

# **Integración en una página web de un método de recogida de imágenes médicas para su uso en algoritmo de IA**

Sergio Díaz Prados

# Índice

- Abreviaturas del trabajo.....	3
- Objetivos del trabajo.....	4
- Capítulo 1: Radiología presente y futuro.....	5
- Capítulo 2: Esquema general del proyecto.....	8
- Capítulo 3: Marco previo del proyecto.....	13
- Capítulo 4: Registro de usuarios.....	14
- Capítulo 5: Edición de una entrada.....	15
- Capítulo 6: Visualización de la entrada.....	18
- Capítulo 7: Imágenes DICOM.....	21
- Capítulo 8: Creación de un algoritmo.....	25
- Capítulo 9: Perspectiva de futuro.....	27
- Capítulo 10: Escalabilidad del proyecto.....	30
- Agradecimientos.....	32
- Bibliografía.....	33
- Anexos.....	34

## Abreviaturas

- IA: Inteligencia Artificial
- RM: Resonancia Magnética
- TC: Tomografía Computarizada.
- CIE-11: Clasificación Internacional de Enfermedades-11
- ML: Machine Learning.
- PLN: Procesamiento del Lenguaje Natural
- DICOM: Digital Imaging and Communications in Medicine
- PACS: Picture Archiving and Communication System
- SAS: Servicio Andaluz de Salud
- RIS: Sistema de gestión de Información Radiológica
- API: Application Programming Interface
- RAM: Random Access Memory
- AET: Application Entity Title

## Objetivo

El objetivo del presente trabajo consiste, en la integración en una página web ya existente de divulgación de imagen médica; <https://radiologymodels.com/>, la posibilidad de que los usuarios puedan subir entradas con fin didáctico, y que adicionalmente, dichas entradas puedan ser utilizadas, si el usuario ha dado su permiso, para el desarrollo de un algoritmo de inteligencia artificial (IA). En concreto, dicho algoritmo versará sobre patología hepática.

Aunque lo primordial ha sido crear un marco interactivo que permita tanto obtener datos médicos fiables para su posterior uso, como la divulgación de conocimiento sobre información radiológica, he querido exponer someramente como sería el comienzo para poder usar dicha información a través de un prototipo de algoritmo de IA.

Por tanto se trata de un trabajo enfocado a recalcar la importancia de los pasos previos al desarrollo de un algoritmo de IA. Disponer de una información estandarizada y correctamente etiquetada, a la vez que se ofrece un uso de la durante el desarrollo del algoritmo, de tal forma que se pueda aprovechar para el uso general.

Actualmente me encuentro desarrollando mi labor como residente en el departamento de Radiodiagnóstico en Granada. Con este trabajo y gracias al conocimiento adquirido en este Máster y durante este periodo de tiempo pretendo realizar este proyecto con el fin de acercar el desarrollo de la IA al conjunto de residentes de mi especialidad, en mi hospital y en otros lugares. Ha sido una tarea compleja por la transversalidad del estudio, teniendo que tener como guía lo expuesto en nuestro curso y profundizar en conocimientos de programación web de forma relativamente autodidacta.

Mi mayor interés ha sido crear una herramienta que permita obtener datos médicos fiables (proporcionado por médicos especialistas en imagen), de forma amena y entretenida y teniendo en cuenta su futuro provecho.

# 1. Capítulo 1: Radiología presente y futuro

## 1.1 El diagnóstico por imagen

Para la consecución de un exitoso sistema de IA necesitamos del desarrollo de una base de datos amplia pero también de una calidad que se ajuste al rendimiento que esperamos tener. En la práctica médica dicho rendimiento por lo general, se espera que sea muy elevado.

La especialidad de radiodiagnóstico es la especialidad médica enfocada al diagnóstico de múltiples patologías guiado por imagen. Para ello se emplean numerosas técnicas, algunas que emplean radiación ionizante como el caso de la radiografía o la tomografía por ejemplo, y otras técnicas que no emplean radiación, como la ecografía y la resonancia magnética, sobre todo.

