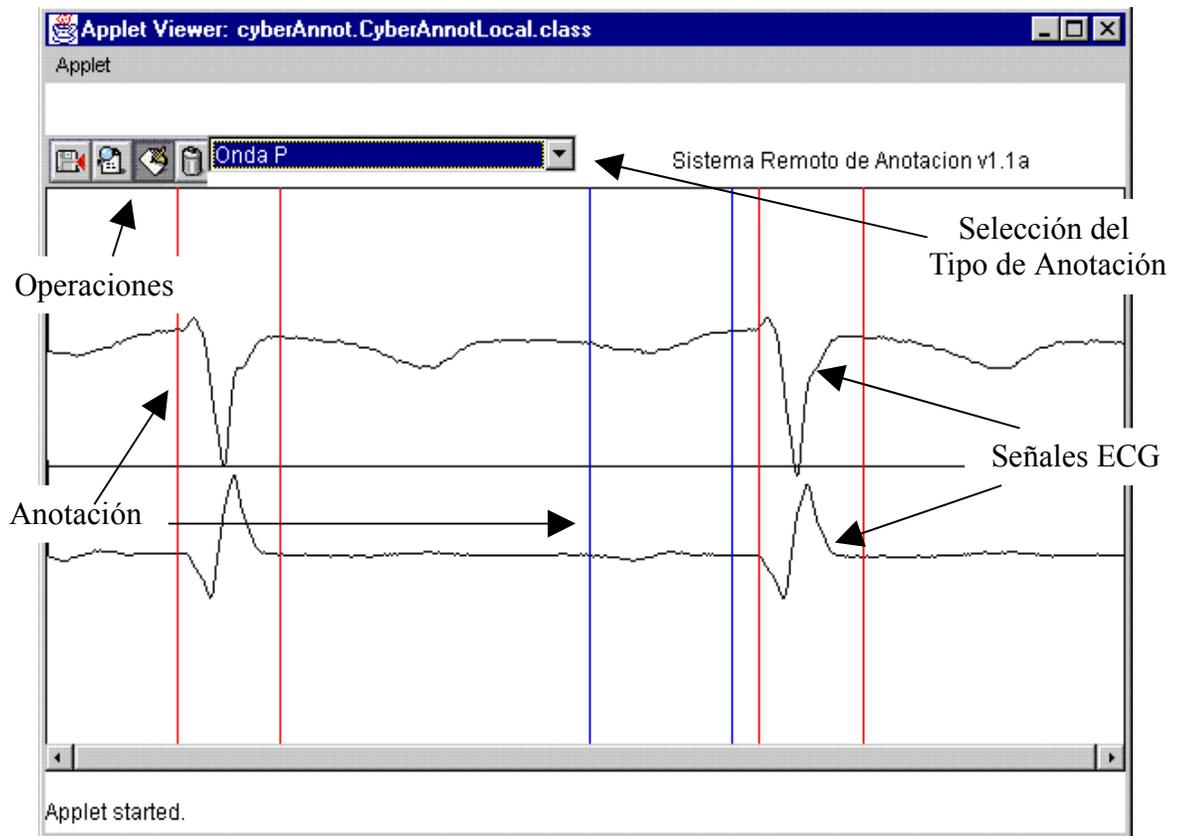


Figura 3. 9 Visualizador Remoto de ECGs

2 El formato HES es un formato binario típico, con ocho canales de adquisición, 10 segundos de duración, etc... Los archivos son de una base de datos europea, enviados por el Departamento de Informática médica del Hospital de Hannover (Alemania).

El applet se carga desde *appletviewer*, que es una herramienta de JDK para visualizar applets. Del mismo modo se cargaría desde un browser convencional. Se muestran dos canales del registro pd2010 suministrado por el Medical Science Informatics de Hannover. Este registro se encuentra en formato binario HES y en formato SCP y se puede leer en cualquiera de ellos. Algunas de las variables, como el número de canales a visualizar, el tamaño de la pantalla, el archivo a visualizar, etc., se han pasado al applet por parámetro, pero una vez desarrollada la interfaz de usuario se podrán acceder a los métodos desarrollados en la librería de clases.

Las operaciones posibles sobre las anotaciones son: lectura desde archivo, borrar y seleccionar tipo. Para realizar las marcas de anotación se utiliza el ratón de la computadora. Para entender mejor cómo se comunican los objetos y qué métodos se utilizan en un funcionamiento normal del visualizador mostramos el procedimiento típico:

- Se inicializa el applet (método *init()*). Crea un objeto *ClientManager* que se conecta con un *host* especificado (*new ClientManager(String host)*).
- Se inicializa la conexión entre *ClientManager* y *ServerManager*. Se cargan tablas con los archivos y protocolos disponibles (métodos *availableID()*, *availableFiles()*). Con ello el objeto *ClientManager* sabe qué estándares y archivos se pueden leer.
- Se finaliza la inicialización del applet y se espera a que se solicite un registro ECG.
- Se crea una consulta. Se crea un objeto *Query* con el tipo de consulta y el parámetro (*new Query(String met, String param)*). Un ejemplo sería *Query("NameFile", "file001.fil")*.
- Se ejecuta una consulta. Se llama al método *execute()* del objeto *Query* creado. Este método regresa una cadena de caracteres con el ID del archivo y el protocolo que usa.
- Con esta información el objeto de la clase *ClientManager* configura la conexión con el método *void setFile (String ID, String prot)*. Este es el

archivo con el que se trabajará hasta una nueva configuración. También se cargan las anotaciones, si las hay (*Vector leeAnnot()*).

- A petición del usuario, se leería el número de canales disponibles (*int [] InfoLead()*) o las muestras del canal X (*int [] getSamples (int X)*), etc...
- Con las muestras visualizadas se puede recorrer la señal, hacer zoom, anotar con diferentes tipos de anotación, etc...
- Para grabar las anotaciones se selecciona dicha operación en el panel de control, y se ejecutará el método *saveAnnot(Vector annotations)* de *ClientManager*.
- (...) Se pueden realizar otras consultas y operaciones.

Aunque algunas de las posibilidades no están implementadas en la interfaz de usuario, los métodos han sido evaluados independientemente para asegurar su funcionamiento. Esto se puede hacer gracias a la metodología orientada a objetos.

3.3.2. NUEVOS ESTÁNDARES Y MÉTODOS DE CONSULTA.

Se pretende ofrecer un diseño para que se sigan implementando clases y que se realice su integración con herramientas que utilicen la transmisión de ECG por Internet. Algunas de las ampliaciones obvias son: añadir estándares y métodos de consulta.

Para añadir un nuevo estándar a la aplicación la mayor parte del trabajo es el desarrollo de la clase que modele el formato de archivo que se ha utilizado y el acceso a los datos que contiene. Debido al diseño orientado a objetos el desarrollo de una nueva clase no altera el diseño global del sistema y puede hacerse por equipos de trabajo locales o por grupos de investigación del exterior. En este trabajo se han diseñado y desarrollado dos clases, *SCPFile* y *HESFile*. La primera es un lector de archivos en formato SCP-ECG y la segunda es un ejemplo del tipo de archivos binarios más comunes en las bases de datos internacionales. Se podrían hacer clases para la base de datos MIT-BIH, IMPROVE, etc... Por ejemplo la clase *SCPFile*

implementa métodos para leer el formato del estándar SCP (más de 70), los métodos “obligatorios” están impuestos por la interfaz *usesCommunicator* que es la que determina qué información se puede pasar desde el servidor al cliente. Es decir, se tienen que implementar métodos que permitan a *ServerManager* leer esta información del estándar a desarrollar.

Por ejemplo, para la implementación de la clase *HESFile*, que lee un formato binario muy sencillo, se han implementado, entre otros, los métodos:

- *int [] getSamplesHES(int lead)*. Regresa los datos en formato binario, del canal especificado en *lead*.
- *Hashtable HESleadsId()* Tabla con los códigos de los canales adquiridos.
- *public int [] getAdquDate(String type)*. Regresa la fecha de adquisición

Estos métodos son empleados por *ServerManager* para ejecutar *getInfoLead()*, *getSamples(int lead)* o *getAdquDate()*, que están especificados por *usesCommunicator* y que un objeto *ClientManager* puede leer remotamente. Esta implementación es sencilla porque el formato HES lo es, en el caso de SCP el procesado de señal necesario es mucho mayor y de ahí la mayor dificultad de implementación de *SCPFile* frente a *HESFile*. De cualquier modo, se demuestra que la inclusión de un nuevo estándar se hace de una manera rápida y sencilla.

La definición de nuevos métodos de búsqueda de información, denominados aquí consultas, se haría dentro de la clase *Query*. El método sólo puede utilizar métodos disponibles en el objeto *ClientManager* asociado. Los métodos que se pueden consultar remotamente están definidos por *usesCommunicator*, y con esta información se realiza la búsqueda. Al ejecutar la consulta (método *execute()*) se llama al método adecuado a los campos *Param* y *Method*. El método *execute()* regresa un arreglo de *String*, con el ID y el estándar utilizado, que es la información necesaria para establecer la comunicación entre *ClientManager* y *ServerManager*.

