Soluciones para mostrar imágenes DICOM en entorno web

Eduardo Martínez Rojo

Resumen— El estándar para imágenes médicas DICOM (Digital Imaging and Communications in Medicine) surgió ante la necesidad de estandarizar las comunicaciones de diagnósticos digitales. Originalmente pensado para la comunicación y transmisión de los objetos médicos mediante el protocolo TPC/IP, ha evolucionado a lo largo del tiempo para incluir definiciones de servicios web en su estándar. En este proyecto se dota a un servidor de imágenes médicas de un servicio WADO (Web Access to DICOM Objects) y una interfaz web que permite explorar el contenido del servidor y visualizarlo en el propio explorador web. Para llevar a cabo esta tarea se estudia el estándar así como el mercado con el fin de encontrar soluciones ya existentes al problema, con especial énfasis en las soluciones de código abierto. El resultado final será un servicio web completamente funcional capaz de servir objetos DICOM vía web y que además permite la visualización de los mismos.

Palabras clave— DICOM, WADO, PACS, HTTPS, HTML, JavaScript.

Abstract— The medical image standard DICOM (Digital Imaging and Communications in Medicine) arose from the need to standardize the communications of digital diagnostics. Originally intended for communication and transmission of the medical objects through the TCP/IP protocol, it has evolved over time to include web services definitions in the standard. In this project we provide a medical imaging server with WADO (Web Access to DICOM objects) capability and a web interface able to explore the server's content and visualize it in the web browser. To carry on this task a study of the standard and the market is conducted with the goal to find existing solutions to the problem, with special emphasis in open source solutions. The final result will be a fully functional web service able to serve DICOM objects via web and also visualize them.

Index Terms— DICOM, WADO, PACS, HTTPS, HTML, JavaScript.

1 Introducción

SOLUCIONES para mostrar imágenes DICOM en entorno web es un proyecto que tiene como fin transmitir imágenes médicas DICOM (Digital Imaging and Communications in Medicine) vía HTTP (Hypertext Transfer Protocol) para posteriormente mostrarlas en un entorno web.

El proyecto se lleva a cabo como prácticas en la empresa Systelab, dedicada al desarrollo de software médico-sanitario, y tiene como cliente el Hospital General Universitario Gregorio Marañón. Durante la realización de este trabajo de final de grado se pasa a formar parte de un equipo de desarrollo que trabaja en el proyecto MediVector de Systelab.

En el proyecto MediVector ofrecen varias soluciones de software enfocadas a los departamentos de cardiología de los hospitales, entre ellas se incluyen un software de visualización DICOM para escritorio, un software de gestión de visitas e informes médicos y un servidor PACS (Picture Archiving and Communication System) que gestiona objetos DICOM.

La necesidad de llevar a cabo este proyecto surge cuando el cliente requiere recuperar objetos DICOM del PACS desde otros sistemas del hospital sin necesidad de instalar el software mencionado anteriormente.

La recuperación de objetos DICOM mediante HTTP es el requisito del cliente, mientras que la visualización de los mismos en una interfaz web es una propuesta de Systelab que tiene como objetivo explorar las posibilidades en este ámbito.

A lo largo de este documento se introducirá el estándar DICOM, se situará el proyecto en un contexto de desarrollo que incluye unos objetivos y la metodología que se seguirá para completarlos. Finalmente se detallará el proceso de implementación y los resultados obtenidos.

2 El estándar DICOM

Antes de analizar el estado del arte, introduciremos brevemente el estándar DICOM, a fin de poder señalar algunos de los aspectos de especial relevancia para el proyecto.

DICOM es un estándar internacional que define un formato de almacenamiento de imágenes médicas así como los protocolos de comunicación de las mismas. La necesidad de desarrollar este estándar surge en la década de 1970 con la introducción de las primeras modalidades[1] de diagnóstico digital, siendo publicada la primera versión del estándar en 1985. [1]

La comunicación entre dispositivos que usan el proto-

- *E-mail de contacto: eduardo.martinez@e-campus.uab.cat*
- *Mención realizada: Tecnologías de la Información.*
- *Trabajo tutorizado por: Joan Serra Sagristà(dEIC)*
- *Curso 2015/16*

[1] Modalidad: en DICOM la modalidad hace referencia al contenido del archivo, que está estrechamente relacionado con el tipo de dispositivo médico que lo genera. (ej. Radiología, resonancia magnética, etc.

colo de red DICOM se hace mediante TCP/IP. Para establecer una conexión, los dispositivos o aplicaciones negocian quién es el cliente y quién es el servidor y establecen una asociación. Una vez asociados pueden enviarse órdenes para copiar, guardar, borrar y mover objetos DICOM.

Los objetos DICOM no se limitan a los datos de la imagen medica si no que contienen una estructura de datos más amplia. Cada elemento dentro un objeto DICOM se clasifica según un diccionario de etiquetas que identifican de forma única un campo de datos. Entre estas etiquetas, muy numerosas, se encuentran campos que apuntan a los datos del paciente (nombre, fecha de nacimiento), los datos de la imagen y a qué estudio y serie pertenece el objeto.

Para clarificar lo que es una serie en DICOM, debemos referenciar la jerarquía, presente en la figura 1. En este estándar un paciente tiene múltiples estudios, un estudio tiene múltiples series y una serie tiene múltiples instancias (imágenes, objetos DICOM). [2]

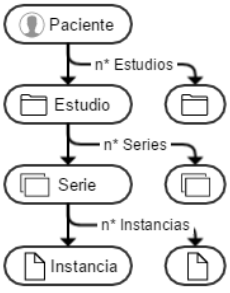


Fig. 1. Jerarquía de objetos DICOM

Uno de los principales objetivos del estándar es que los objetos se puedan identificar de forma única. En un objeto DICOM, los datos del paciente, la imagen y otros datos que relacionan un objeto con los demás (como las series y estudios) quedan vinculados de forma inequívoca.

2.1 Acceso web a objetos DICOM

Desde el año 2004 el estándar DICOM parte 18 incluye la definición de WADO (Web Access to DICOM objects). Esta parte ha evolucionado considerablemente desde entonces y actualmente se titula "Web Services". En esta parte del estándar se describe cómo deben ser las peticiones y respuestas en un servicio web que sirve objetos DICOM y a lo largo de su evolución ha incorporado definiciones para servicios web y servicios web RESTful[2]. [3]

Esta parte del estándar es de especial interés para el proyecto ya que el objetivo de permitir el acceso a los objetos DICOM vía web puede cumplirse idealmente implementando un servicio que se atenga a las especificaciones del estándar.

La ventaja que ofrece un servicio WADO es ofrecer una vía alternativa de comunicación con un servidor PACS mediante HTTP para recuperar objetos, sin necesidad de implementar el protocolo de comunicaciones del estándar DICOM. Esto abre las puertas a la información del servidor para clientes web que entienden el formato de un objeto DICOM, pero no necesariamente el protocolo de comunicaciones, como sería el caso de un visualizador DICOM basado en web.