Se trata de una especialidad transversal y central en un hospital debido a que muchas otras especialidades necesitan la información que proporcionan sus pruebas para llegar a un tratamiento médico correcto, como puede suceder en caso de patologías pulmonares, digestivas, cardíacas, musculoesquelético, nerviosas....

Actualmente las necesidades de estudios radiológicos son muy elevadas. De hecho se realizaron más de 40 millones de pruebas de imagen al año en España en 2022 (SERAM, 2022). Pero esto no tiene aviso de estancarse, si no que se ha previsto que se produzca un incremento mucho mayor de las mismas. Esto se debe a varios aspectos como pueden ser: que todo acto médico debe haber sido respaldado previamente con todas las herramientas disponibles para disminuir todo riesgo asociado, por otra parte, el concepto de “medicina defensiva”, que es una realidad que provoca que en ocasiones los profesionales sanitarios respalden en exceso sus actos. Otra causa del aumento en los próximos años de estudios radiológicos se debe a como la tecnología ha evolucionado, de forma que el rendimiento de muchas de estas pruebas ha desplazado a otros métodos en la detección de la patología. Adicionalmente esto también ha producido que se planteen nuevos screening a los ya presentes (Hsu & Hoyt, 2019), (Balsera, 2022).

También quería resaltar aquí que dicha especialidad no está exenta de la carencia de profesionales que se está produciendo en otras ramas médicas (Fariña & Fariña, 2022).

Esto tiene a su vez varios factores, que no son de nuestro interés en este trabajo, pero entre los que destaca el envejecimiento poblacional del conjunto general de gran parte de los sanitarios, sin un reemplazo ajustado a dicho proceso. Esto sumado a lo anteriormente descrito hace que o bien se producen cambios en la formación MIR que permita incrementar el número de plazas o lo que tiene más sentido, teniendo en cuenta la evolución tecnológica, es que además de ajustar la plantilla con los cambios demográficos que se van a producir, ir incorporando de forma paulatina estrategias de automatización de procesos.

En primer lugar, la automatización tiene más sentido en aquellos procedimientos monótonos y no tan médicos. Se ha visto que gran parte del tiempo los radiólogos y otros médicos pasan en su trabajo es realizando tareas de tipo “administrativo”. También existen muchos aspectos de su trabajo que son repetitivos, en el caso de la toma de mediciones de lesiones, comparación de estudios actuales y previos o el conteo de ciertos elementos. Sin embargo, el tiempo a la interpretación y a la investigación suele ser mucho más reducido de lo esperado.

El uso de algoritmos, que faciliten algunos de los aspectos del proceso diagnóstico, ya se utilizan en nuestros hospitales y la tendencia es que se sigan incrementando en los próximos años. Alguno de estos ejemplos se da en la detección de lesiones de aspecto sospechoso en el screening de cáncer de mama, otro caso mucho más extendido consiste en la clasificación de las radiografías en normales o no, reservando la visualización de las mismas solamente para aquellos casos que no sean caracterizadas como normales.

Como hemos podido comprobar en este curso, el uso generalizado de la IA permite reducir el tiempo destinado por estudio y permitir enfocar la atención de los lectores sobre la anomalía en la imagen disminuyendo la fatiga asociada a la lectura de los casos, sin restar exactitud en la interpretación como ya se ha podido comprobar en numerosos estudios al respecto

Por otra parte, antes de pasar al siguiente capítulo, también quería destacar brevemente como es el aprendizaje en una unidad de Radiodiagnóstico (en la sección de radiodiagnóstico; no incluyendo intervencionismo). En este aspecto en el tiempo que llevo de residencia he podido comprobar como el conocimiento esencial para realizar en informes radiológico proviene en su gran mayoría de la información existente en

internet en sus diversas formas, ya sea a partir de revistas médicas, y de imagen médica especializadas como pudiera ser *Radiographics*, como también en otras páginas webs de divulgación como por ejemplo la *Radiopaedia*. Estos últimos recursos resultan de gran interés debido al propio objeto de estudio de la radiología diagnóstica. Nos encontramos con que toda la información que debemos exponer en nuestro informe radiológico se encuentra ya en sí misma en el estudio en las imágenes que estamos visualizando, por tanto, el hecho de tener información verificada, contrastada y accesible, en cualquier momento y lugar, permite la comparación con casos representativos de diferentes enfermedades y cómo se ajustan a nuestro caso particular (Maleck et al., 2001).