Una implementación más eficiente permitiría a Query comunicarse directamente con el servidor, e incluso en el futuro tener una interfaz con una base de datos relacional. En este trabajo se ha pensado en que sea un objeto que trabaje independiente del estándar utilizado, por ello simplemente se utiliza a ClientManager. Además es utilizado solamente por el objeto gráfico que así obtiene la información de la consulta. El nuevo método se añade en la clase Query.

IV. CONCLUSIONES Y RECOMENDACIONES

Este capítulo presenta una visión en conjunto de las potencialidades de la aplicación que ha sido desarrollada, presenta las conclusiones y también propone futuras versiones y recomendaciones para la investigación en este campo. También se citan los puntos débiles a resolver dentro de la aplicación, fundamentalmente en el lector de SCP.

4.1. CONCLUSIONES.

En este trabajo se ha discutido acerca del problema de la aplicación de un estándar en la transmisión de ECG a través de Internet y acerca de las limitaciones que existen aún para el uso de la Telemedicina. Los factores limitantes han sido de tipo tecnológico y humano, así como la carencia de un estándar de transmisión aceptado. Como finalidad de este trabajo se ha realizado una aplicación cliente-servidor que transmite ECG a través de Internet como un intento de superar estos problemas.

La transmisión por Internet permite utilizar una infraestructura de hardware y software que ya es un estándar mundial “de facto”. Esto permite que la aplicación sea de bajo costo y que se pueda acceder a un registro ECG con un navegador desde cualquier parte del mundo, a través de un ISP. Esta aplicación es una de las primeras en intentar la transmisión de ECG por Internet, planteando la posibilidad de uso de estándares mundiales de transmisión.

El diseño utilizado (aplicando el DOO) permite la ampliación a otros estándares y la colaboración de otros desarrolladores para mejorar la aplicación. El código desarrollado está disponible públicamente y está programado en Java que es un

lenguaje de uso común en Internet. Se ha implementado una arquitectura cliente-servidor donde la comunicación usa una de las técnicas estándar para la creación de objetos remotos (RMI), por lo tanto, al no utilizar un protocolo de comunicaciones propio, se puede generalizar su uso. Estas clases se han integrado con un visualizador-annotador remoto, para demostrar la facilidad de uso del paquete Java desarrollado.

Una de las clases que se ha implementado es un lector de archivos con el formato especificado por la norma SCP-ECG. En la actualidad no existe ningún lector público disponible de este formato de registro ECG.

Como puntos débiles de este trabajo se puede mencionar que el lector SCP ha de ser comprobado por los miembros del Proyecto OEDIPE para obtener una certificación de que su funcionamiento es 100% SCP. También se debería tener un visualizador que utilice todos los campos y posibilidades del SCP para comprobar su funcionamiento. Esto implica el desarrollo exhaustivo de una interfaz de usuario.

Las pruebas se han realizado con los tipos de anotaciones implementados por el visualizador, mientras que SCP tiene un diccionario de anotaciones muy completo y una sintaxis compleja que está todavía sin implementar .

En cuanto al diseño, se pueden desarrollar algunos más eficientes, por ejemplo el desarrollo de la clase Query en el servidor, la ampliación de los criterios de búsqueda de archivos o el acceso a una base de datos.

4.2. FUTURAS VERSIONES

Una vez que se han visto las potencialidades y puntos a mejorar de la aplicación, se pueden hacer recomendaciones para trabajos futuros en esta área, que tomen esta aplicación como referencia. Algunas de las características que podrían estar disponibles en futuras versiones de la aplicación que ha sido implementada serían:

- Eficiencia del Código. Mayor integración del código y eficiencia de los métodos
- Visualizador-Anotador. Mejorar el desempeño del visualizador-anotador, u otra aplicación que use el paquete de clases desarrollado para realizar una mejor evaluación.
- Interfaz Remota. Utilización de versiones superiores de Java (ya existe una versión beta de JDK 1.2) para aprovechar mejores características de RMI.
- Emplear CORBA y DCOM para la implementación de los objetos.
- Nuevas clases para ampliar los estándares utilizados.
- Mejorar la implementación de las consultas.

El código está público para que otros desarrolladores colaboren en la creación de una mejor aplicación. Este ha sido uno de los objetivos generales de trabajo.

Finalmente, algunas recomendaciones de cara al futuro de esta aplicación dentro de la Telemedicina serían:

- Integración con aplicaciones en tiempo real, dispositivos de adquisición, otros visualizadores, etc...
- Actualizar los estándares empleados, y si es necesario, los protocolos de transmisión. Una proposición sería la utilización de HL7 y XML, que es una de las proposiciones actuales para la transmisión de ECG en Internet [MUSE].
- Desarrollar Diccionarios de Anotaciones “a medida” y/o sintaxis apropiadas.

Se ha propuesto una aplicación de Telemedicina que, lógicamente, no es definitiva pero que da una visión general de los inconvenientes que se plantean en un problema tan complejo como este.

Es necesaria la participación de diferentes grupos de investigación y de la industria para mejorar estas aplicaciones y crear estándares de tipo hardware y software para la transmisión de ECG a través de los diferentes canales de comunicación y, específicamente, de Internet.

V. REFERENCIAS BIBLIOGRÁFICAS

- [1] Anónimo. Sistema de monitoreo de ECG con Java 1.1 <http://www.interbit.co.il/success/> Israel 1997
- [2] Anónimo. Proyecto Telemed <http://www.acLlanL.gov/Telemed> USA 1996
- [3] Varios Autores. Avances Recientes en Bioingeniería. Investigación y Tecnología Aplicada. Sociedad Venezolana de Métodos Numéricos. pp T 1-87. Venezuela 1997
- [4] Tsikinakis M et al. An Integrated Architecture for the Provision of Health Telemeatic Services based on Digital Library Technologies <http://www.ics.forth.gr/~chornaki/diglib/diglib.html> Greece Jun 1997
- [5] Lee HS, et al. Remote Patient Monitoring Service through World Wide Web Proc. IEEE/EMBS Oct 1997. Chicago USA
- [6] Bloom D et al. VIEWPOINT: The Acceptability of Telemedicine Among Health-Care Providers and Rural Patients. <http://www.recoletos.es/dm/telemedicina/deinternet.htm> (*Telemedicine Today*, v.4(3):5-6, 1996)
- [7] Graham, MA. & Leibel, B. Issues Facing the Wide Scale Implementation of Telemedicine. <http://www.va.gov/telemed/human.htm> Center for Public Service Communications USA Dic 1997
- [8] Archila J. et al Avances Recientes en Bioingeniería. Investigación y Tecnología Aplicada. Sociedad Venezolana de Métodos Numéricos. pp T 33-49. Venezuela 1997
- [9] Dakins, DR and Levens S. Multimedia medicine: the basics. http://www.telemedmag.com/db_area/1997/970401co.htm Miller Freeman Inc 1997
- [10] Ochoa M et al. Avances Recientes en Bioingeniería. Investigación y Tecnología Aplicada. Sociedad Venezolana de Métodos Numéricos. pp T 49-65. Venezuela 1997
- [11] Carter S. and Rowberg, Alan. NASA ACTS Mobile Terminal. <http://www.rad.washington.edu/ACTS.html> University of Washington 1995
- [12] Anon. Telemedicine at Mayo Clinic <http://www.mayo.edu/telmed/telmed.html> Mayo Clinic Jul 1997
- [13] An. Telemedicine Systems. <http://www.tvrc.com/medical/medical.htm> TVR Communications Inc. 1997
- [14] Telemedicine Information Exchange <http://www.telemed.org/> Telemedicine Research Center 1996

