

Capítulo 4

Estructura del Sistema

4.1. Introducción

Como hemos venido comentando en los capítulos anteriores, el sistema que tratamos de desarrollar en este Proyecto tiene como objetivo realizar el envío de los datos clínicos de los pacientes que acuden a las consultas de los centros ambulatorios del HUVR adjuntando a dichos datos una imagen digital de la lesión tomada con la ayuda de una webcam. La idea inicial era implantar nuestro sistema en las instalaciones que constituyen la intranet del Servicio Andaluz de Salud (SAS), pero a la hora de implementar las comunicaciones nos encontramos con el problema de que la empresa que gestiona dicha red no facilita el acceso a la misma por motivos de seguridad, por lo que se decidió integrar nuestro trabajo en un Sistema de Gestión de Imágenes Médicas que fue realizado como Proyecto Fin de Carrera en el año 2002 [10] para desarrollar dichas comunicaciones.

Cuando tenemos un sistema de comunicaciones con arquitectura en red en el que varios nodos solicitan a un nodo centralizado que realice diversas acciones con las

unidades de información que le envían (como por ejemplo almacenarlas en una base de datos centralizada), lo lógico es emplear una arquitectura de tipo cliente-servidor.

4.2. La arquitectura cliente-servidor

Los nodos de red en la arquitectura cliente-servidor no son todos iguales, sino que se dividen en dos tipos: servidores y clientes.

La arquitectura cliente-servidor es una extensión lógica de la programación modular, que asume que partir un programa extenso en varias partes o módulos permite un más rápido desarrollo y mejor mantenimiento del sistema en su conjunto. La arquitectura cliente-servidor va un paso más allá y reconoce que todos esos módulos no tienen por qué ser ejecutados en el mismo espacio de memoria. Dentro de esta arquitectura, el módulo que genera las llamadas al sistema es el módulo cliente, mientras que el que responde es el módulo servidor. Una ventaja que ofrece lo anterior es que cada módulo puede estar corriendo en un hardware apropiado a las funciones que dicho módulo debe desempeñar. Así, por ejemplo, los servidores de sistemas de gestión de base de datos funcionarán en plataformas específicas para llevar a cabo peticiones, o los servidores de archivos estarán equipados con elementos específicos del manejo de archivos y mantenimiento de los datos (copias de seguridad, etc.).[11]

4.2.1. El cliente

Cuando hablamos de cliente a secas, podemos referirnos al nodo físico que ejecuta las aplicaciones cliente, o a la aplicación cliente en sí. Igual ocurre con el servidor, ya que podemos referirnos al hardware o al software. En general, no es fácil separar máquina y aplicación, ya que esta última se ejecutará preferentemente sobre una máquina que cumpla con los requisitos hardware para su correcto funcionamiento y al contrario, también ocurre que por motivos económicos y de fiabilidad, el hardware de las máquinas se ajustará a los procesos que en ella se tengan que ejecutar. En cualquier caso, es importante distinguir lo que es de la aplicación de los que es el nodo físico, tal y como se ha dicho antes.

La aplicación cliente es una aplicación que envía un mensaje al proceso servidor, requiriendo que el servidor lleve a cabo una determinada tarea. Las aplicaciones cliente suelen manejar la parte de interfaz de usuario (que suele ser gráfica) de la aplicación, validan los datos introducidos por el usuario, realizan las peticiones al servidor y algunas veces realizan operaciones con los datos de respuesta. El cliente también gestiona los recursos locales con los que el usuario interactúa, como el monitor, el teclado, los periféricos...

4.2.2. El servidor

La aplicación servidor ejecuta las peticiones del cliente, llevando a cabo las tareas pedidas por éste. Las aplicaciones servidor normalmente reciben las peticiones de las aplicaciones cliente, ejecutan la captura y la modificación de información de la base de datos, gestionan la integridad de los datos y responden a las peticiones de los clientes.

Mientras que el tipo de operaciones que realiza el cliente son normalmente específicas del proceso concreto, las funciones realizadas por el servidor suelen ser comunes a muchos sistemas distintos.

4.2.3. Características principales de la arquitectura cliente-servidor

Las características más destacables de una arquitectura cliente-servidor son las siguientes:

- ♣ Se combina un cliente que se comunica con el usuario con un servidor que interactúa con uno o varios recursos compartidos. Como resultado, el proceso cliente proporcionará la interfaz de usuario, mientras que el proceso servidor proporcionará los medios para gestionar los recursos compartidos.
- ♣ El proceso cliente y el proceso servidor tienen diferentes requerimientos en términos de hardware.
- ♣ El entorno en el que se desenvuelve una arquitectura cliente-servidor es fundamentalmente un entorno heterogéneo con múltiples proveedores de hardware y múltiples plataformas. Como consecuencia de esto, para que la

comunicación transcurra correctamente, es necesario que los clientes y el servidor se comuniquen por medio de una interfaz bien conocida.