Por tanto el actual trabajo pretende aspirar a dos objetivos: ser una opción más de las que ya hay en el panorama formativo para el residente, y el conjunto de los profesionales dedicados al análisis de imagen médica y, por otra parte, ser un instrumento para futuras investigaciones, permitiendo la automatización de los procesos diagnósticos, en concreto, como se expondrá más adelante, sobre la patología hepática. Para ello, mi trabajo se centra sobre todo en crear un repositorio de imágenes médicas perfectamente etiquetadas, que puedan servir en un futuro cercano para alimentar algoritmos desarrollados por terceras empresas.

Como hemos estudiado, a pesar del increíble potencial de los datos disponibles, el desarrollo de modelos predictivos es complicado ya que requieren de la creación de conjuntos de datos personalizados con variables específicas. Además debido a que este proyecto se inicia desde las etapas más precoces de la elaboración de un algoritmo de IA el volumen de datos inicial no va a ser muy elevado lo que va a limitar en un primer momento la realización de un modelo con resultados significativos.

No obstante he querido a fin de exponer y dar uso aparte de los conocimientos adquiridos durante este Máster el esquema general de cómo sería la elaboración de un algoritmo de inteligencia artificial que permita identificar estructuras sencillas.

Esto es así porque considero que para poder realizar herramientas útiles en el ámbito de la IA primero nos debemos centrar en una correcta recopilación de la información, que a mi parecer es una de las piezas que más falla en estos procesos. También creo que lo que es realmente útil, teniendo en cuenta que grandes compañías se están introduciendo en este ámbito, como pudiera ser por ejemplo, Google, entre otras, es disponer, como

digo de una correcta información para abastecer los cálculos necesarios y llevados a cabo por estas empresas.

## Capítulo 2: Esquema general del proyecto

Según lo comentado anteriormente y basándome en tres pilares principales que guían el proyecto: divulgación de información médica, incentivar la formación entre los residentes y utilidad para investigación futura, quiero presentar a continuación un diagrama que presente de forma sencilla los pasos a seguir para cumplir con nuestros objetivos. Antes de nada, resaltar que se ha querido seguir adecuadamente con los pasos mostrados en los capítulos expuestos en nuestro curso, en la elaboración de un proyecto de IA, teniendo en cuenta cuales son los pasos habituales que siguen los datos médicos hasta generar modelos predictivos:

1. Recogida de datos: Como he comentado anteriormente tenemos a nuestro favor que los datos médicos provienen de una única fuente; un formulario Django de carácter multidatos, lo que facilita su uso.

En este apartado quería poner de manifiesto una dificultad intrínseca de nuestro proyecto. Aunque se trata de un recurso en web disponible para cualquier profesional médico del mundo, inicialmente nuestro proyecto va a ser utilizado principalmente por profesionales de nuestro entorno lo que provoca que pueda existir a la hora de utilizar los datos para nuestro algoritmo una mayor representación de la población de esta región. Esto a la larga pueda revertirse si este recurso empieza a ser utilizado por profesionales de la imagen de otros países.

Los usuarios reciben información clara y esencial del proceso a seguir, para que tengan en cuenta, a quien va dirigida las imágenes e información médica que depositan en la página, incluyendo un apartado aclaratorio sobre uso futuro en IA. También se informa a los usuarios de las modificaciones necesarias en caso de que no se cumplan con los requisitos expuestos por la página. También se les informa de que serán avisados por correo electrónico de los avances que se estén produciendo en el desarrollo de algoritmo de IA.



Como los datos se van a ir recopilando de forma paulatina se nos ofrecen de forma natural varios puntos positivos, como por ejemplo el hecho de ir entrenando el algoritmo “poco a poco” pudiendo evaluar problemas que puedan ir surgiendo como la evaluación de riesgos de reentrenamiento o evitar riesgos de sobreajuste.