- [15] Reinig K, et al. The Visible Human Project. <http://www.ncar.ucar.edu/METASCI/CHP.htm> University of Colorado School of Medicine & NCAR Denver USA 1997
- [16] Lehman C and Hayes K. The Interactive Patient. <http://medicus.marshall.edu/medicus/htm> Marshall University School of Medicine 1997
- [17] Telemedicine Project <http://256.106.10.15/pages/projects/projmap.html> Department of Defense USA 1997
- [18] Anon. Telemed Team. <http://www.acl.lanl.gov/> TeleMed Los Alamos national Laboratory CA USA 1998
- [19] Graham, MA. & Leibel, B Telemedicine Strategic Healthcare Group <http://www.va.gov/telemed> USA 1997
- [20] Recursos sobre Telemedicina en Internet. Diario Médico <http://www.recoletos.es/dm/telemedicina>. Madrid España 1998
- [21] Crespo del Arco, J et al. Aplicaciones Médicas de Trabajo Colaborativo en Internet. InforS@alud98 <http://www.conganat.org/seis/inforsaludnet98/comunicaciones/004/Default.htm> España 1998
- [22] Marina Serrano, S. El Manual de Internet. <http://www.elmanual.net/> España 1998.
- [23] Java Development Kit API 1.1 <http://java.sun.com/products/jdk/1.1/index.html> Sun Microsystems Inc 1998.
- [24] Jaquenoud, M. Internet, una herramienta que debemos aprovechar en salud. <http://www.caduceus.com.pe/en-linea/no-01/aprovechar.html> Revista Caduceus On Line Perú 1998.
- [25] Graham, MA. & Leibel, B. The Internet, Intranets and Health Care. <http://www.va.gov/telemed/internet.htm> Center for Public Service Communications USA Dic 1997
- [26] Diario de Divulgación para profesionales del Sector Salud. Diario Médico-Internet Sanitario. <http://www.recoletos.es/dm/>. Madrid España 1998
- [27] Primeras Jornadas de Internet en Salud. InforS@alud98 <http://www.conganat.org/seis/inforsaludnet98/> España 1998
- [28] MUSE Cardiology Management System. <http://www.mei.com/products/prodline.museline.html> Marquette Medical Systems Inc. 1997
- [29] Moody GB.ECG Databases <http://ecg.mit.edu/dbag/evnode2.htm> Harvard-MIT division of Health Sciences and Technology Jul 1998.
- [30] Röscheisen M et al. Shared Web Annotations As A Platform for Third-Party Value-Added Information Providers: Architecture, Protocols and Usage Examples. Technical Report CSDTR/DLTR. Computer Science Department Stanford University 1998
- [31] Anon. Standard Communication Protocols for ECGs: Current Status. <http://www.mei.com/resource/perspectives/protocol>. Marquette Medical Systems Inc 1997

- [32] Quinton Inc. Strategies <http://www.quinton.com/background/andover.html> Quinton Inc 1997
- [33] Zakkaria M.A. "Health Informatics standards" http://www.med-informatik.uni-hildesheim.de/english/mi_faq.html Duke University Medical Center 1996
- [34] Anon. Medical Informatics Standards <http://www.mcis.duke.edu/standards/> Duke Medical Center 1997
- [35] Blair, Jeffrey S. Program Manager. An overview of healthcare information standards. <http://www.cpri.org/docs/overview.html> IBM Healthcare Solutions, Atlanta USA 1997
- [36] HL7. <http://www.mcis.duke.edu/standards/hl7/hl7.htm> Duke Medical Center. 1997.
- [37] Microsoft for Healthcare Users <http://www.mshug.org/index.asp> 1997
- [38] Franklin, D y Ostler, D. The P1073 Medical Information Bus. IEEE Micro Oct 89 pp 52-60.
- [39] P1073 Standard <http://grouper.ieee.org/groups/mib> IEEE 1998
- [40] Willems JL, Rubel P and Zywietz C. Standard Interchange for Computerized Electrocardiography. Progress in Standardization in Health Care Informatics. IOS Press 1993
- [41] Willems JL. (Ed) SCP-ECG project. Standard Communications Protocol for Computerized Electrocardiography. Final Specifications and recommendations, Advanced Informatics in Medicine (AIM #1015) ISBN 90-73402-01-7 Leuven: Acco Publ 1993 Belgium
- [42] Glass M and Costa L. "IEEE 1073 (MIB): Standardized Connectivity for Patient connected Devices". Medical Electronics Issue Sep 1996
- [43] Anon. Current Activities of Selected Healthcare Informatics Standards Organizations (A Compilation) <http://www.va.gov/publ/standard/health/toc.htm> USA 1996
- [44] Anon. HL7 2.3 <http://www.mcis.duke.edu:80/standards/hl7/pubs/version2.3/html/httoc.htm> Duke University Medical Center 1997
- [45] Moore D. Mobile transmittion on Networks <http://www.nvl.virginia.edu/papers/Moore/thesis.html> Virginia Aug 1996
- [46] Salvador, C et al. An implementation of the IEEE Medical Information Bus Standard. IEEE Engineering in Medicine and Biology. Jun 93 pp 81-88
- [47] Figler, A. Medical Device Data Language. IEEE Eng. In Medicine and Biology 9th Annual International Conference 1987.pp 1215-1216
- [48] Norden-Paul, R. E. IEEE Proposed Standard 1073-The Medical Information Bus: an introduction and progress report. IEEE Eng. In Medicine and Biology 9th Annual International Conference 1987.pp 1209-1210
- [49] Soule , Dan T. Selection of a Area Network for Patient Care Management. IEEE Eng. In Medicine and Biology 11th Annual International Conference 1989.pp 1581-1582.

- [50] Rubel P et al. OEDIPE. Open European Data Interchange and Processing for Electrocardiography. Health in the New Communications Age. pp 313-321 M. F. Laires et al. IOS Press 1995
- [51] Willems JL A universal Set of Statement Codes for the Exchange and Storage of Diagnostic ECG Interpretation Results. Computers in Cardiology IEEE Computer Society 1992 pp 101-104
- [52] Al-Ahmad W, Willems JL and Rubel P. ECG Data Interchange within the Framework of the SCP-ECG and the OEDIPE Projects. Computers in Cardiology IEEE Computer Society, Los Alamitos (USA) 1994 pp 337-340
- [53] OEDIPE Project. Experimental European Network of Distributed Cooperative ECG Databases. AIM #2026 Deliverable 12 (Public Report) Nov 94
- [54] Nelson, M The Compression Data Book. Ed M&H Books CA USA 1991
- [55] Cox JR et al. AZTEC: a preprocessing program for real-time ECG rhythm analysis. IEEE Trans Biomed. Eng. BME 15 pp 128-29 1968
- [56] Hamilton, PS and Tompkins WJ. Compression of the ambulatory ECG by average beat subtraction and residual differencing. IEEE Trans Biomed. Eng. BME 38 (3) pp 253-59 1991
- [57] Zywietz C et al. Digital ECG Transmission from Ambulance Cars with Application of the European Standard Communications Protocol SCP-ECG Computers in Cardiology IEEE Computer Society, Los Alamitos (USA) 1994 pp 341-344
- [58] Tompkins WJ Biomedical Digital Signal Processing. pp 193-216 Ed Prentice Hall. New Jersey 1993
- [59] Annon. Java and Databases. <http://discussions.developer.com/HyperNews/get/forums/JavaDatabase.html>. 1998 EarthWeb Inc
- [60] Mor Y. Do it in the Java Way!. <http://discussions.developer.com/HyperNews/get/forums/3/4.html>. 1998 EarthWeb Inc
- [61] Campione M. and Walrath K. All About Sockets. <http://java.sun.com/docs/books/tutorial/networking/sockets/index.html>. Sun Microsystems Inc. Dic 1997
- [62] Campione M. and Walrath K. The Java Tutorial. Object-Oriented Programming for the Internet. <http://java.sun.com/docs/books/tutorial/> Sun Microsystems Inc. Dic 1997
- [63] Albertson T. Best practices in distributed object application development: RMI, CORBA and DCOM (Part 1 of 4) http://www.developer.com/news/teachfocus/022398_dist1.html Earth Inc 1998.
- [64] Albertson T. Distributed Object Application Development: The Java-RMI Solution (Part 2 of 4) http://www.developer.com/news/teachfocus/022398_dist2.html Earth Inc 1998.

- [65] Albertson T. Distributed object application development: The Java-CORBA solution (Part 3 of 4) http://www.developer.com/news/teachfocus/022398_dist3.html Earth Inc 1998
- [66] Albertson T. Distributed object application development: The DCOM solution (Part 4 of 4) http://www.developer.com/news/teachfocus/022398_dist4.html Earth Inc 1998
- [67] Anon. Java Remote Method Invocation <http://java.sun.com/products/jdk/rmi/index.html> Sun Microsystems Inc. 1998.
- [68] Anon. Socket vs. RMI <http://www.developer.com/classroom/experts/answer140.html> Earth Web Inc
- [69] Booch, G. Object Oriented Design with Applications The B. J Publishing Company. CA USA.1991
- [70] CD-ROM. Java Developer's Companion. From the source. Sun Microsystems Inc 1997.
- [71] Cuenca J., PM Programación en Java para Internet. Ed Anaya Multimedia 1996
- [72] Winters P. et al. Aprendiendo Visual J++ en 21 días. Ed Prentice Hall Mexico 1996

VI. LISTA DE ABREVIATURAS

ACTS: Tecnología para la comunicación por satélite.

ADT.: Admisión, Transferencia y Alta de pacientes.

ANSI: Organización Norteamericana que estudia diferentes estándares.

BCC: Controlador de cama (P1073)

BD: Base de datos.

CEN: Comisión Europea de Normativización.

CGI: Interfaz Común de Paso. Mecanismo para lanzar programas desde una página HTML

CRC: Código de Redundancia Cíclica.

CS: Cliente-Servidor.

