

DICOM:

Protocolo de red DICOM: todas las aplicaciones de imágenes médicas que están conectadas a la red del hospital utilizan el protocolo DICOM para intercambiar información, principalmente imágenes, pero también información sobre el paciente y el procedimiento. El protocolo DICOM es un protocolo de nivel superior binario (ULP, Upper Level Protocol) sobre TCP / IP. Los puertos utilizados por DICOM son 104, 2761, 2762 y 11112. El protocolo dispone de múltiples tipos de mensajes destinados a transmitir, buscar/consultar, integrar, distribuir, imprimir, compartir, almacenar y mostrar imágenes médicas. También hay comandos de red más avanzados que se utilizan para controlar y seguir el tratamiento, programar procedimientos, informar estados y compartir la carga de trabajo entre médicos y dispositivos de imágenes. DICOM utiliza el protocolo TCP/IP (protocolo usada para comunicación de dispositivos en redes), y agrega su propio lenguaje de red: -- ≥servicios de alto nivel (DICOM DIMSE) construidos sobre un lenguaje de bajo nivel (DICOM ULP).



Las definiciones de estos objetos y atributos están estandarizadas de acuerdo con **las definiciones contenidas dentro de las DICOM Information Object Definitions (IODs)**. Los IODs son conjuntos de atributos que describen las propiedades de un IOD. Un IOD de paciente, por ejemplo, se puede describir por nombre del paciente, ID, sexo, edad, peso, estado de tabaquismo, etc.. DICOM **mantiene una lista de todos los atributos estándar (más de 2.000 de ellos), conocida como DICOM Data Dictionary (DDD)**, DICOM es una arquitectura apoyada en instancias, un concepto que es probable que resulte familiar a quien haya trabajado con programación orientada a objetos. De forma simple, una instancia es una copia de un objeto y todas sus propiedades, realizada a partir de un objeto inmutable. Cualquier cambio en el objeto siempre produce una nueva instancia, es decir, una nueva copia del objeto.

Los datos de la imagen (Raw Data) aparecen en el archivo “.dcm” como un campo más en el DICOM, con el campo (7FE0,0010) y el nombre “Pixel Data”. Podemos ver que algunos campos son del tipo “UI” y tienen nombres terminados en UID. Estos son conocidos en DICOM como “Unique Identifiers” ¿Como trabajamos con python y DICOM?

Pydicom es un paquete Python para trabajar con archivos DICOM. Permite leer, modificar y escribir datos DICOM de una manera sencilla. Pydicom puede ejecutarse en cualquier lugar donde se ejecute Python sin ningún otro requisito, aunque cuando trabajemos con los Pixel Data, normalmente necesitaremos otros paquetes como Numpy. Múltiples ejemplos de uso de pydicom se pueden encontrar en su documentación: General examples — pydicom 2.2.0 documentation Pynetdicom es una implementación Python del protocolo de comunicaciones DICOM. Trabajar con pydicom permite la creación sencilla de usuarios de clase de servicio (SCU) y

proveedores de clase de servicio (SCP). La clase de usuario principal de Pynetdicom es AE y se utiliza para representar una entidad de aplicación DICOM. Con él puedes:

** Iniciar la aplicación como un SCP especificando los contextos de presentación admitidos y luego llamar a `AE.start_server ()` y establecer conexiones con otros servicios respondiendo a las solicitudes de asociación entrantes.

** Iniciar la aplicación como un SCU especificando los contextos de presentación que desea que admita el SCP, luego solicitar una asociación a través del método `AE.associate ()`, para, por ejemplo, poder solicitar la descarga de imágenes desde un servidor PACS enviando mensajes C-FIND y C-GET.

Pynetdicom facilita el envío y recepción de estos datos médicos entre dispositivos y aplicaciones, apoyando funciones como la consulta, recuperación y almacenamiento de imágenes médicas, así como otros servicios relacionados con DICOM. Es especialmente útil para desarrolladores que trabajan en el área de tecnologías de la salud, permitiéndoles integrar funcionalidades DICOM en sus aplicaciones de manera eficiente

Cada dispositivo dentro de una red TCP/IP recibe una dirección IP, a través de la cual otros dispositivos en la red pueden encontrarlo. **Cada DICOM Application Entity (AE) dentro de una red se identifica por su dirección IP, ya sea un servidor DICOM/PACS (almacenamiento de imágenes), una estación de trabajo, o un escáner.** Además de su dirección IP, DICOM asigna a cada AE un identificador conocido como Application Entity Title (AET), y suele ser un identificador comprensible para los humanos, como por ejemplo, "PACSSERVER" o "MRSCANNER"

A partir de aquí estaremos listos para solicitar a un servidor PACS que nos envíe, por ejemplo, imágenes que cumplan una serie de criterios. Una de las principales ventajas de DICOM es la capacidad de buscar datos específicos mediante mensajes "DICOM C-Find". Los mensajes C-Find son objetos DICOM con la misma estructura de parámetros. Los valores de atributo de nivel de query actual serán devueltos por C-Find. En el ejemplo anterior, una búsqueda a nivel de estudio, siempre se devolverá al menos una lista de "Study UIDs" (0020,000E) en respuesta a cualquier solicitud de C-Find. Esto nos permite pasar a los niveles de búsqueda inferiores (Serie, Imagen). Diferentes niveles de C-Find nos permitirán confeccionar una lista de los estudios que queremos descargar.

Para descargarlas, recurriremos a otro tipo de mensajes, los mensajes C-Get, con una arquitectura similar. Aunque ciertamente C-Get permite realizar búsquedas al estilo de C-Find, el procedimiento correcto es utilizar C-Find para obtener una lista de objetos a descargar, y después descargarlos utilizando C-Get. Existe una diferencia principal entre los IOD C-Find y C-Get: la SCU C-Get debe proporcionar valores clave únicos para identificar una entidad y recuperarla. Es decir, en lugar del enfoque de relleno con máscaras que vimos en C-Find, C-Get necesita las claves únicas para el nivel actual y superior para identificar las imágenes para su recuperación.