- ♣ Se trata de una arquitectura escalable, tanto en vertical como en horizontal. Un escalado horizontal implica que se pueden añadir más ordenadores que actúen como cliente, mientras que un escalado vertical implica que podemos migrar del servidor hacia sistemas más potentes o rápidos o a un sistema de varios servidores.

4.3. Estructura general del sistema

En el capítulo 2 se explicó que la ficha médica generada por cada paciente en los distintos centros ambulatorios, se enviaría al servidor del que dependiese dicho centro y los datos junto con las imágenes serían almacenados en una base de datos centralizada. Por lo tanto y de acuerdo a lo que se ha visto en el apartado anterior, se deduce que la aplicación creada para los centros ambulatorios será la aplicación cliente, y el destino al que se envían los datos y las imágenes no es otra cosa que la aplicación servidor, que se encargará de recibir dichos datos, almacenarlos en una base de datos centralizada y procurar que se puedan consultar desde la central del HUVR mediante una segunda aplicación cliente que será implementada para sus equipos.

También se ha comentado con anterioridad que el tratamiento y almacenamiento de imágenes médicas y los datos que a éstas acompañan debía cumplir una serie de requisitos muy exigentes que se encuentran recogidos en el estándar DICOM [12] [13] (*Digital Imaging and Communications in Medicine*), que fue creado por el Colegio Americano de Radiología [14] (ACR, *American College of Radiology*), y la Asociación Nacional de Fabricantes Eléctricos [15] (NEMA, *National Electrical Manufacturers Association*). Su objetivo fundamental es facilitar la interoperabilidad de los equipos de imágenes médicas especificando para ello un conjunto de protocolos que deben seguir los equipos, junto con la sintaxis y semántica de los comandos e información asociada que puede ser intercambiada mediante estos protocolos, y lo estudiaremos en profundidad en capítulos posteriores. Por este motivo, el formato de los datos médicos que se introduzcan desde la aplicación cliente seguirá las pautas marcadas por este estándar y como servidor aprovecharemos el servidor DICOM que fue implementado en

el Proyecto Fin de Carrera que mencionamos anteriormente [10] con pequeñas modificaciones en algunos de sus módulos.

En la figura 4.1 podemos ver cómo quedará la estructura general de nuestro sistema. En ella encontramos los siguientes elementos:

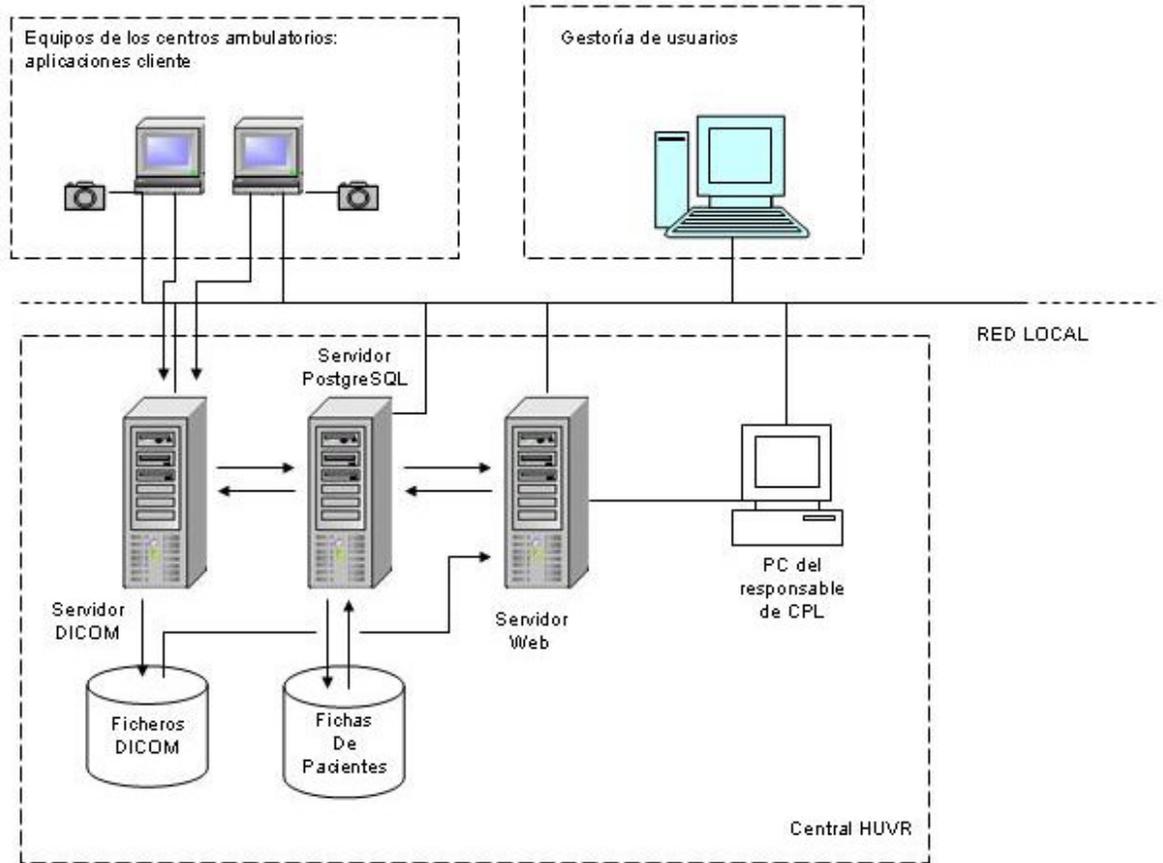


Figura 4.1. Estructura general del Sistema

- ♣ *Equipos de los centros ambulatorios:* son los orígenes de las imágenes médicas, es decir, el PC junto con la webcam en cada uno de los centros ambulatorios. El operador deberá introducir los datos del paciente al generar el fichero, de forma que quede correctamente identificado. El fichero se enviará a través de la red hacia el Servidor DICOM para su almacenamiento.
- ♣ *Servidor DICOM:* recibe los ficheros que envían los equipos de los centros ambulatorios, almacenándolos en una base de datos situada en un directorio determinado. Al guardar el fichero, se inserta en la base de datos de pacientes la

información relativa al propio fichero, y se actualiza la ficha del paciente al que pertenece el fichero. Implementa los servicios de almacenamiento y de verificación de la conexión, y lo estudiaremos después con algo más de detalle.

- ♣ *Servidor PostgreSQL*: en este caso se trata del gestor de base de datos PostgreSQL [16] [17] [18]. Gestiona la información del sistema, atendiendo las distintas consultas que se realizan tanto desde el Servidor DICOM como desde el servidor Web.
- ♣ *Servidor WEB*: permite a los usuarios interactuar con el sistema, a través de páginas Web. Constituye la base de la interfaz de usuario en la central del HUVR, y por tanto, accede tanto a la información relativa a los pacientes como a los propios ficheros DICOM.
- ♣ *Gestoría de usuarios*: cuando el médico responsable en la central del HUVR decide que un paciente es apto para CmA en acto único, la gestoría de usuarios recibe una notificación con la descripción de la intervención que debe ser realizada a dicho paciente. Su función es simplemente la de asignar el cirujano, la fecha y la hora en que dicha operación tendrá lugar y avisar al paciente en la forma en que éste haya solicitado ser avisado (por teléfono, por personal del hospital o por correo).
- ♣ *Red Local*: permite interconectar todos los equipos entre sí. El protocolo utilizado será TCP/IP.

En realidad, excepto los equipos de los centros ambulatorios y la gestoría de usuarios, todos los demás pueden residir en el mismo equipo físico, reduciendo así los costes. De forma general, el funcionamiento del sistema se recoge en cuatro operaciones:

- ♣ *Envío de un fichero DICOM*:
 1. El facultativo responsable recoge los datos personales y clínicos del paciente y los inserta en el formulario de consulta implementado mediante la aplicación cliente, así como la imagen de la lesión que es objeto de la consulta.
 2. Una vez que el formulario está completo, se indica en la aplicación cliente, que creará un archivo acorde con los requerimientos del estándar

DICOM y lo enviará a través de la red siguiendo el protocolo correspondiente a dicho estándar.

♣ *Almacenamiento de un fichero DICOM:* los pasos a seguir serían:

1. El equipo del centro ambulatorio envía el fichero DICOM al Servidor DICOM, mediante los protocolos recogidos en el Estándar.
2. El Servidor DICOM consulta, a través del Servidor PostgreSQL, la ficha del paciente referenciado en el fichero, comprobando que éste existe.
3. El Servidor DICOM almacena en el directorio correspondiente el fichero recibido, e inserta la información correspondiente en la ficha del paciente, a través del Servidor PostgreSQL.

♣ *Consulta de datos e imágenes:* los pasos a realizar son:

1. El médico responsable en la Central del HUVR solicita una página Web de la interfaz de usuario.
2. El Servidor Web consulta al Servidor PostgreSQL para obtener información acerca del paciente.
3. Una vez obtenida toda la información del paciente, se accede al fichero DICOM que contiene la imagen solicitada. Para ello se utilizan programas CGI, que traducen el formato de la imagen contenida en el fichero DICOM al formato jpeg, compatible con el navegador Web del usuario.

♣ *Envío de la respuesta del facultativo a la Gestoría de Usuarios:* se hace mediante los siguientes procedimientos:

1. El médico responsable en la Central del HUVR solicita la ficha con los datos personales y clínicos del paciente y toma la decisión correspondiente acerca de la clasificación del paciente como apto o no para CmA en su modalidad de acto único.

2. Emplea la aplicación de contestación que se ha creado y rellena los datos pertinentes. Con esto, se crea un fichero de texto con los datos personales del paciente y el diagnóstico del facultativo referente a la clasificación del paciente.
3. El fichero de texto que se ha creado se almacena en el PC de la central del HUVR y posteriormente se envía a la Gestoría de Usuarios siguiendo las pautas del protocolo de transferencia de ficheros por excelencia: *FTP*¹.

¹ File Transfer Protocol