DCC: Controlador de dispositivo (P1073)

DCOM: Modelo de Objetos Distribuidos propuesto por Microsoft.

ECG: Electrocardiograma.

FV: Fibrilación Ventricular.

GE: General Electric.

HL7: Estándar de informática médica.

HTML: Lenguaje estándar para el formato de páginas Web

HTTP: Protocolo de transmisión de páginas Web.

IEEE: Instituto de Ingenieros Eléctricos y Electrónicos.

ISO: Organización Internacional de estandarización.

ISP: Proveedor de Servicios de Internet.

JAHIS: Institución Japonesa de Salud.

JPL: Laboratorio de Propulsión

LAN: Red de Área Local.

MDDL: Lenguaje especificado por el estándar 1073..

MIB: Bus para el intercambio de información médica.

MSHUG: Grupo de Trabajo sobre las aplicaciones de salud en ambiente Microsoft.

NASA: Agencia Aeroespacial Norteamericana

OEDIPE: Proyecto Europeo para el intercambio de registros ECG.

OSI: Modelo de capas de red.

QRS: Uno de los componentes del ECG.

RMI: Invocación de Objetos Remotos.

SCP-ECG: Norma europea para el intercambio de ECG.

TCP/IP: Protocolo de la capa de transporte de Internet.

UCI: Unidad de Cuidados Intensivos.

USA: Estados Unidos de América.

VA: Administración de Veteranos de guerra de USA.

WAN: Red de Área Extendida.

VII. APÉNDICE A

El objetivo de este Apéndice es ofrecer documentación más detallada acerca de la Norma SCP. Debido a que no existe ninguna publicación de acceso público por parte de grupos de investigación que no están dentro del proyecto OEDIPE, se ha hecho en este trabajo un aporte a la divulgación de esta norma. Se presenta aquí con un poco más de detalle el procesado de la señal, así como del formato de datos. Toda la información está extraída de la documentación oficial del año 1993.

VII.1. MECANISMOS DE COMPRESIÓN Y DESCOMPRESIÓN

Uno de los puntos de más trabajo para el grupo SCP-ECG fue el de la compresión de los datos. Por una parte surge la necesidad de comprimir los datos para mejorar la eficiencia del canal de comunicaciones (cualquiera que sea), i.e., mejorar la velocidad de transmisión, y por otra parte las capacidades limitadas de almacenamiento. El resultado ha sido un mecanismo de codificación que es un procesado de señal que involucra filtraje, algoritmos de diferencias, diezmado y compresión Huffman. Este esquema obtiene altas tasas de compresión.

A.1.1. ESPECIFICACIONES PARA LA COMPRESIÓN DE DATOS.

Debido a la problemática planteada, en realidad existen diferentes esquemas de compresión. Como resultado del trabajo del grupo SCP-ECG se obtuvieron las especificaciones que debe cumplir el mecanismo de codificación y compresión utilizado. La norma SCP-ECG especifica un mecanismo de compresión y los requisitos sobre los datos comprimidos y codificados.

En el apartado siguiente se presenta el mecanismo de compresión de la norma SCP-ECG. Aquí se pretende plantear brevemente la necesidad de tener unas especificaciones de compresión y cuales son en este caso.

Todas estas consideraciones se han concretado en la propuesta de unas especificaciones para la señal original y la recuperada. El mecanismo de compresión utilizado calcula una señal promedio para todo el registro y se obtiene de ella una señal residuo, como se explica en el apartado siguiente. Como resultado de las investigaciones se proponen las siguientes especificaciones:

(SR: Frecuencia de Muestreo. LSB: Bit Menos Significante. RMS: Valor Cuadrático Medio)

ECG grabado con al menos	8 canales simultáneos (I, II, V1..V6)
Digitalización	SR500 Hz; LSB $\leq 5\mu\text{V}$
Señal Promedio	SR500 Hz; LSB = $5\mu\text{V}$
Señal Residuo	Error Truncamiento $\leq \pm 15\mu\text{V}$ Intervalo de Muestreo $\leq 8\text{ms}$
Error de Reconstrucción	RMS $\leq 10\mu\text{V}$
Error Absoluto	$\leq 100\mu\text{V}$ fuera del intervalo P-QRS-T
Error Absoluto en el QRS	$\leq \pm 15\mu\text{V}$

Tabla A. 1 Especificaciones de la norma SCP-ECG para la codificación y compresión de las señales ECG.

En la tabla se puede observar como son requisitos para los dispositivos especialmente (frecuencia de muestreo, intervalo de muestreo y resolución). A continuación se presenta el esquema de compresión propuesto dentro del protocolo.

A.1.2. PRESENTACIÓN DEL ESQUEMA DE COMPRESIÓN-DECOMPRESIÓN EMPLEADO

El esquema aquí planteado cumple las especificaciones de la tabla anterior y permite obtener una tasa de compresión de hasta 20 veces. Como resultado de las investigaciones sobre los métodos de compresión, se extrajeron las siguientes conclusiones, que son características del método propuesto:

- Se ajusta la resolución a $5\mu\text{V}$ (Truncamiento)
- Se calcula una señal promedio del registro original. Esta señal promedio no tiene porque ser la suma promediada de todos los ciclos, sino una señal representativa del registro (puede ser la señal promedio o no).
- Se detectan los complejos QRS del registro y se alinean con el QRS de la señal promedio y se restan. Se obtiene así una señal residuo.
- El ciclo (o señal) representativo se comprime por reducción de redundancia (Huffman)
- La señal residuo se filtra paso bajo, se diezma y se trunca la resolución (escalón de cuantización). En este apartado los complejos QRS están protegidos del diezmo y el truncamiento.
- Se elimina la correlación entre muestras aplicando un algoritmo de diferencias, de 1er o 2º orden. El resultado se codifica por reducción de redundancia.

En realidad la norma SCP-ECG especifica dos esquemas distintos para la compresión, uno denominado reducción de redundancia (*pure reduction redundancy*) y otro de alta compresión (*high compression*).

Por lo tanto el proceso que se sigue desde la señal original hasta la señal que se almacena en el registro SCP sigue un proceso más bien complicado. En la figura A.1 se muestra un diagrama del procedimiento.