2. Almacenamiento: Se lleva a cabo en una única base de datos, con una moderada capacidad de almacenaje y rápido acceso desde cualquier ordenador que tenga acceso a ella. En nuestro caso, he utilizado la base de datos *sqlite3* en el entorno de desarrollo local, debido a que se trata de una opción de la que existe bastante información en Internet y cuya integración en el entorno es sencilla. Por otra parte a la hora de subir el proyecto real he querido utilizar una base de datos, que por lo que he podido contrastar, es mucho más escalable. Se trata de *PostgreSQL*, un poco más compleja que la anterior, pero con muchas opciones. He utilizado para acceder a ella el adaptador *postgresql\_psycopg2* con las medidas correspondientes de seguridad, en el ámbito del uso de contraseñas, especificaciones de puerto y el e-mail. Todos estos parámetros quedan recogidos en el archivo *env*.

3) Anotación: Las imágenes y los campos de textos son datos totalmente diferentes. Pero en algún momento la información de texto tiene que confluir con las imágenes para darles un sentido. Para ello, aparte de asignar a las imágenes una patología en concreto, mediante el sistema de clasificación internacional, disponemos de campos opcionales de anotación médica que se pueden llegar a interpretarse, si se necesitase, mediante sistemas de procesamiento de lenguaje natural (PLN).

La anotación consiste en aplicar una serie de estándares de normalización y armonización con el fin de homogeneizar los datos.

3. Integración: La integración tanto de las imágenes médicas como de los campos de texto asociados es muy sencilla. Esto se debe a que ambos tipos de información van a corresponder a *atributo* de un mismo *class model*. El *modelo* que posee las imágenes DICOM (Digital Imaging and Communications in Medicine), a su vez, posee de diferentes *atributos* que permiten el fácil acceso a ellas permitiendo que cada objeto de imagen tenga su identificador y sus campos de información perfectamente asociados.

4. Análisis: Último paso llevado a cabo. Es obviamente el que proporciona ese valor adicional en nuestro trabajo. Se expondrá en un apartado específico.
5. Pruebas de rendimiento: Suelen ser un paso habitual y fundamental en la evaluación de cualquier producto desarrollado con IA. En nuestro caso no es un objetivo tan importante porque como he comentado me he querido centrar principalmente en crear un adecuado marco para el desarrollo de un algoritmo de IA.

Ahora sí exponemos nuestro diagrama de flujo de trabajo desarrollado en la siguiente *figura 1*:

Regístrate en PORTAL. Sube tus casos radiológicos y compártelos con el resto de la comunidad

Los casos pueden ser subidos en formato DICOM, JPG, PNG...

[ACCEDER A PORTAL.RADIOLOGYMODELS](#)

1. Creación de una aplicación independiente a Radiologymodels

## Registro

Username

Email

Contraseña

Confirmar

**Registrarse**

¿Ya tienes una cuenta? Inicia sesión

2. Registro de usuarios de forma segura

Formato:

JPEG/JPG

DICOM

Imagen de cabecera:

Seleccionar archivo  Ninguno archivo selec.

Axial:

Elegir archivos  Ninguno archivo selec.

Descripción axial:

3. Opción de subir entrada a partir de un formulario

**RadiologyModels**

JPEG/JPG

DICOM

Si tu estudio versa sobre patología hepática, ¿aceptas que usemos tus datos para la creación de un proyecto piloto de un algoritmo de IA de Imagen Médica? Al aceptar participar tu estudio se cargará en una base de datos que será utilizado en un futuro para el desarrollo de un algoritmo de inteligencia artificial sobre lesiones hepáticas. Te mantendremos informado con los resultados.

SI

Etiquetado de patología según CIE-11

.....

Región de interés patológica

.....