3 Estado del arte

El propósito de este proyecto no es pionero en el ámbito de la informática para sistemas médicos. Actualmente existen múltiples herramientas de licencia libre o comercial y de código libre o abierto que ofrecen soluciones a diferentes niveles dentro del ámbito de la gestión y visualización de imágenes médicas DICOM vía web.

Es importante señalar que las herramientas estudiadas son principalmente de código libre debido a que los intereses de Systelab para el proyecto MediVector no hacen deseable introducir un nuevo actor en la cadena de requisitos y cambios del proyecto.

Entre las herramientas disponibles en el mercado podemos encontrar principalmente tres tipos de soluciones. A continuación tratamos algunas de las más relevantes para cada una de las clasificaciones.

El primer tipo de solución son servidores DICOM completos con capacidad almacenamiento, gestión, exploración y visualización de objetos DICOM. Entre las opciones encontramos los proyectos "dcm4che" y "Orthanc", ambos de código abierto y bajo una licencia de software libre. Ambos proyectos ofrecen un servidor de imágenes médicas con capacidad de gestión y exploración de los contenidos así como la visualización de los mismos en el explorador web mediante su propia implementación de un visualizador de imágenes médicas. Adicionalmente disponen de implementaciones del protocolo WADO para que otras aplicaciones de terceros como visualizadores web u otros sistemas PACS puedan recuperar los objetos del servidor vía HTTP sin la necesidad de establecer comunicaciones mediante el protocolo de comunicaciones de DICOM. [4][5]

El segundo tipo de solución son herramientas de visualización de objetos DICOM que permiten configuración e integración con un PACS existente. Entre estas herramientas podemos encontrar Oviyam (que forma parte del proyecto dm4che), Cornerstone y DICOM Web Viewer. Estos tres visualizadores de imágenes tienen en común estar basados en HTML5 y JavaScript por lo que permiten la manipulación, visualización y acceso a información de los objetos DICOM íntegramente en un explorador web. [6][7]

El tercer tipo de solución son frameworks[3] enfocados a facilitar la creación y personalización de visualizadores de imágenes médicas. En este ámbito pueden encontrase soluciones comerciales como los kits de desarrollo de software de LeadTools, que incluyen extensas librerías para visualización web DICOM basándose en HTML5 y JavaScript. [8]

Finalmente, podría optarse por construir un software de visualización propio desde cero. Esta última alternativa se descarta en una de las fases iniciales del proyecto, cuando se realizó un estudio de mercado en busca de soluciones

[2] Servicio web RESTful: Estilo arquitectural de software para diseño de servicios web.

[3] Framework: Marco de trabajo, conjunto de herramientas, prácticas y piezas de software que facilitan el desarrollo de aplicaciones.

como las presentadas aquí. Esta decisión se basa en la calidad de las herramientas existentes y la libertad que ofrecen las licencias bajo las que se encuentran (algunas de ellas bajo licencia MIT[4]), por lo que se considera apropiado hacer uso de una herramienta de código abierto.

4 CONTEXTO DEL PROYECTO

Antes de describir como se ha desarrollado el proyecto estudiamos brevemente el contexto de arquitectura hardware y software sobre la que se implanta, definimos los objetivos propuestos y la metodología usada para alcanzarlos.

4.1 Arquitectura

En el entorno para el que se desarrolla este proyecto, el sistema de información del Hospital General Universitario Gregorio Marañón, existe un PACS dedicado exclusivamente al área de cardiología, desarrollado por el equipo de MediVector. El servidor en el que se ubica este PACS será el objetivo de este proyecto.

En la figura 2 se puede observar una representación de la infraestructura en la que se implantaría el proyecto una vez finalizado. El elemento más importante que debemos destacar son los objetos COM (Component Object Model). [9]

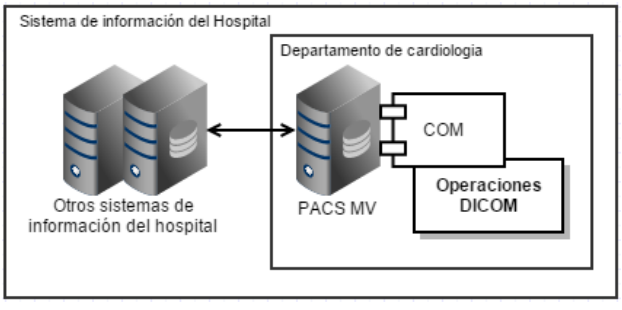


Fig. 2. Arquitectura general en la que se implantará el proyecto.

El PACS de MediVector dispone de librerías de objetos COM que exponen sus servicios y actualmente se usan para las comunicaciones DICOM. Los servicios que se desarrollarán en este proyecto interactuarán con el servidor mediante estas librerías para ejecutar operaciones análogas a las que ofrece el protocolo DICOM.

4.2 Objetivos

El objetivo del proyecto, transmitir y recuperar imágenes médicas DICOM vía HTTP para posteriormente mostrarlas en un entorno web, se puede dividir principalmente en tres partes.

1. Implementar la capacidad de recuperar objetos DICOM vía HTTP del servidor PACS de MediVector. Este servicio debe cumplir los siguientes requisitos:
    a. Capacidad para recibir y contestar a peticiones HTTP.
    b. Traducir las peticiones HTTP a búsquedas sobre el servidor PACS.
    c. Servir los objetos DICOM de la base de datos del PACS vía HTTP.
    d. Cumplir las anteriores características en base a lo especificado en el estándar DICOM WADO.
2. Implementar una interfaz web que permita visualizar el contenido del PACS. Esta interfaz web debe interactuar con el servidor para poder llevar a cabo búsquedas sobre la base de datos. Esta interfaz web debe cumplir los siguientes requisitos:
    a. Poder identificar a un usuario con las mismas credenciales que en el sistema de información de MediVector.
    b. Capacidad para llevar a cabo búsquedas sobre la base de datos del PACS.
    c. Recuperar y devolver la información en un formato que facilite la interpretación en el cliente.
    d. Opcional: Al ser una interfaz web, se puede considerar compatibilidad con dispositivos móviles, aunque no es un requisito del proyecto.
3. El tercer objetivo es implementar una forma de visualizar los objetos DICOM en un entorno web. Los requisitos para este objetivo son:
    a. Capacidad para recuperar un objeto DICOM vía http.
    b. Visualizar el objeto.
    c. Permitir manipulaciones básicas sobre el objeto (aumentar, mover, reproducir).
    d. Integración con la interfaz web.
    e. Opcional: Al ser una interfaz web, se puede considerar compatibilidad con dispositivos móviles, aunque no es un requisito del proyecto.