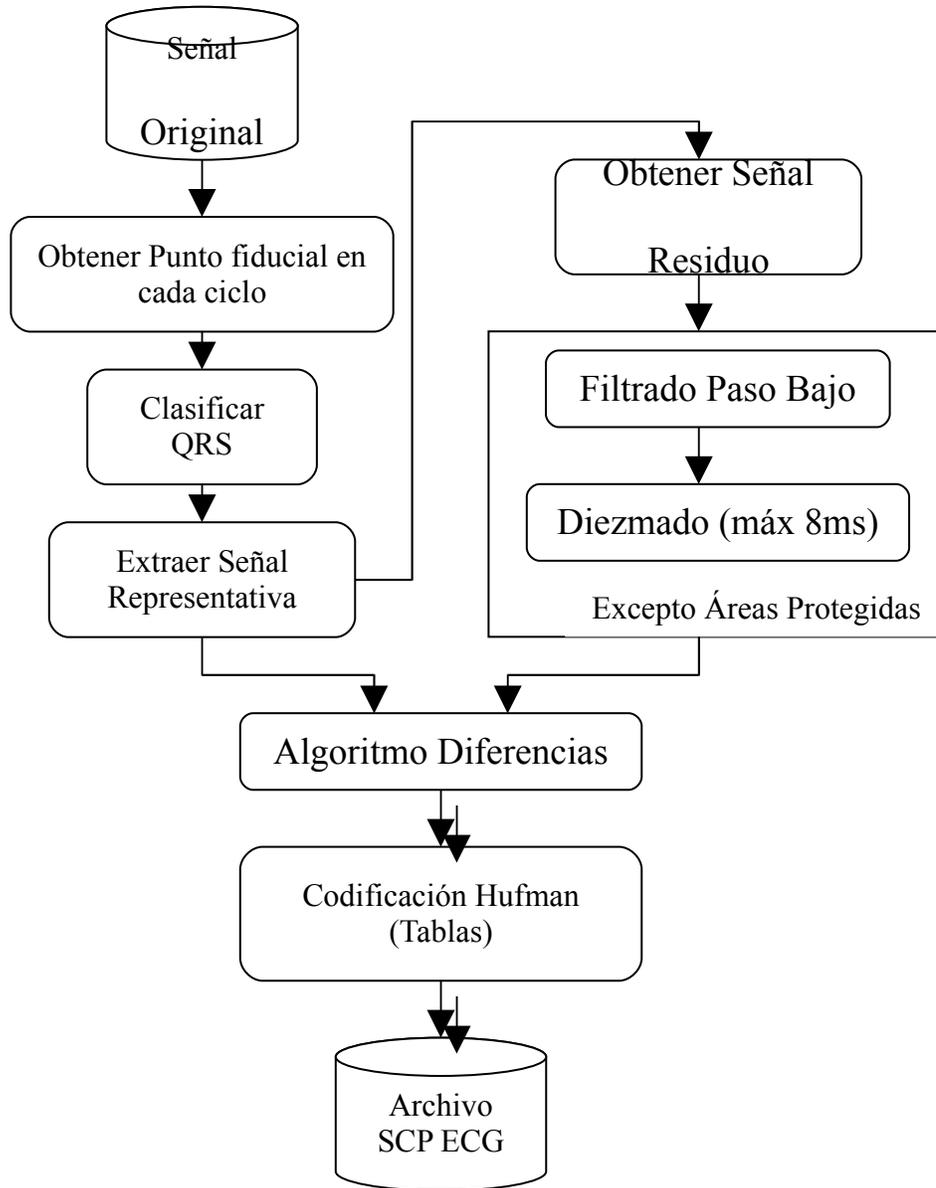


Figura A. 1 Esquema de compresión de la norma SCP- ECG

Este procedimiento no difiere de otros esquemas de compresión con señal promedio, la aportación de este esquema es la protección de los complejos QRS. Además se especifican los márgenes de error permitidos. Estas especificaciones son cuantitativas pero reflejan algunas consideraciones cualitativas del análisis de la señal original y reconstruida. El compromiso se produce entre obtener altas tasas de compresión y tener grandes errores en la reconstrucción de la señal. Un estudio detallado del efecto

de este esquema de compresión sobre 89 registros ECG fue elaborado para determinar las especificaciones de la tabla anterior. La tasa promedio de compresión fue de 21.5 y la media del error cuadrático medio fue de 10.8 μV . Además los registros de prueba contenían casos con latidos ectópicos, sístoles polimórficas, fibrilación auricular y otras cardiopatías, que representaban la alta variabilidad de los registros ECG.

Para recuperar la data comprimida el proceso es el inverso, a continuación detallamos el proceso a seguir:

- Se decodifican las muestras según las tablas Huffman
- Se recupera la data que estaba procesada según los algoritmos en diferencias
- Se interpolan las muestras que fueron diezmadas (señal residuo).
- Se filtra pasa bajo la señal residuo.
- Se multiplica por AVM (Amplitud Value Modifier, valor modificador de amplitud, es decir, las unidades de amplitud).
- Si el esquema utilizado es el de alta compresión se recupera la señal original sumándole la señal promedio a la señal residuo.

Todos los datos necesarios se encuentran en el formato del archivo. El formato de un archivo SCP se presenta más adelante en este mismo capítulo. A continuación vamos a mostrar unos ejemplos de compresión de datos.

A.1.3. EJEMPLOS DE COMPRESIÓN Y EXPLICACIÓN DEL ESQUEMA

A continuación se van a presentar unos ejemplos de compresión de los datos. En la tabla A.2 se muestran el número de muestras y varias columnas para seguir los diferentes pasos. La columna ORI muestra la muestra original, TRU la muestra truncada, RES la señal residuo, FIL la señal filtrada, DEC el diezmado, 2D el resultado del algoritmo en diferencias y HUF los bits resultantes de aplicar una tabla Huffman ficticia, en este caso. Además las especificaciones son: intervalo de muestreo 2 ms, resolución de 1 μV a 5 μV por bit menos significativo. El filtro es un “moving average” de 9 muestras y el diezmado es a 8 ms, es decir se toma una de cada cuatro muestras.

Muestra	ORI	TRU	RES	FIL	DEC	2D	HUF
1	63	13	13	13	14	14	111111111000001110
2	70	14	14	14			
3	74	15	15	14			
4	71	14	14	15			

5	79	16	16	16	18	18	111111111000010010
6	89	18	18	17			
7	96	19	19	19			
8	102	20	20	20			
9	108	22	22	21	22	0	0
10	112	22	22	22			
11	114	23	23	22			
12	116	23	23	23			
13	116	23	23	23	22	-4	111101
...

Tabla A. 2 Ejemplo del Mecanismo de Compresión

Se toman las primeras 13 muestras, el mismo procedimiento se hace para todas las muestras, y para todos los canales. En este caso aplicamos todo el mecanismo completo, es decir, son muestras que no están “protegidas”. Para los segmentos protegidos (QRSs) no se realiza el filtraje ni el diezmado. Las especificaciones se toman de los campos correspondientes dentro del archivo SCP.

A.1.3.1.1.1. Truncamiento.

Si la señal original está a $1\mu\text{V}$ de resolución, por ejemplo, se cambia esta resolución (se empeora) a $5\mu\text{V}$ por bit. Esta información está almacenada en el archivo SCP (Sección³ 5, bytes 1 y 2 del archivo SCP). Si se sigue el proceso completo, se ve como esto implicará que los valores estarán más cercanos al cero, que son los valores codificados en la tabla Huffman y por lo tanto se tendrá un flujo de datos con menor número de bits. Para hacer la conversión se usa la siguiente fórmula :

$$TRU = \frac{ORI \pm 2}{5}$$

El signo \pm es debido a que depende de si el dato de entrada es positivo o negativo. Para las tres primeras muestras se tiene $TRU = (63+2)/5 = 13$, $TRU = (70+2)/5 = 14.5 \rightarrow 14$ y $TRU = (74+2)/5 = 15.2 \rightarrow 15$. Cuando hay que redondear se produce un error de cuantificación que producirá un error total entre la señal original y la recuperada. A mayor error, mayor RMS que es el parámetro que se utiliza para ver cuán diferente es la señal original de la recuperada. El concepto del truncamiento de la señal se muestra en la figura siguiente.

³ Aunque no hemos presentado el formato de datos, se pretende mostrar que cada valor está especificado en el archivo.

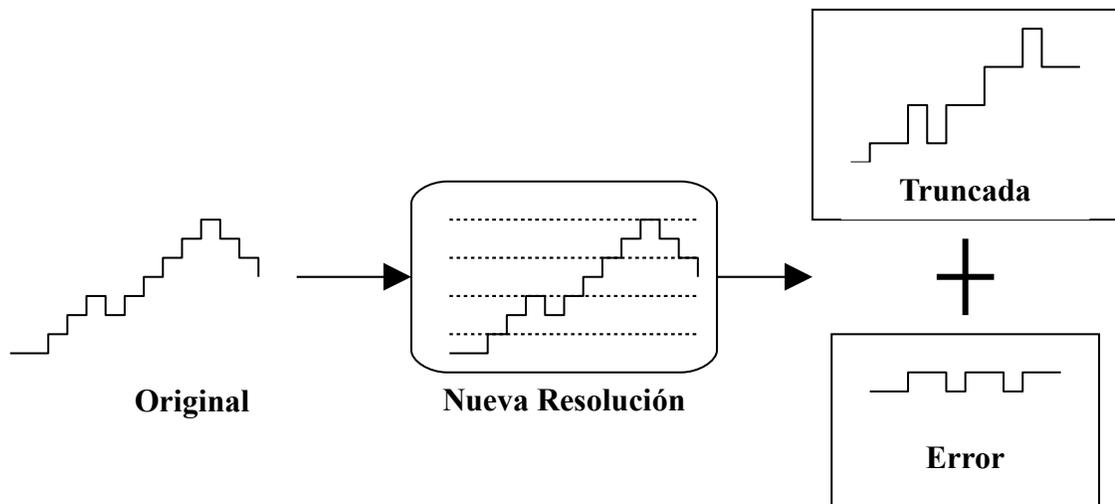


Figura A. 2 Truncamiento de la señal

A.1.3.1.1.2. Señal residuo.

Esta sería la señal que quedaría después de restarle la señal promedio a este dato. Para ello es necesario un alineamiento del ciclo representativo escogido con este ciclo. La alineación entre los ciclos, antes de hacer la sustracción se hace con el punto fiducial, que es el punto central del QRS. Como se supone que todos los ciclos son muy parecidos al ciclo representativo (señal promedio) la resta de uno y otro dará valores cercanos al cero. Es importante decir aquí que los complejos QRS se protegen de la resta ya que contienen información de mucha importancia.

La alineación se hace con la información que está guardada en el archivo SCP. En el archivo se guardan punteros al punto fiducial, duración del ciclo, principio y fin del QRS, etc. Esta sustracción se hace en el caso del esquema de “alta compresión”. Para este ejemplo no se ha utilizado este esquema, ya que lo que se quiere mostrar es el cálculo numérico.

A.1.3.1.1.3. Filtraje y Diezmado.