4. ¿Imágenes en formato DICOM?  
¿Versan sobre patología hepática?  
¿Permiso de uso?

```
if accepted_data_use == 'SI':
    output_path_dicom_image = 'media/images/user_{0}/dicom_data.save_as(output_path_dicom_image)
    img.save(output_path_image, format='JPEG')
    output_path_image = output_path_image.replace('media/output_path_image', output_path_image)
else:
    if image_set is axial_images:
        FilePost.objects.create(axial_images=output_path_image)
    elif image_set is sagittal_images:
        FilePost.objects.create(sagittal_images=output_path_image)
    else:
        FilePost.objects.create(coronal_images=output_path_image)
```

Axa (definición anatómica/etiológica)

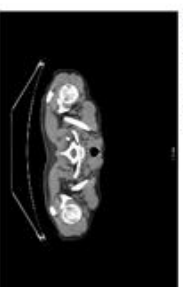
## Contenido

Definición: Área geográfica de hipocapación que a menudo precede a la formación de un verdadero absceso pulmonar. ~7% de las neumonías bacterianas. Gran mortalidad.

## Signos

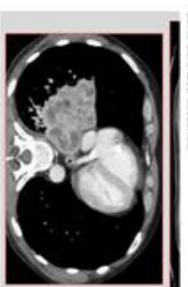
Heterogeneidad del realce del parénquima pulmonar afecto donde las áreas de baja atenuación pueden corresponder a parénquima pulmonar licuofacto + /- afectación típica de consolidación. Se aprecian múltiples abscesos pequeños con focos sin realce en una consolidación pulmonar

Corta Axial/Transversal



Descripción del caso

Corta Coronal/Sagital



7. Guardado de la misma

8. Publicación de la información médica con caso asociado

9. Utilización de la información para desarrollo de algoritmo

No tenemos que perder también de vista cuales son los recursos utilizados en este proyecto: datos de pacientes. Para ello, lo primordial es siempre tener en cuenta la privacidad y anonimización de los datos.

Esto resulta de vital importancia ya que el Reglamento General de Protección es muy claro al respecto. Para ello he considerado varias medidas que permitan que se puedan subir datos médicos sin que exista riesgo de cometer errores.

En primer lugar, en el momento de subir una entrada se lanza un *popover* que avisa al usuario de que la página Radiologymodels no se hace responsable en caso de que se suban datos médicos que hagan identificativas las imágenes, además en cuyo caso la página se hace cargo de la eliminación de dicha entrada y de dicho usuario del portal de registro.

Además, en el caso de que se suban imágenes en formato DICOM, y se acepte su uso como parte de la elaboración del algoritmo, se procederá a la eliminación del *metadato* asociado correspondiente a información médica del paciente. Todo ello como paso previo en el procesamiento, y antes de ser utilizado para el cálculo computacional. Esto permite la anonimización de los datos como exige el Reglamento general de Protección de datos.

Por otra parte se han implementado numerosas medidas de seguridad, que expondrán en los próximos capítulos, y que son necesarias en cualquier servicio web actual y más en aquellos sitios en los que exista registro de usuarios, junto con información personal, como correos electrónicos. Esto pretende hacer más invulnerable nuestro servicio al ciberrobo o divulgación accidental.

Los datos almacenados, como se ha comentado anteriormente, van a consistir en campos de información de texto e imágenes en diferentes formatos. Esto va a requerir un almacenamiento en base de datos, que van a provenir de una misma fuente, que en nuestro caso son los formularios necesarios para realizar una publicación. He querido almacenarlos inicialmente sin procesar, en lo referente a las imágenes. Esto permite ahorrar pasos que son innecesarios en un primer momento, para después utilizando las librerías adecuadas de python, se puedan procesar sin ningún inconveniente. En este aspecto nos encontraríamos ante una especie de *data lake* donde los datos se almacenan en bruto, en un lugar centralizado, para que después debido a la correcta asociación con

las etiquetas correspondientes podemos acceder a ellos para darles el uso deseado. En este aspecto la programación orientada a objetos, con los modelos que se desarrollan en Django se ajusta perfectamente a estos requerimientos.

## Capítulo 3: Marco previo del proyecto

Quisiera introducir este apartado exponiendo que mi trabajo ha sido desarrollo con el *framework Django*, que utiliza como lenguaje de programación *backend, python*. Esto no se trata de casualidad ya que *python* es uno de los lenguajes de programación más utilizados en el desarrollo de IA, como hemos podido ver a lo largo de nuestro curso. Además de ello, cuenta con numerosas librerías para poder analizar imágenes médicas, y la iniciación en su uso es relativamente sencilla para personas ajenas al mundo de la programación informática, como es mi caso.