Finalmente, todos los servicios de este proyecto deben establecer sus comunicaciones exclusivamente mediante una conexión segura debido a que sirven imágenes médicas DICOM, que contienen información personal de carácter médico y por lo tanto están protegidos bajo la LOPD (Ley Orgánica de Protección de Datos). [10]

Esta condición se cumplirá usando el protocolo SSL (Secure Sockets Layer) para establecer conexiones seguras vía HTTPS.

4.3 Metodología de desarrollo

El proyecto se lleva a cabo trabajando dentro de un equipo que sigue la metodología de desarrollo de software Scrum, que se encuentra dentro de las metodologías agiles y asume que los requisitos de los clientes pueden cambiar durante el proceso de desarrollo.

En la metodología Scrum se definen fases de tiempo en la que se llevan a cabo "sprints". En estos sprints se definen

[4] Licencia MIT: Licencia corta y muy permisiva. Permite uso comercial, distribución, etc.

una serie de tareas y una fecha para la que deberían haberse realizado y ese trabajo debería ser potencialmente un producto entregable (o un cambio del producto entregable). Se llevan a cabo reuniones diarias para hablar del trabajo que se hace de cara a completar los objetivos de un sprint.

Si bien durante el desarrollo de este proyecto no se han cambiado los requisitos en ningún momento, esta metodología de trabajo con revisiones diarias ha ayudado notablemente a descubrir fallos en la planificación y la previsión de las tareas. Como resultado la línea de desarrollo ha estado bien definida a lo largo de todo el proyecto.

Para el control del proyecto se hace uso de Team Foundation Server. Esta plataforma permite seguimiento de las tareas del proyecto, relaciones entre ellas y el tiempo invertido en cada una así como compartir y almacenar versiones de los códigos. En el anexo pueden verse imágenes del seguimiento de algunas de las tareas llevadas a cabo en el proyecto mediante esta aplicación.

5 El proyecto

En esta sección se detalla todo el trabajo llevado a cabo a lo largo del proyecto.

En primera instancia se detallan las tecnologías elegidas para el desarrollo.

Seguidamente se explica el funcionamiento de las diferentes partes del proyecto y como encajan e interactúan con la arquitectura existente.

Finalmente mostramos los resultados obtenidos para cada una de las partes correspondientes a los diferentes objetivos del proyecto.

5.1 Tecnologías de desarrollo elegidas

En el desarrollo de este proyecto, Systelab ha ofrecido un alto grado de libertad a la hora de elegir las tecnologías de implementación, siempre que las elecciones fueran apropiadas a los objetivos y pudieran justificarse. Dicho esto, las tecnologías de las que se hace uso en este proyecto son las siguientes:

- Arquitectura de software Modelo Vista Controlador de la plataforma ASP.NET (MVC 5) [11]

La arquitectura de software MVC es muy apropiada para construir y mantener páginas web. A grandes rasgos, en esta arquitectura el usuario interactúa con la Vista (representada por la interfaz) y esta interacción se traduce a la ejecución de funciones en el controlador. El controlador accede a un modelo de datos que modifica en base a la acción que lleva el usuario. Este sistema ayuda a separar de forma clara las responsabilidades de las diferentes partes, simplificando el mantenimiento del software.

Las aplicaciones de este proyecto se desarrollan bajo la plataforma .NET en lenguaje C# (CSharp).

- AngularJS [12]

Un framework JavaScript de código abierto que aplica la arquitectura MVC en la parte de cliente de una aplicación. Permite extender el lenguaje HTML con sintaxis que expresa la lógica de la aplicación JavaScript que hay tras el documento. Al igual que en la aplicación de la arquitectura MVC en la parte de servidor, angular permite vincular partes de la vista (elementos de la página HTML) a un modelo de datos, de forma que cuando uno de los dos se modifica, la otra parte se actualiza automáticamente.

- Bootstrap CSS (Cascading Style Sheets) [13]

Bootstrap está diseñado para facilitar la escalabilidad de los estilos de una página web. Para conseguir eso ofrece un amplio abanico de estilos combinables y extensibles y un sistema de rejilla que puede configurarse para adaptarse automáticamente al tamaño de ventana de diferentes dispositivos. También ofrece un amplio abanico de iconos que ayudan al diseño de una interfaz.

- Html5 [14]

La quinta revisión del lenguaje HTML ofrece nuevos elementos y se deshace de algunos no necesarios. Más importante aún, es un componente esencial de algunas tecnologías usadas en este proyecto, como Bootstrap y algunos de los visualizadores DICOM basados en web que se han analizado. Las herramientas de visualización hacen uso del elemento Canvas, que ofrece múltiples ventajas a la hora de mostrar y manipular imágenes mediante JavaScript. Aunque la principal ventaja que ofrece en el ámbito de este proyecto es la rapidez para dibujar los datos de pixel de una imagen. [15]

5.2 Fases del Proyecto

En esta sección se detallan las fases del proyecto que se han llevado a cabo a fin de cumplir los objetivos planteados anteriormente. Estas fases corresponden con las tareas más importantes de la planificación del proyecto.

En la fase inicial se definen las características del servicio a desarrollar, seguido de una fase de implementación. En la siguiente fase se lleva a cabo un estudio de mercado a fin de decidir la vía de implementación para fases posteriores. Finalmente las últimas fases están enfocadas en el desarrollo de la interfaz web y el visualizador de objetos DICOM que la complementa.

5.2.1 Fase inicial

Al comenzar el proyecto la primera tarea que se llevó a cabo fue una captura de requisitos con el cliente para determinar con exactitud las características que debía tener el servicio que permitiera el acceso a los objetos DICOM vía web. De esta captura de requisitos se deriva la necesidad de crear el servicio WADO.

Adicionalmente se desea la capacidad de acceder a los objetos DICOM en formato de video MP4, con codificación H.264, para visualizar las imágenes directamente en el explorador. Si bien el estándar WADO no contempla peticiones HTTP con un tipo MIME[5] "application/mp4", el que correspondería a este tipo de peticiones, se considera apropiado desarrollar el servicio ampliándolo más allá de las especificaciones del estándar con el fin de cumplir las necesidades del proyecto.

La razón de que el formato elegido sea MP4 es porque

[5] MIME: Multipurpose Internet Mail Extensions. En el protocolo HTTP es una cabecera que identifica el formato o tipo del contenido que se transmite.

es un contenedor de video y audio soportado por los exploradores de internet más usados. [16]

5.2.2 Implementación del servicio WADO

Para la implementación del servicio WADO se estudió la parte 18 del estándar DICOM a fin de determinar los parámetros que debe aceptar un servicio de este tipo y como deben ser las respuestas. El estándar define parámetros en dos variantes; requeridos y opcionales. Los parámetros requeridos son aquellos que identifican de forma inequívoca un objeto DICOM; los identificadores de estudio, serie e instancia. El resto de parámetros hacen referencia a modificaciones sobre la imagen o el formato de los datos. [17]

Adicionalmente el estándar contempla una excepción respecto a estos parámetros. Si un servidor PACS soporta búsqueda por identificador de instancia único, los parámetros de estudio y serie no son necesarios. Este es el caso del servidor para el que se implementa el servicio WADO en este proyecto, por lo que la cantidad de parámetros que deben tratarse y validarse en las búsquedas se reducen.