Sólo se aplica a las áreas no protegidas. Estas áreas están determinadas por los valores QB y QE que son los puntos de inicio y fin del QRS, en cada segmento. Un filtro de promediado (*moving average*) sirve como filtro paso bajo. La fórmula general para este filtro es:

L: número de coeficientes, impar.

$$x(n) = \frac{x(n-(L-1)/2) + \dots + x(n) + \dots + x(n+(L-1)/2) + (L-1)/2}{L}$$

Para la muestra 17, y con un filtro de L=9 coeficientes, $(L-1)/2 = 4$, por lo tanto la ecuación quedaría como:

$$FIL(17) = [RES(17-4) + RES(17-3) + RES(17-2) + RES(17-1) + RES(17) + RES(17+1) + RES(17+2) + RES(17+3) + RES(17+4) + 4] / 9 = 18$$

El diezmado sólo se aplica a las áreas no protegidas también. No se especifica qué algoritmo se va a utilizar para el diezmado y para el algoritmo de interpolación (el caso contrario), aunque se propone un cálculo del valor medio de las muestras. Cada fabricante puede emplear cualquier algoritmo o hacer un diezmado puro, pero al final del proceso de compresión-descompresión debe cumplir los requisitos planteados más arriba. Se puede ver en la tabla anterior como se toma el valor medio de cada 4 muestras. EL intervalo de muestreo se supone que es 2 ms y el máximo tolerado es 8 ms, por lo tanto la razón es de 1 a 4. Si el valor resultante de aplicar el algoritmo no da un entero se redondea.

Por ejemplo, para la muestra 1 el valor medio de la 1, 2, 3 y 4 sería 14, y para la 17 el valor medio de la 17, 18, 19 y 20, sería 13:

$$DEC(1) = [13+14+14+15]/4 = \underline{14}$$

$$DEC(17) = [18+14+12+9]/4 = 13.25 \underline{13}$$

En la tabla se puede observar como de 13 muestras, nos quedan 4, lo que a efectos de compresión ya es un factor importante. Aunque parece una manera “fácil” de comprimir los datos, se debe tener presente que este diezmado no se aplica a todas las muestras de la señal, sino realmente a las muestras con frecuencias menores, es decir con pendientes pequeñas (ondas P y T). Y no se debe de olvidar que la señal comprimida debe cumplir con las especificaciones de fiabilidad de la norma.

A.1.3.1.1.4. Algoritmos en Diferencias.

Por último se aplican los algoritmos de diferencias a las muestras. El objetivo es eliminar la correlación entre las muestras, es decir, información redundante de una muestra sobre la otra. El algoritmo escogido puede ser de primeras o segundas diferencias sucesivas, que corresponden a las ecuaciones mostradas en la tabla siguiente:

Primeras Diferencias	$2D(n) = DEC(n) - DEC(n-1);$ La primera muestra es la original
Segundas Diferencias	$2D(n) = DEC(n-2) - 2*DEC(n-1) + DEC(n)$ Las dos primeras muestras son las originales

Tabla A. 3 Algoritmos en Diferencias

El algoritmo utilizado se almacena en el byte 5 de la sección 5 (señal promedio) y 6 (señal residuo). Sobre el formato de datos se hablará más adelante. Simplemente se pretende reforzar que todas las especificaciones se toman del archivo SCP. En el ejemplo que se está mostrando, se supone que el algoritmo empleado es el de segundas diferencias. Para las muestras 5 y 13, se tiene:

$$2D(5) = 18, \text{ igual que la original porque es la segunda muestra.}$$

$$2D(13) = 14 - 2 * 18 + 22 = 0$$

En la recuperación de la señal se hace con los algoritmos inversos, que se muestran en la tabla siguiente:

Primeras Diferencias	$X(n) = X(n-1) + 2D(n);$ La primera muestra es la original
Segundas Diferencias	$X(n) = 2 * X(n-1) - X(n-2) + 2D(n);$ Las dos primeras muestras son las originales

Tabla A. 4 Algoritmos inversos de la señal en diferencias

A.1.3.1.1.5. Codificación Huffman

La codificación Huffman consiste en la asignación de un flujo de bits para cada valor. La creación de la tabla que asigna bits a valores se puede crear de diferentes formas aunque el resultado siempre será el mismo, se asignará un flujo de bits menor a los valores más probables. Más información sobre este tipo de codificación se puede encontrar en por ejemplo.

No se entrará en detalle de cómo se ha construido la tabla sino en el formato y cómo se realiza la asignación dentro de las especificaciones del estándar. En un archivo ECG los valores más probables son los que están cerca de la línea de base (*baseline*), y todo el esquema que se está presentando lleva a tener los valores más probables cerca del cero. Por lo tanto en un archivo SCP siempre se verán tablas Huffman que codifiquen con menor número de bits los valores cercanos al cero. En el Anexo C del manual de la norma SCP se pueden ver algunos ejemplos de tablas y explicación del procedimiento que se va a explicar aquí.

La codificación Huffman, como ya se dijo, asigna un flujo de bits a un valor determinado. Esto significa que se ha de tener una tabla con todos estos valores. La tabla Huffman que se emplea para codificar el registro ECG se almacena en la sección 2 del archivo SCP. La organización de una tabla Huffman, dentro de la norma SCP, se muestra a continuación:

Número de Estructuras	Número de Bits		Modo Tabla	Valor Base	Prefijo
	Todo el Código	Prefijo			
1	1	1	1	0	0
2	3	3	1	1	100
3	3	3	1	-1	101
4	4	4	1	2	1100
5	4	4	1	-2	1101
6	4	4	0	# (Cambiar a tabla #)	1110

7	12	4	1	0 Ningún Valor	1111
---	----	---	---	----------------	------

Tabla A. 5Ejemplo de Tabla Huffman

La tabla está compuesta por estructuras (esta nomenclatura de estructura, prefijos, valores base, etc, está tomada de las especificaciones SCP), y cada estructura tiene la información para codificar un valor, el llamado valor base, o para decodificar un flujo de bits de entrada. Para ello, cada estructura contiene el número de bits totales del código y del prefijo. Si estos dos valores coinciden (como en las estructuras 1 a 5) el prefijo es el flujo de datos que codifica al valor base. En caso de que no coincidan (estructura 7) se especifican el número de bits que faltan para el código definitivo. Estos bits serían la codificación directa del valor base. En este caso, los 8 bits de la señal original.

Un caso especial es el apartado del “Modo de Tabla”. Este valor especifica si se ha de tomar otra tabla Huffman para la codificación. Entonces, ¿cuántas tablas Huffman existen? Las especificadas en la sección 2 del archivo SCP. La norma SCP permite el empleo de varias tablas Huffman para la codificación de las muestras de la señal.

La justificación se debe a la necesidad de obtener, en algunos casos, una mejor eficiencia en la compresión. Por ejemplo, se puede tener una tabla para valores (-3, 3) y otra para valores entre ± 3 y ± 6 . O para codificar la señal promedio o la señal residuo. El número de tablas depende del dispositivo de adquisición (en realidad la norma permite hasta 65535 tablas). El valor “Modo Tabla” especifica que se debe cambiar a la tabla número #, especificada en el valor base. Cuando esto ocurre se pone el prefijo y la siguiente tabla añadirá el código asignado.

Aunque el estándar no obliga al empleo de ninguna tabla en concreto, se especifica una tabla por defecto (Default SCP Huffman Table) que además es una sola para toda la compresión. La tabla que se especifica por defecto codifica los valores entre [-8,+8], si el valor es mayor se codifican en 8 o 16 bits, es decir como la señal original. Esta tabla está apoyada en los test que se realizaron con varios registros de ECG de la base de datos CSE.

En este caso se pueden dar casos de valores que se almacenan con más bits que los originales (16 bits por muestra), porque si en la tabla se permiten almacenar valores del original más el prefijo, por ejemplo de 4 bits, el resultado sería que la muestra comprimida ocuparía $(16 + 4) = 20$ bits, frente a los 16 bits originales. De hecho en esta tabla una muestra no codificada (> 8 p.ej) se codifica con 26 bits (10 de prefijo mas 16 originales). Obviamente ocurrirá muy pocas veces ya que los valores más probables siempre estarán codificados.

Para concluir se va a tratar de aclarar el empleo de la tabla Huffman con unos ejemplos (tabla A.6 y A.7). El algoritmo de compresión es simple, veamos el pseudocódigo :

*si valor es algún ValorBase y
longCodigo es longPrefijo
bits = prefijo*

si valor es algún Valor Base y
longCodigo > longPrefijo
flujoBits = prefijo
añadir (longCodigo-longPrefijo)
bits de original a flujoBits
si Modo Tabla es 0
cambiar a la tabla indicada en ValorBase

Este programa se repite para todas las estructuras ya que se busca el ValorBase adecuado. En la tabla siguiente se muestra un ejemplo con valores prácticos:

Muestra	Prefijo	Código Resultante
2	1100	1100
1	100	100
2	1100	1100
4	1111	1111 y 00000100
-1	101	101
-5 ⁴	1111	1111 y 11111011
#2	1110	1110 y código en otra tabla
...