Quisiera de antemano exponer que en el presente capítulo y en los consecutivos no voy a hacer hincapié en el apartado *frontend*, que ha sido llevado a cabo principalmente mediante el uso de *HTML, CSS o JavaScript*. Esto es así, no porque sea poco importante a la hora de cumplir con nuestros objetivos, sino porque se trata de una sección que no se ha desarrollado en el Máster y que adicionalmente su utilidad final en la formación de un algoritmo de IA es reducida. Por otra parte, por las características propias de mi proyecto si ha desempeñado un rol clave, sobre todo a la hora de ser un factor importante en la consumición de tiempo, y por ser una puerta de entrada para los usuarios, comportándose en el caso de que no estuviera bien llevada a cabo una barrera de entrada para el uso de la herramienta y dificultando en gran medida lo que se pretende; un uso amplio y sencillo por parte de la comunidad de radiólogos de mi hospital.

Previo a la realización de este proyecto, en la página web existente, se podían visualizar estudios subidos personalmente por mí, tanto en formato video como en JPG/PNG. El primer paso entonces, era permitir el registro de usuarios, de tal forma que muchas más personas puedan participar en esta herramienta. Esto lo vamos a ver en el siguiente capítulo.

## Capítulo 4: Registro de usuarios y seguridad

Como he comentado anteriormente el proyecto se ha querido desarrollar utilizando lo que en el mundo de la informática se conoce como *buenas prácticas*.

En el apartado de registro de usuarios lo que concierne al tema de la seguridad ha sido tenido muy en cuenta, ya que nos encontramos manejando correos electrónicos personales.

Para ello, en el proceso de registro, validación de usuario, recuperación de contraseñas, he utilizado el módulo “**auth**”, junto con sus respectivas funciones, por ejemplo “**auth\_login, auth\_logout**”..., y con las plantillas HTML correspondientes, ejemplo: *'users/login.html*. **Este modelo ha proporcionado una gran sencillez en esta sección del proyecto.**

Por otra parte como describo anteriormente y siendo coherente, he utilizado como modelo de usuario el que proporciona el módulo “**auth**”; **UserProfile**, con los campos personalizados en el *anexo 1*.

Especial mención a la encriptación de contraseñas, realizada igualmente con la función “**make\_password**” de “**hashers**”, del mismo módulo anterior.

También hay que tener en cuenta que se ha utilizado la validación por correo electrónico a la hora de registrarse, por tanto la forma que he considerado más segura ha sido mediante el uso de encriptación de datos en secuencias en cadenas base64 (*anexo 2*).

En el subapartado de inicio de sesión considero que ha sido importante introducir las habituales posibilidades que nos encontramos en las diferentes páginas webs de Internet. Algún ejemplo de ello son los casos en los que el usuario no se encuentra validado (en nuestro caso he utilizado validación por correo electrónico), o el hecho de si la contraseña del correo no es válida. En estos casos se remite nuevamente al **users/login.html** (*anexo 3*).

El resto de los subapartados utilizados en el registro y activación de usuarios, como por ejemplo, el cierre de sesión y el reseteo de contraseña no van a ser desarrollados y especificados en este apartado porque siguen una dinámica similar a los expuestos anteriormente.

Por otra parte, en cuanto a la edición del perfil por parte del usuario se ofrecen una serie de opciones entre las que se encuentran definir el país al que se pertenece, yo sobre

todo el año académico y la especialidad radiológica que se posee. Esto, aunque puede ser una medida un poco reducida, nos ofrece un panorama general sobre el tipo de usuario que se registra, atendiendo sobre todo al conocimiento sobre imagen médica que posee, de forma que nos permita aproximarnos a la fiabilidad de la entrada subida.

## Capítulo 5: Edición de una entrada

Una vez que el usuario se ha registrado y validado correctamente, ya puede editar su perfil y crear entradas para la página. Esta es la parte más importante de nuestro trabajo y la que ha requerido más tiempo. Debemos de crear un flujo de datos desde el usuario hasta nuestros servidores.