En la figura 3 puede verse una secuencia UML que corresponde con la ejecución de una petición HTTP sobre el servicio WADO. El diseño del servicio se explica a continuación.

Al recibir una petición HTTP, recoge los parámetros y los vincula a un modelo de datos. Seguidamente valida que los parámetros contengan valores soportados por el servicio, como por ejemplo que el contenido solicitado sea "application/dicom" y no uno desconocido.

Si la petición se valida correctamente el servicio busca la ubicación del objeto en el PACS y si está disponible responde a la petición HTTP devolviendo los datos del objeto.

En caso contrario se responde a la petición con un código de error HTTP según lo especificado en el estándar WADO. Las posibles respuestas son *406 Not Acceptable* para peticiones no válidas y *404 Not Found* para búsquedas sin resultados.

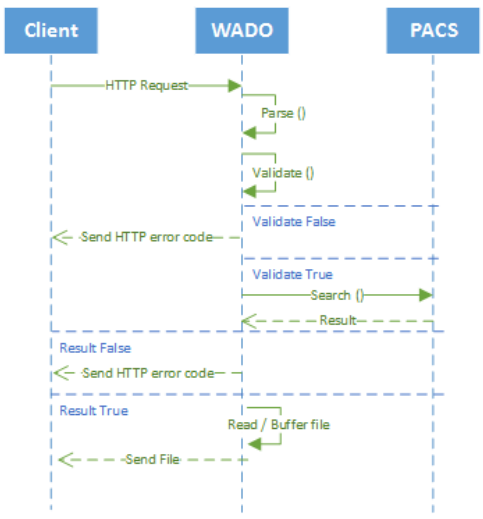


Fig. 3. Secuencia UML de petición al servicio WADO.

5.2.3 Estudio de mercado

Esta fase del proyecto marca un punto de inflexión importante para el desarrollo del mismo. En esta fase se llevó a cabo un estudio de mercado sobre los diferentes tipos de visualizadores DICOM y herramientas similares que existen en el mercado.

El objetivo de este estudio fue determinar la opción más adecuada para el desarrollo del visualizador DICOM basado en web, por lo tanto el foco de búsqueda fueron soluciones basadas en web.

Las soluciones estudiadas fueron principalmente de código abierto. Algunas de las opciones estudiadas se mencionan en el estado del arte en este mismo documento y otras de las que se consideraron se pueden ver en la tabla del anexo.

En el estudio de mercado se consideraron varios aspectos:

a. Capacidad para ejecutarse en entorno web.
b. Compatibilidad con las sintaxis de transferencia[6] (o codificaciones) soportadas por el PACS de MediVector.
c. Capacidad de recuperar objetos vía WADO.
d. Capacidades de manipulación de las imágenes
e. Capacidad de integración (posibilidad o facilidad de adaptar el visualizador al proyecto).

A modo de clarificación, para la capacidad de integración se consideran principalmente dos factores, si la herramienta permite configurarla fácilmente para un PACS existente y si el tipo de aplicación es compatible con un servidor IIS (Internet Information Services) de Microsoft. Este último aspecto sólo se considera por la conveniencia que supone, ya que las aplicaciones del PACS de MediVector están desplegadas sobre un servidor IIS.

Como resultado de este estudio se decidió explorar la herramienta Cornerstone, mencionada en el estado del arte, para crear un visualizador DICOM basado en web. Si bien es una solución que requiere mayor esfuerzo de implementación que otras de las estudiadas, que ofrecen un visor completamente funcional, es una base de desarrollo con muchas capacidades de expansión.

Cornerstone es una librería JavaScript diseñada para mostrar imágenes médicas en un entorno web haciendo uso del elemento Canvas de HTML5.

A nivel de arquitectura está compuesto por tres elementos:

- Núcleo: Como motor de la aplicación, se encarga de los aspectos relacionados con la representación de la imagen; gestionar el cache de las imágenes cargadas, dibujarlas sobre el Canvas, habilitarlas y deshabilitarlas según sea necesario, modificar la representación de los pixeles si el elemento cambia de tamaño, etc.
- Cargador de imágenes: Este componente permite definir el método de transporte y el contenedor que usará la aplicación. Es el encargado de llevar

[6] Sintaxis de trasferencia: En DICOM identifica una sintaxis abstracta. Son las reglas de codificación usadas en comunicaciones DICOM, indica si los datos están comprimidos o no y en que formato.

a cabo la petición de un objeto DICOM e interpretar los datos, por lo que debe ser capaz de decodificar las sintaxis de transferencia apropiadas. El propio proyecto tiene ya implementado un ImageLoader para WADO que permite recuperar un objeto DICOM de un servicio WADO.

- Herramientas: Las herramientas modifican la representación de un objeto. Herramientas típicas aplicadas a un objeto DICOM son; mover, zoom, invertir colores, cálculo de ángulos y líneas, reproducción (si es multi-frame), etc.

La ventaja de este diseño es que la aplicación es agnóstica a las herramientas y cargadores que se usan, por lo que es posible definir nuevas herramientas o cargadores para diferentes usos sin necesidad de reescribir la aplicación base o alguno de los otros componentes.

En este proyecto se parte de las herramientas básicas ofrecidas por Cornerstone para desarrollar el visualizador DICOM ya que ofrecen suficientes capacidades de manipulación para satisfacer los objetivos del proyecto.

5.2.4 Interfaz web

Una vez construidos el servicio WADO y seleccionada una tecnología para la implementación de un visualizador web se inició la fase de diseño de la interfaz web que servirá para explorar el contenido del PACS y lanzar el visualizador.

El diseño de la interfaz se divide en una fase de especificación de requisitos, esbozo de la interfaz y finalmente implementación de la lógica del servidor para integrar la capacidad de búsqueda con el PACS.

El *back-end*, o lógica de servidor, de la aplicación se implementa en lenguaje C# bajo la plataforma ASP.NET siguiendo la arquitectura MVC para facilitar la división de responsabilidades y la mantenibilidad de la aplicación. Los controladores de la aplicación se encargan de traducir la interacción del usuario a peticiones sobre los sistemas de MediVector (para login) y el PACS (para búsquedas).

Para el *front-end*, o lógica de cliente, de la aplicación se usa AngularJS a fin de tratar los datos que devuelve el servidor y modificar el comportamiento de la vista del cliente. El cliente recibe una vista en formato HTML y los resultados de la búsqueda en formato de datos JSON (JavaScript Object Notation). La aplicación de cliente se encarga de presentar los datos de forma dinámica cargando solo aquellas partes de la tabla de resultados con las que interactúa el cliente.