Tabla A. 6 Ejemplo de Compresión

El convenio de cambio de tabla vendrá determinado por el dispositivo, por ejemplo, para valores entre 3 y 5 utilizamos otra tabla, el mismo dispositivo emplea la tabla y al comprimir pone los códigos adecuados. Lo importante es que el dispositivo que realiza la descompresión, que no conoce el criterio para el cambio de tabla, pueda recuperar la señal con la información proporcionada en el archivo. En este pequeño ejemplo se ve el efecto de la compresión, ya que para cada muestra de entrada serían 16 bits, y el código equivalente tiene muchos menos. Para la descompresión, el pseudocódigo respectivo es:

agarrar un prefijo
agarrar un bit del byte que entra
si el bit es primero de un código
agarrar otro bit
sino, agarrar el siguiente prefijo
(repetir hasta que se consiga un prefijo)
si ModoTabla es cero, para ese prefijo

4 Los números negativos son en complemento a 2 (Ca2)

ir a la otra tabla
si longPrefijo es longCodigo para ese prefijo
agarrar el ValorBase
sino, agarrar los siguientes (longCodigo-longPrefijo) bits y
agarrar su valor

En la tabla siguiente se recuperará la señal codificada en la tabla anterior, que es 1100100110011110000010010111111011110...

Bit	Bits Acumulados	¿Condiciones?	ValorBase
1	1	No prefijo	
1	11	No prefijo	
0	110	No prefijo	
0	1100	Codigo es Prefijo	2
1	1	No prefijo	
0	10	No prefijo	
0	100	Codigo es Prefijo	1d
...	1100	Codigo es Prefijo	2
...	1111	Codigo es Prefijo	Tomar 8 bits
...	00000100		4
...	101	Codigo es Prefijo	-1
...	1111		Tomar 8 bits
...	11111011		-5
...	1110	Modo Tabla es 0	Cambiar de tabla

Tabla A. 7. Recuperación de data comprimida

Al tener el prefijo para el cambio de tabla, en este caso 1110, se tomarían los 16 bits siguientes y sería el número de la tabla Huffman que habría que seleccionar.

En este apartado se ha mostrado el esquema de compresión y recuperación de datos comprimidos dentro de las especificaciones de la norma SCP. En realidad, el método propuesto no es el único para ser “compatible con SCP”, ya que pueden existir

diferentes métodos que cumplen con las especificaciones. Para ser compatible, el método sólo debe mantener los parámetros de error de la señal recuperada.

Toda la información necesaria para la descompresión de los datos está en el archivo SCP. Ya se ha ido presentando la localización de algunos de los valores empleados. En el apartado siguiente se va a presentar una visión general del formato de datos

VII.2. FORMATO DE LOS DATOS

Ahora se quiere presentar una visión general de lo que es el formato de datos del archivo SCP. Dentro de la definición de un estándar una de las tareas a realizar es especificar el formato de datos. Para estas especificaciones se toman algunas consideraciones generales, que se presentan a continuación.

A.1.4. CONSIDERACIONES GENERALES

La norma SCP está orientada a ECGs de reposo, de 10 segundos de duración típica y con 12 canales de adquisición. También se pensó en su transmisión por redes asíncronas y enviando la información completa o en bloques, por ello se hace un incapié especial en la utilización de códigos de detección de errores (código de redundancia cíclica, CRCs). En el apartado de ventajas y desventajas se analizará en más detalle los inconvenientes de estas consideraciones, por el momento se presentará el formato de datos sin mayor análisis.

El archivo SCP está conformado por secciones. Cada sección es un conjunto de datos especificados dentro de la norma. Están especificadas 12 secciones, aunque se ha dejado la posibilidad de definir hasta la 1024 (se deja a los fabricantes que puedan definir sus propias secciones). Se recomienda el uso de una misma estructura para cada sección que consiste en un encabezado de sección (Section Header) de 16 bytes y un bloque de datos (Data Part) según las especificaciones de cada sección. En este capítulo no se pretende explicar todos los detalles del formato de datos sino una explicación general para la mejor comprensión de la norma.

En la Figura 3 se pueden observar las secciones que ya se han definido, y a continuación se da una pequeña explicación del contenido de cada una.

Obligat	2 Bytes. CRC de todo el archivo.
Obligat	4 Bytes. Tamaño de todo el archivo en bytes.
Obligat	(Sección 0) Punteros a las secciones del archivo
Obligat	(Sección 1)

	<p style="text-align: center;">Información general Datos sobre el paciente y la adquisición</p>
Opcion	<p style="text-align: center;">(Sección 2) Tabla/s Huffman SI se utilizan en la compresión</p>
Opcion	<p style="text-align: center;">(Sección 3) Definición de los canales de adquisición empleados</p>
Opcion	<p style="text-align: center;">(Sección 4) Localización de los QRSs (si se codifica la señal promedio)</p>
Opcion	<p style="text-align: center;">(Sección 5) Muestras codificadas de la señal promedio (si se utiliza)</p>
Opcion	<p style="text-align: center;">(Sección 6) Muestras codificadas de la señal residuo (sustracción señal promedio) O Muestras codificadas de la señal sin sustracción</p>
Opcion	<p style="text-align: center;">(Sección 7) Medidas sobre el registro ECG</p>
Opcion	<p style="text-align: center;">(Sección 8) Diagnóstico del dispositivo que realiza al análisis</p>
Opcion	<p style="text-align: center;">(Sección 9) Diagnósticos especiales del fabricante y nuevas interpretaciones</p>
Opcion	<p style="text-align: center;">(Sección 10) Medidas sobre cada canal de adquisición</p>
Opcion	<p style="text-align: center;">(Sección 11) Anotaciones</p>

Figura A. 3 Formato de Secciones

Cada sección tiene la misma estructura general: el encabezado consta de 16 bytes donde se almacena la longitud, el CRC, el número de sección, etc, y el bloque de datos contiene los datos especificados para esa sección y su longitud es variable (Figura A.4).

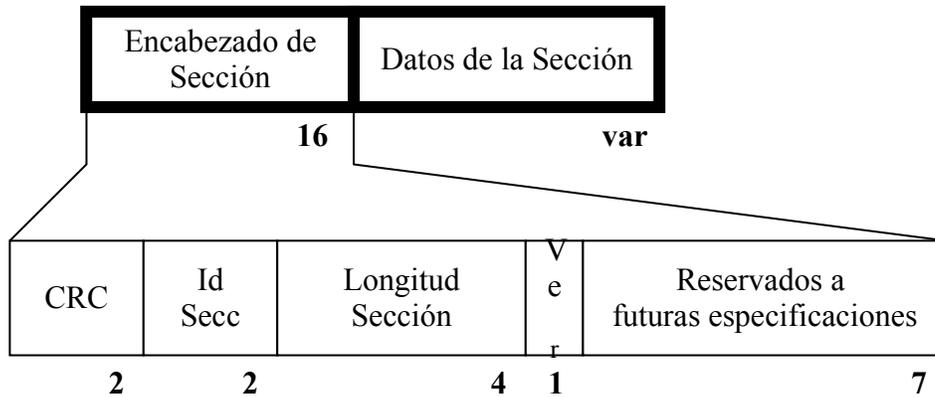


Figura A. 4 Formato General de la Sección

Los datos que contiene cada sección son:

- ❑ Sección 0. Obligatoria. Contiene información de las diferentes secciones y los punteros (índices) a cada una de ellas dentro del archivo.
- ❑ Sección 1. Obligatoria. Tiene información sobre el paciente y la adquisición de la señal ECG. Los únicos campos que deben estar presentes son: id del paciente, fecha y hora de la adquisición e id del dispositivo.
- ❑ Sección 2. Estructura de las tablas Huffman empleadas para la compresión y descompresión de los datos.
- ❑ Sección 3. Especificaciones de los canales de adquisición
- ❑ Sección 4. Si se codifican las señales promedio para cada canal, en esta sección se almacenan los punteros que localizan los QRSs y determinan el comienzo y el fin del área protegida del diezmado y filtrado.
- ❑ Sección 5. Si se ha utilizado el esquema “high compression” aquí se almacenan los datos de las señales promedio para cada canal.
- ❑ Sección 6. Aquí se almacena la señal comprimida. Si se utilizan las señales promedio (high compression) tendríamos la señal residuo, y en otro caso (no habría Sección 5) tendríamos la señal sin restar (rhythm data).
- ❑ Sección 7. Medidas de la señal. Por ejemplo, intervalos RR, comienzo o fin onda P, onda T, QRS, etc.
- ❑ Sección 8. Historia médica. Aquí se tiene el formato para almacenar el último diagnóstico del paciente con este registro.
- ❑ Sección 9. Diagnósticos específicos del fabricante. La estructura del bloque de datos (data part) se deja al criterio del fabricante. También se puede utilizar esta sección para las interpretaciones a posteriori.
- ❑ Sección 10. Medidas para cada canal de adquisición. Se especifican hasta 50 parámetros diferentes por ejemplo las duraciones y/o amplitudes de cada figura del ECG.

- Sección 11. Anotaciones (Universal Interpretative Statement Codes). En esta sección se almacenan las anotaciones del ECG según la sintaxis especificada. También se especifica un diccionario de códigos.