Como expongo, la creación de entradas es la piedra angular de nuestro proyecto y debe contemplar muchas opciones posibles. Como este trabajo queda englobado dentro de un proyecto ya existente he dado la oportunidad de que se puedan subir entradas sin necesidad de asociar imágenes médicas, de tal forma que nos encontraríamos ante una publicación de información médica sin poder acercarnos a un caso en concreto. Por nuestro interés este debe ser el menor número de casos pero quería dar esta oportunidad porque mucha información médica de interés en el ámbito radiológico no necesita de que haya imágenes asociadas. Un ejemplo de ellos sería una lista de protocolos de estudios por órganos en resonancia magnética (RM) o información sobre los contrastes yodados en pruebas de tomografía computarizada (TC).

El mayor interés y trabajo asociado que ha habido en programación se plantea en aquellos casos donde los usuarios quieran subir a la página web información médica ejemplificada con un caso real con imágenes médicas. En estos casos, adicionalmente se despliega una casilla que permita dar la opción al usuario de que dicha entrada junto con las imágenes asociadas pasen a formar parte de nuestro algoritmo de IA.

He querido especificar que el uso de las imágenes médicas en el algoritmo de IA se lleve a cabo únicamente para aquellos casos en los que la región anatómica de interés sea el hígado. Esto es así porque he considerado que es más fácil a la hora de desarrollar un algoritmo el hecho de que nos centremos en una única región anatómica, tanto para facilitar el proceso de segmentación como también para el correcto etiquetado de la información.

Hablando un poco más en concreto sobre el etiquetado de la información médica en patologías hepáticas he querido utilizar un marco internacional de información médica en clasificación de patologías médicas (Willeminck et al., 2020). De esta forma, además de los datos que el usuario proporcione sobre el caso, es decir el informe radiológico pertinente, proporciono un marco único de categorización de tal forma que podamos tener clasificados y a nuestra disposición, todos los estudios sobre patología hepática que los usuarios estén conformes de introducir en nuestro algoritmo. He querido que sea la patología hepática aquella que verse nuestro algoritmo debido a varios factores: La frecuencia de la patología hepática es muy elevado, su diagnóstico por imagen médica se encuentra en una fase de sensibilidad y rentabilidad alta, existe poca variación en cuanto a género en sus diferentes afectaciones y es un órgano que a priori puede resultar sencillo de segmentar.

Ejemplificando todo lo descrito anteriormente:

1. Usuario sube una entrada de cirrosis hepática
2. Completa o no diferentes campos disponibles: Contenido, fisiopatología, clínica de la enfermedad, bibliografía....
3. Sube imágenes asociadas o no.
  - 3.1 En el caso de que suba imágenes en formato DICOM, tiene oportunidad de asociar una descripción asociada a dichas imágenes. También se le pregunta si quiere que dichas imágenes formen parte de nuestro algoritmo en caso de traten sobre patología hepática.
  - 3.2 En caso afirmativo se despliegan una serie de opciones para categorizar adicionalmente su trabajo según la clasificación CIE-11 (Clasificación Internacional de Enfermedades- 11)

## **¿Por qué utilizar la clasificación CIE- 11?**

Aproximadamente el 80% del esfuerzo de un modelo analítico consiste en preprocesar, combinar, personalizar y limpiar los conjuntos de datos. El disponer de una herramienta previamente que facilite estova a permitir que podamos escalar en un futuro cercano nuestro proyecto, en caso de que fuera necesario, de una forma relativamente fácil.



A la hora de utilizar un gran volumen de datos y de información para desarrollar un modelo de IA, los datos deben ser fácilmente interpretables y sobre todo homogéneos. Para ello lo más recomendable y lo que va a permitir un ahorro de tiempo a posteriori es utilizar desde el principio un protocolos de estandarización y homogeneización de datos, de carácter a poder ser internacional, que permita en un momento dado la ayuda de otros profesionales que se quieran asignar al proyecto y que dispongan de un marco de información real y universales de tal forma que el entrenamiento mediante algoritmos de machine learning (ML) se puedan aplicar independientemente del país de origen de los datos.