5.2.5 Visualizador DICOM basado en web

Para la fase del visualizador DICOM Systelab ofreció libertad en cuanto al diseño y características, limitando únicamente las herramientas de manipulación de objetos que pueden estar presentes en dicha aplicación. Esto es debido a que la aplicación de herramientas de medición requiere un alto grado de verificación que no se desea para una aplicación que no será usada para fines de diagnóstico médico.

La aplicación de visualización de objetos DICOM está basada en Cornerstone y AngularJS. En esta aplicación se usa Cornerstone obtener y manipular los datos de la imagen mientras que el estado de las herramientas de manipulación y la interfaz se controlan con una aplicación AngularJS.

El funcionamiento del visualizador se explica a continuación.

Cuando un cliente selecciona una imagen desde la interfaz web, la aplicación del visualizador DICOM se lanza en una nueva ventana.

En la parte del servidor el controlador responde con una vista que contiene la interfaz del visualizador DICOM, una referencia HTTP a un objeto DICOM y a una aplicación basada en AngularJS que controla la aplicación Cornerstone.

Cuando la vista ha cargado en el cliente, la aplicación del visualizador pasa la referencia HTTP al cargador WADO de Cornerstone, que pide el objeto y lo muestra.

5.3 Resultados

Una vez completados los objetivos del proyecto tendríamos una arquitectura con capacidades ampliadas, habilitando servicios web mediante WADO y la aplicación Web. En la figura 4 puede verse un diagrama del estado tras un despliegue teórico.

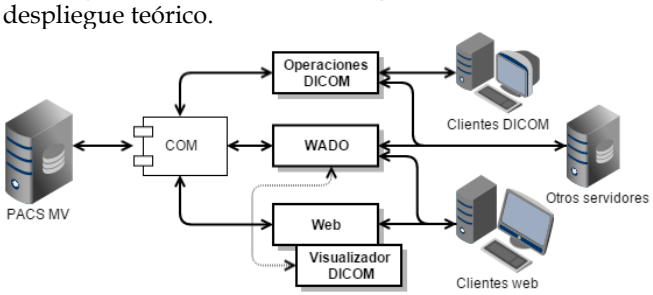


Fig. 4. Despliegue de tecnologías web en el PACS.

5.3.1 Servicio WADO

Para mostrar los resultados del servicio WADO usamos la aplicación Postman. Esta aplicación es una extensión de google Chrome que permite simular peticiones HTTP(S) con todo tipo de parámetros, cookies, autenticación, etc. y visualizar el resultado. En las figuras 5 y 6 podemos ver los resultados de enviar una petición al servicio WADO especificando una instancia de un objeto real y el tipo de respuesta que deseamos como parámetro "contentType".

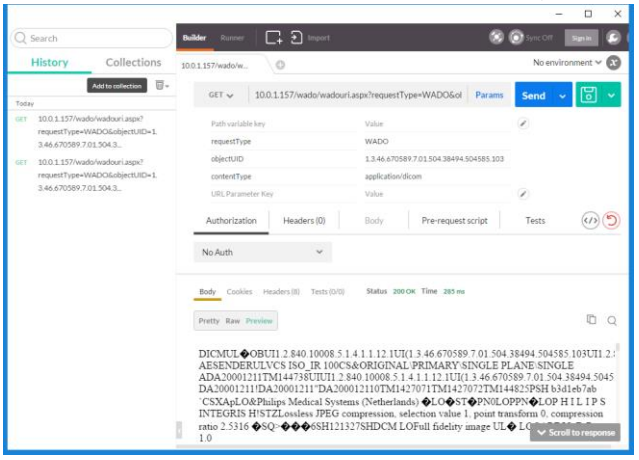


Fig. 5. Petición de objeto DICOM.

La primera petición es para un objeto DICOM mientras la segunda es para un video en formato mp4.

Como el servicio es capaz de interpretar y contestar a las peticiones adecuadamente esta parte del proyecto puede considerarse completada de forma satisfactoria.

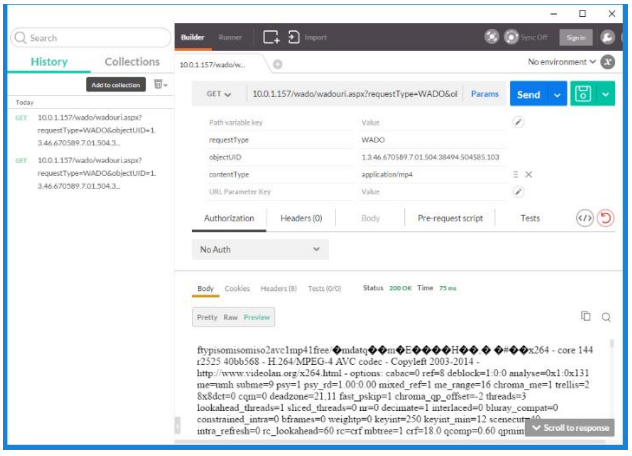


Fig. 6. Petición de objeto mp4

Como prueba de concepto se incorporó al servicio una página web simple que recibe las mismas peticiones que el servicio WADO pero responde a la petición con una página HTML que hace uso del elemento <video> de HTML5 para reproducir el video en el navegador sin necesidad de herramientas adicionales.

Esta página se probó con éxito en los navegadores más comunes; Internet Explorer, Mozilla Firefox, Google Chrome y Opera.

Esta misma página se incorporará a la aplicación web como método de visualización alternativo al visualizador de objetos DICOM. El resultado será similar al del visualizador de objetos DICOM pero limitado a los controles típicos de un reproductor de video.

5.3.2 Interfaz Web

Seguidamente pasamos a mostrar los resultados del desarrollo de la interfaz.

La implementación de la interfaz sigue un Diseño Web Adaptable[7] que se ha creado haciendo uso del sistema de rejilla de Bootstrap. Las versiones móviles de la interfaz pueden verse en el anexo.

La página por defecto de la aplicación web es la página de login, como esta página aporta poca información sobre el proyecto, solo está incluida en el anexo de las versiones móviles.

La página principal de la aplicación es la página de búsqueda que aparece en la figura 8. En esta página el usuario tiene acceso a un formulario que permite introducir datos correspondientes a los que el servidor PACS almacena sobre los pacientes. Estos campos son; nombre, apellidos, número de historia clínica, numero de acceso, modalidad de diagnóstico y un rango de fechas.

[7] Diseño Web Adaptable: Responsive Web Design, es una forma de diseño web que permite a la interfaz adaptarse automáticamente al tamaño del dispositivo con el que se visualiza.