Además de estas secciones se tienen 6 bytes al principio del archivo, los dos primeros son el cálculo del CRC sobre todo el archivo y los otros 4 la longitud en bytes de todo el archivo. Si se calcula la “posible” longitud del fichero (4 bytes sin signo) se ve que se permiten ficheros muy grandes, aunque en otras especificaciones (por ejemplo para cada canal) hay restricciones para señales de mucha duración, como es el caso de registros Holter, por ejemplo. El objetivo general del estándar son registros de ECG de 12 canales y de duración 10 seg, que son parámetros típicos de ECGs de reposo.

Un archivo SCP tiene que tener obligatoriamente dos secciones, la 0 y la 1. Además de los 6 bytes del encabezado del archivo. En la sección 0 es obligatorio que esté presente la información de cada sección, y si la sección no está presente se pondrá longitud 0 e índice 0 también. Por lo tanto, la sección 0 *siempre* tendrá la misma longitud que son 136 bytes. En realidad esto no es cierto ya que si algún fabricante especifica otro tipo de sección se debe añadir aquí, pero en nuestras consideraciones se toman sólo 12 secciones.

La sección 1 es obligatoria, pero no toda la información que allí se especifica. Los campos que son obligatorios son la identificación del paciente y del dispositivo y la fecha y hora de la adquisición. La longitud aproximada, depende de los bytes empleados para codificar cada campo, de la sección 1 es de 50 bytes. Por lo tanto, con los 6 bytes del principio, 136 de una sección 0 típica y 50 de la sección 1, tendríamos cerca de 200 bytes que es la longitud mínima de un archivo SCP.

La longitud máxima no es fácil de determinar ya que no interesa exactamente la longitud del archivo (de hecho las especificaciones permiten una longitud enorme) sino que la pregunta es : ¿Cuántos canales puedo tener? ¿Cuál es la máxima duración en la adquisición de un canal?

Las especificaciones permiten hasta 65 canales diferentes. En realidad ya se sabe que las derivaciones típicas son 12, y que muchas de ellas son combinaciones unas de las otras. También se había pensado en lo que se ha denominado mapeo de electrodos (vectorcardiograma) pero no está todavía especificado.

La duración para un solo canal está especificada en dos bytes, en la sección 6. Por lo tanto se pueden tener $2^{16} = 65535$ bytes de data comprimida. Proponiendo una hipótesis de adquisición, por ejemplo, que se ha adquirido a 500 muestras por segundo, y que la tasa de compresión fue de 5, tendríamos que 65536 bytes comprimidos (aprox 60 Kb) darían 300 Kb de información, es decir (dos bytes por muestra) 150 mil muestras aprox. Y esto son como 300 segundos. Por lo tanto de un cálculo aproximado se observa que la capacidad máxima está bastante limitada ya que 5 minutos de adquisición es muy poco en muchos casos, por ejemplo en registros Holter o de monitoreo por ejemplo. En realidad ya se ha dicho que la norma está orientada a registros ECG ambulatorios, que no sobrepasan este límite. En la literatura no se encuentra ningún estudio acerca de los límites de aplicación de la norma SCP, sólo explicaciones acerca de ventajas e inconvenientes de aplicación.

A.1.5. DATOS DEMOGRÁFICOS, DE DIAGNÓSTICO Y DE ADQUISICIÓN (NO DATA)

Por datos demográficos se entienden todos aquellos relativos al paciente. Los campos típicos serían nombre, edad, sexo, etc... Además se ha hecho un incapié especial en los datos de la adquisición. Se tienen en cuenta datos del dispositivo empleado, la institución y el personal técnico que realiza la adquisición, las condiciones, el médico que ha hecho el diagnóstico, etc... Todos estos van en la sección 1. Los datos sobre los canales empleados están en la sección 3. La identificación del registro se hace con el ID del paciente, que se toma (recomendaciones de la norma) como la clave de identificación de la base de datos. En el capítulo III donde se explica la etapa de diseño y análisis de la aplicación implementada se toma este valor como único representante del archivo.

El otro tipo de datos que no son “binarios” serían los relativos al diagnóstico. Algunos de estos datos, por ejemplo la historia médica, diagnóstico, drogas suministradas, etc, están también en la sección 1. La sección 8 contiene el último diagnóstico de este paciente (relativo a este registro) y otros diagnósticos anteriores. Se almacenan cadenas de texto. Si el fabricante tiene algún diagnóstico específico se especificará en la sección 9.

Por último se tendrían las anotaciones, que estarían en la sección 11 y cuyo mecanismo de anotación y almacenamiento se va a explicar en el siguiente apartado.

A.1.6. ANOTACIONES

El proyecto SCP hizo un esfuerzo notable para tratar de ofrecer un léxico adecuado para el almacenamiento de interpretaciones clínicas de la señal ECG. Por ello la propuesta de la norma SCP es un conjunto de mnemónicos y modificadores que pueden reflejar desde interpretaciones simples hasta las más complejas.

El objetivo final del proyecto OEDIPE está en la colaboración de diferentes centros de investigación sobre bases de datos remotas, y lo que se intercambia suele ser la interpretación de los ECGs. Una vez adquirido el ECG, éste no cambia, sólo la interpretación que uno, o varios médicos puede hacer de la señal. Por ello se dedicó tanto esfuerzo a proveer un mecanismo eficiente y universal de anotación. A continuación se presenta primero la sintaxis y la composición del código y después algunas apreciaciones de los mnemónicos y modificadores empleados.

A.1.6.1. Sintaxis y composición del código

En primer lugar decir que el código que representa a una anotación esta compuesto por varios mnemónicos que representan interpretaciones básicas, modificadores y otros que sirven para formar interpretaciones complejas.

La sintaxis básica está formada por tres campos principalmente. El primer campo define la interpretación básica (acrónimo), el segundo define el grado de precisión de la interpretación (probable, definitivo, etc...) y los otros campos incluyen otros modificadores

A.1.6.2. Mnemónicos y modificadores empleados

Los mnemónicos que están en el diccionario de anotaciones son fáciles de recordar y han sido ampliamente utilizados en la literatura. Por tanto, son fáciles de manejar por los médicos (generalmente cardiólogos) que realizan las interpretaciones. Además, los mnemónicos en realidad son acrónimos, es decir siglas que representan la interpretación completa. Estos códigos se dividen en 10 clases. Se van a mostrar ahora estas clases y algunos ejemplos de mnemónicos y su explicación. También se ha puesto la interpretación completa en inglés, que es el origen del acrónimo:

- Hipertrofia Ventricular
 - LVH: Hip. Ventric. Izquierda
(*Left Ventricular Hypertrofia*)
- Infarto de Miocardio
 - ALMI: Inf. Mio. Anterolateral
(*AnteroLateral Myocardial Infarction*)
- Problemas de conducción
 - IRBBB: Bloqueo incompleto en la rama derecha
(*Incomplete Right Bundle Branch Block*)
- Anotaciones generales
 - NLECG: ECG Normal
(*Normal ECG*)
- Descripción de otras morfologías del QRS
 - PE: enfisema Pulmonar
(*Pulmonary Enphysema*)
- Ritmo
 - ACAR: Ritmo auricular acelerado
(*Accelerated Atrial Rhythm*)
- Ejes
 - LAD: Desviación Eje Izquierdo
(*Left Axis Deviation*)
- Descripción de segmento ST y/o T
 - NST: Elevación no significativa del intervalo ST
(*Non specific ST changes*)
- Descripción Auricular
 - RAO: Sobrecarga Auricular. Izquierda
(*Right Atrial Overload*)

- Problemas técnicos
- FAULT: fallo en la adquisición
(*FAULTy lead*)

La lista completa se puede consultar en los manuales para médicos de muchos dispositivos. Los modificadores son acrónimos que se añaden información a la interpretación básica. Esto puede ser por varias razones, por ejemplo, el grado de gravedad, el tiempo, la evolución, etc... Ejemplos de modificadores serían RE (reciente, *REcent*), SP (subepicárdico, *SubePicardial*), LA (lateral, *lateral*) y IM (intermitente, *InterMittent*).

También se pueden tener términos que enlazan varias interpretaciones para obtener una más compleja. Ejemplos típicos serían los operadores AND y OR.

A.1.6.3. Ejemplos de anotación

Algunos ejemplos de anotaciones que se pueden encontrar en un registro SCP se exponen a continuación:

- P_AMP_INN_LEAD_V1_EQU_120_MVOLT: La amplitud de la onda P en V1 es de 120 mv
- Q_DUR_INN_LEAD_D3_EQU_40_MSEC: La onda Q en D3 dura 40 ms
- (MAX_R_AMP_ADD_MAX_S_AMP)_INN_V_LEADS_IGT_4.5_MVOLT: Las amplitudes máximas de R y S en los canales V es mayor de 4.5 mv.