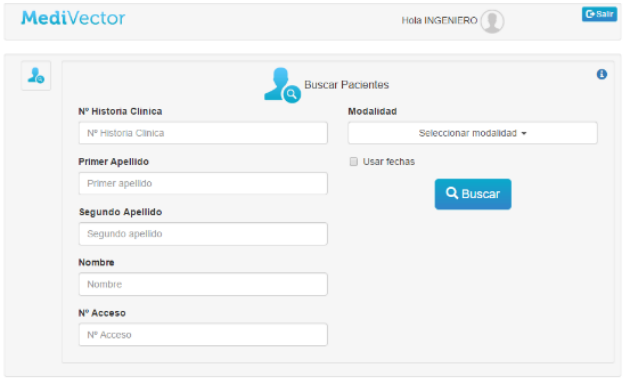


Fig. 8. Formulario de búsqueda.

Finalmente la página de resultados en la figura 9 muestra los datos obtenidos en la búsqueda en tablas anidadas. Los niveles de estas tablas se corresponden con la jerarquía DICOM así que cada paciente tiene múltiples estudios, cada estudio múltiples series y cada serie múltiples instancias (imágenes médicas).

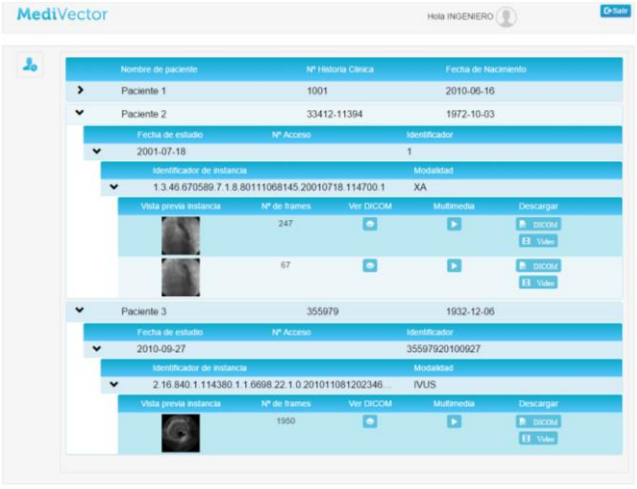


Fig. 9. Tabla de resultados.

5.3.3 Visualizador de objetos DICOM

Finalmente mostramos el visualizador de imágenes DICOM.

En la figura 10 puede verse una imagen de modalidad XA (Angiografía con rayos X) cargada en el visualizador DICOM. Este tipo de imagen es de las más comunes generadas en el área de Cardiología del hospital donde se encuentra el PACS de MediVector.

En su estado actual la interfaz es tan sólo un prototipo, pero las funcionalidades que se tomaron como objetivo han sido implementadas.

El visualizador permite reproducir un objeto multiframe, seleccionar imágenes con un selector de rango, seleccionar la velocidad de reproducción y manipular la imagen con herramientas básicas (contraste, invertir color, ampliar, mover).

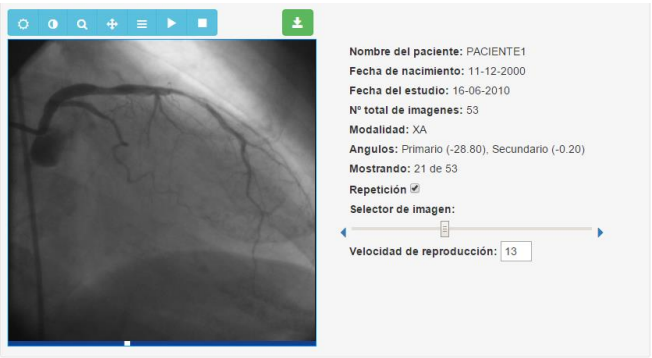


Fig. 10. Prototipo de visualizador DICOM

Adicionalmente puede descargarse la imagen contenida en el Canvas, con las modificaciones que se hayan aplicado (contrastes, zoom, etc).

5.3.4 Medidas de seguridad

Finalmente, debido a que esta aplicación trata con objetos que almacenan información de carácter médico sobre los pacientes, se aseguran las trasferencias de datos de los diferentes componentes de la aplicación haciendo uso de SSL.

Para proteger el acceso a estos medios se han habilitado las siguientes medidas de seguridad:

- Comunicación mediante SSL: Tanto el servicio WADO como la interfaz web establecen sus comunicaciones usando HTTPS.
- Credenciales: Para poder acceder a la interfaz web se ha hecho imperativo la necesidad de que el usuario aporte unas credenciales que sean verificables en el sistema de MediVector. Para implementar esto se ha hecho uso de la etiqueta[8] "authorize" tanto en los controladores como en las vistas de la interfaz de la web. En MVC de asp.net la etiqueta "authorize" bloquea el acceso a aquellos métodos y clases que decora ante usuarios que no hayan sido registrados previamente en la aplicación de autenticación. En esta aplicación esta etiqueta se usa como filtro global con el fin de proteger todos los controladores. En el anexo puede observarse el resultado de intentar acceder a una página de la interfaz sin autorización. [18]

6 Conclusiones

Una vez finalizado el proyecto, se han cumplido los tres objetivos planteados inicialmente y como resultado se han obtenido tres componentes individuales que pueden funcionar en conjunto para cumplir el objetivo global del proyecto.

Recapitulando sobre las tareas llevadas a cabo podemos analizar principalmente dos aspectos del mismo. El primero son las decisiones técnicas como las elecciones en tecnologías de desarrollo. El segundo son los resultados de la investigación en la búsqueda de soluciones para mostrar imágenes DICOM en entorno web.

Desde un punto de vista técnico se está satisfecho con los resultados obtenidos con las tecnologías elegidas para el desarrollo de las diferentes partes. En la parte de servidor la plataforma ASP.NET de Microsoft ha facilitado considerablemente la implementación de seguridad y la aplicación de la arquitectura MVC en la aplicación. En la parte de cliente el framework AngularJS ha permitido controlar y manipular con facilidad la información devuelta por el servidor y la aplicación del visualizador de objetos DICOM.

Sobre el uso de AngularJS cabe destacar que tiene una curva de aprendizaje que supuso un cierto contratiempo durante el desarrollo del proyecto. Por esta razón podría cuestionarse si era necesario recurrir a un framework para una aplicación de cliente relativamente sencilla, pero no fue el caso. Este framework fue especialmente útil a la hora de controlar los datos de la aplicación de visualización de objetos DICOM basada en Cornerstone.

Si bien estas herramientas añaden una capa de complejidad adicional al proyecto, una vez que se es familiar con su uso aceleran considerablemente la construcción de un entorno web y facilitan su mantenibilidad.

Adicionalmente, aunque el foco del desarrollo de este proyecto no son los dispositivos móviles el uso exclusivo de herramientas compatibles con exploradores web, especialmente Bootstrap, hace que el proyecto sea fácilmente escalable y ampliable en esa dirección.

En general el progreso del proyecto a nivel de desarrollo ha sido satisfactorio, aunque quizá con una mejor planificación y previsión de las tareas podría haber llegado más lejos, especialmente en lo que respecta a las capacidades del visor de objetos DICOM.

Finalmente un aspecto muy importante del proyecto es el resultado satisfactorio de la investigación sobre las soluciones para mostrar imágenes DICOM en entorno web.

Con la investigación llevada a cabo en el transcurso del proyecto se ha determinado que existen multitud de opciones para llevar a cabo el desarrollo de una aplicación web de visualización de objetos DICOM, tanto si se decide desarrollar una solución propia con la ayuda de alguna herramienta o framework o si se decide optar por la vía comercial.

7 Tareas para el futuro

Pese a que se ha implementado seguridad en las comunicaciones de los diferentes servicios del proyecto mediante SSL, el servicio WADO aún no ha sido completamente protegido.

Por lo tanto en su estado actual el servicio WADO es vulnerable a ataques. Esto se debe a que el servicio actualmente no requiere incluir acreditación en las peticiones para determinar si el cliente está autorizado a llevar a cabo una petición. Es un aspecto pendiente que debe ser solucionado antes de poder desplegar la aplicación en un entorno real, si no se expondría información especialmente protegida ante posibles atacantes.

Directamente relacionado con WADO está la implementación de un servicio QIDO (Query based on ID for DICOM Objects). Este servicio complementa a WADO (que

[8] Las etiquetas en ASP.NET modifican el comportamiento de un controlador o método.

solo permite recuperar objetos) permitiendo la búsqueda de identificadores de estudios e instancias mediante la identidad de un paciente.

Por otro lado el visualizador DICOM está limitado a una sola imagen por instancia de la aplicación. La vía de desarrollo más lógica para mejorar la aplicación es permitir gestionar múltiples imágenes en una sola instancia para facilitar la visualización de estudios en un entorno médico.

AGRADECIMIENTOS

Me gustaría agradecer a Joan Serra Sagristà, tutor de mi proyecto, su ayuda con comentarios e ideas para mis documentos pese a que le enviara mis borradores siempre en el último momento.

También me gustaría agradecer a Jaume Estaun, de Systelab, su apoyo y ayuda diarios con todas mis dudas y consultas.

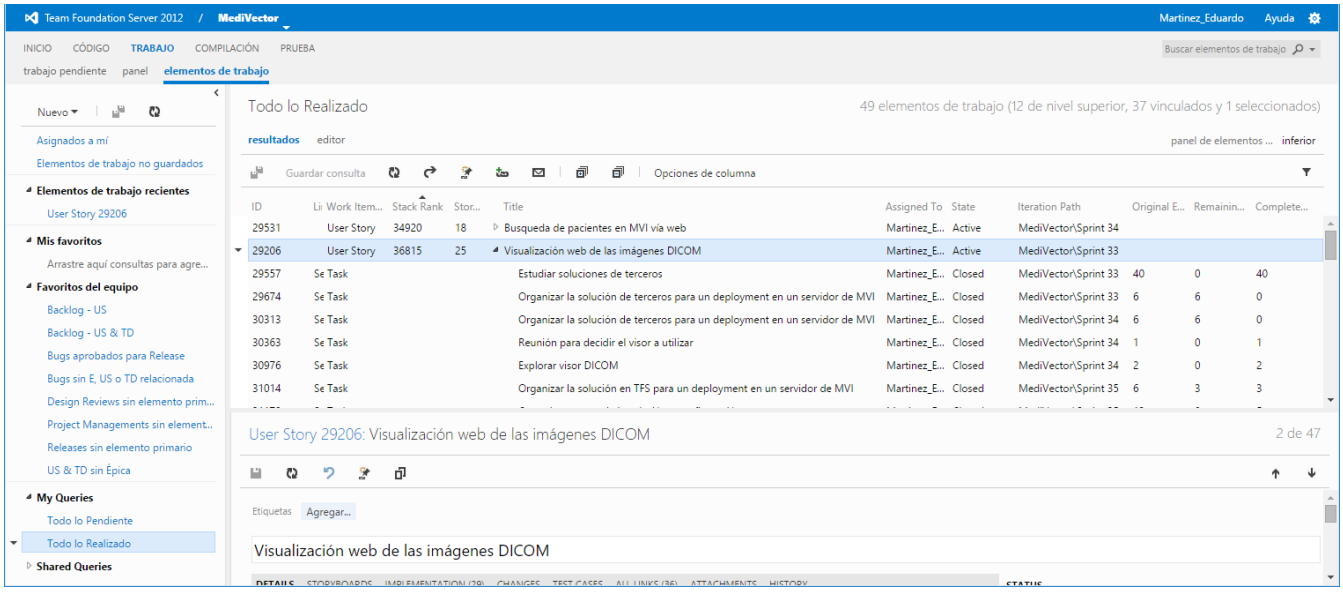
Finalmente agradecer al resto del equipo de MediVector en Systelab la oportunidad de formar parte de su equipo y llevar a cabo este proyecto.

BIBLIOGRAFIA

[1] DICOM standard, History. Web, accedida 24 enero. http://dicom.nema.org/medical/dicom/current/output/html/part01.html#sect_History
[2] DICOM standard, Data dictionary. Web, accedida 24 enero. http://dicom.nema.org/Dicom/2011/11_06pu.pdf
[3] DICOM standard, Status of updates. Web, accedida 24 enero. http://www.dclunie.com/dicom-status/status.html
[4] Dm4che, Open Source Clinical Image and Object Management. Web, accedida 24 enero. http://www.dcm4che.org/
[5] Orthanc, Open-source, lightweight DICOM server. Web, accedida 24 enero. http://www.orthanc-server.com/
[6] Cornerstone. Web, accedida 24 enero. https://github.com/chafey/cornerstone
[7] DICOM web Viewer. Web, accedida 24 enero. http://ivmartel.github.io/dwv/
[8] Leadtools HTML5 Zero-footprint Medical Viewer SDK. Web, accedida 24 enero. https://www.leadtools.com/sdk/medical/html5
[9] Component Object Model. Web, accedida 24 enero. https://www.microsoft.com/com/default.mspx
[10] Agencia española protección de datos, datos especialmente protegidos, "Análisis de diferentes cuestiones relativas a la historia clínica". Web, accedida 24 enero. http://www.agpd.es/portalwebAGPD/canaldocumentacion/informes_juridicos/datos_esp_protegidos/index-ides-idphp.php
[11] ASP.NET MVC. Web, accedida 24 enero. http://www.asp.net/mvc
[12] AngularJS. Web, accedida 24 enero. https://angularjs.org/
[13] Bootstrap CSS. Web, accedida 24 enero. http://getbootstrap.com/css/
[14] W3schools HTML5 intro. Web, accedida 24 enero. http://www.w3schools.com/html/html5_intro.asp
[15] Malcolm Sheridan, "The Developer's Guide to HTML5 Canvas". Web, accedida 3 de febrero. https://msdn.microsoft.com/en-us/hh534406.aspx
[16] HTML5 video and MP4 support. Web, accedida 4 febrero. http://www.w3schools.com/html/html5_video.asp
[17] WADO parameters. Web, accedida 24 enero. http://dicom.nema.org/medical/dicom/current/output/html/part18.html#chapter_8
[18] Asp net authorize attribute. Web, accedida 24 enero. https://msdn.microsoft.com/en-us/library/system.web.mvc.authorizeattribute%28v=vs.118%29.aspx#Anchor_6

Anexo

A1. TEAM FOUNDATION SERVER



Captura de pantalla de la aplicación Team Foundation Server. En la captura aparece un panel con las tareas completadas dentro del proyecto.
Para cada tarea se especifican las relaciones con otras tareas, para que sprint está previsto completarla, las estimaciones de dificultad, tiempo de implementación previsto y tiempo total invertido en su desarrollo.

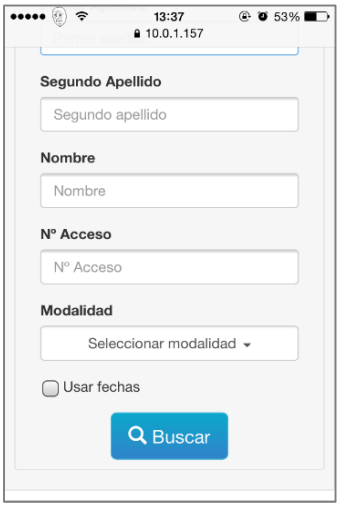
A2. TABLA DE ESTUDIO DE SOLUCIONES

Nombre	web	Soporte sintaxis	WADO	Manipulación	integración	Licencia
Cornerstone	Si	1, 2, 3	Si	Avanzada	No	MIT
DWV	Si	2, 3			No	GNU GPL 3
Leadtools HTML5 ZMV	Si	1, 2, 3	Si	Avanzada	Si	Comercial
Orthanc web viewer	Si	1, 2, 3	Si	Básica	No	AGPL
Oviyam2	Si	1, 2, 3	Si	Avanzada	Si	MPL1.1,GPL2.0,LGPL2.1
Papaya	Si	-	No	Básica	No	BSD
WEASIS	Parcial	1, 2, 3	Si	Avanzada	SI	EPL
JSdicom	Si	-	No	Básica	No	GPL
webMango	Si	-	No	básica	No	Solo uso científico

Web: Indica si puede ejecutarse íntegramente en web. En el caso de WEASIS se especifica parcial debido a que requiere que Java esté instalado en el equipo.
Soporte de sintaxis y compatibilidad: Campos que indican cuales de las sintaxis de codificación que usa el PACS de MediVector también soporta la aplicación de visualización DICOM. Los tres formatos son: 1-RLE Lossless (1.2.840.10008.1.2.5), 2-JPEG Baseline, (1.2.840.10008.1.2.4.50), 3-JPEG Lossless (1.2.840.10008.1.2.4.70). Todas las sintaxis de compresión en DICOM usan *explicit VR little endian* (1.2.840.10008.1.2.1) para la ordenación de bytes.
WADO: Indica si la aplicación puede recuperar objetos vía WADO.
Manipulación: Señala los diferentes niveles de interacción que posibilita el visualizador. Básica corresponde a operaciones como aumentar, mover la imagen, ajustar contrastes y en algunos casos medir distancias. Las funcionalidades avanzadas incluyen todas las anteriores además de la capacidad de reproducir objetos multi-frame y calcular ángulos.
Integración: Indica si la herramienta incluye alguna aplicación que facilite la integración con un PACS existente.

A3. VERSIONES ADAPTABLES DE LA INTERFAZ

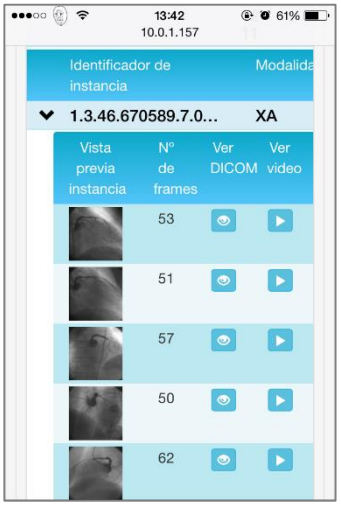
Login



Búsqueda parte 1

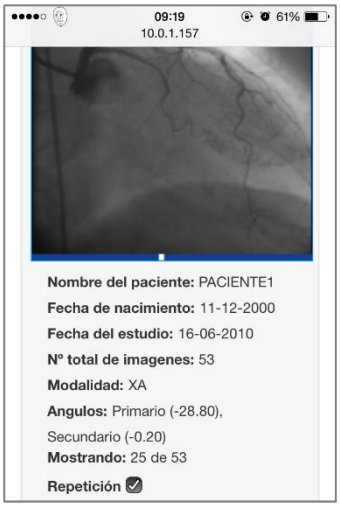
Búsqueda parte 2

Resultados parte 1



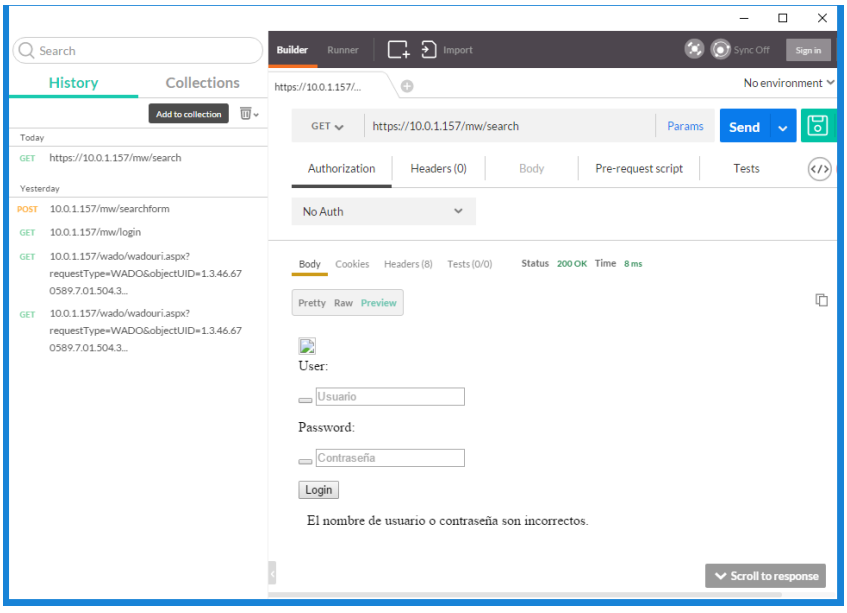
Resultados parte 2

Prototipo visualizador parte 1



Prototipo visualizador parte 2

A4. Redirección a login



Resultado de una petición a una sección de la web sin estar autorizado. En este caso la petición se hace a la página de búsqueda pero el servidor responde con la página de login.