CIE- 11 se trata de una de al herramientas más universales y estandarizadas si queremos clasificar enfermedades. En concreto, su última versión ha sido la 11, que fue desarrollada en 2018 y es ampliamente utilizada en los diferentes sistemas médicos diagnósticos del mundo, a nivel epidemiológico y en ámbito de la investigación.

Como algunos de los campos empleados en el ámbito de la patología hepática no tienen mucha relevancia en el ámbito de la imagen médica, como por ejemplo las enfermedades hepáticas metabólicas, he querido añadir un campo que no se encuentra de por sí en la CIE-11. Este campo es “*Otras*” en donde el usuario podrá introducir manualmente una cadena de texto de hasta 100 caracteres, pero que se pretende que sea una descripción lo más escueta posible. Un ejemplo del uso de esta opción sería: Cambios hepáticos postembolización, ya que esta descripción como tal no se encuentra en la CIE-11.

### **¿Por qué utilizar solamente imágenes DICOM en nuestro estudio?**

Esto se expondrá de forma exhaustiva en un apartado específico.

### **¿Por qué utilizar imágenes de TC y no de RM?**

No tenemos que dejar de presente que el actual estudio ha sido desarrollado sobre otro ya existente. Nos encontramos que la página web <https://radiologymodels.com/> se encuentra alojada en una máquina virtual con sistema operativo Linux en un servidor de Microsoft Azure en Estados Unidos. Este proyecto es personal y no dispone de financiación extraordinaria, por tanto nos encontramos con una máquina que cumple su

función pero no tiene todas las características como para desde un primer momento disponer de una subida masiva de datos a la nube y a nuestra de base de datos.

Dicho esto he decidido implementar un algoritmo que utilice solamente imágenes de DICOM de TC. Sobre todo porque para implementar un algoritmo de IA basado en RM consumiría más recursos, que por el momento este proyecto no es capaz de mantener.

En un futuro, si la iniciativa coge recorrido, la escalabilidad es relativamente sencilla tanto a nivel del formulario necesario para recabar más información, por ejemplo de RM, como para pasar nuestro proyecto entero a una máquina virtual más potente.

Igualmente no se ha planteado este trabajo de forma que los cálculos computacionales desarrollados por el algoritmo de IA se lleven a cabo en la nube, no obstante dicha configuración puede ser igualmente sencilla de cambiar, para utilizar de paso las opciones que Microsoft Azure propone. Además dispone de recursos bastante prácticos como los que se dan en Azure ML con algoritmos ya preentrenados. Además su uso por mi parte sería más sencillo por el hecho de tener experiencia en su interfaz.

## Capítulo 6: Visualización de la entrada

En este capítulo quería repasar las dificultades experimentadas para hacer que la visualización de la publicación se produzca de forma agradable, que se ajuste lo más parecido posible a como visualizan los estudios los radiólogos y que incite a usar la plataforma y registrarse, para así incrementar el número de estudios disponibles.

Los radiólogos tienen que examinar en un estudio estándar de TC aproximadamente 300 imágenes. Esto es de forma general, y tirando a la baja, porque en muchas ocasiones, y cada vez más, se generan muchas más imágenes asociadas, con especificaciones concretas según lo que se quiera examinar.

Por otra parte, a modo de ejemplo, un estudio de un paciente politraumatizado puede contar con hasta más de 1000 imágenes. En una guardia de urgencias el médico radiólogo tiene que realizar informes para una media de 20 exploraciones. Con esto quiero decir que el volumen de imágenes a estudiar es altísimo.

He querido exponer estos datos porque quiero resaltar que resulta imposible fijarnos de forma estática en cada una de las imágenes durante un tiempo determinado. Además de no existir el tiempo físico para ello, la utilidad y la rentabilidad para el diagnóstico, salvo para casos concretos es escasa, ya que es necesario primeramente identificar las estructuras de forma rápida y luego centrarnos en la región de interés según el caso que se nos presente.

Con respecto a este tema es muy habitual que los radiólogos empleen el "*hacer scroll*" a la hora de enfrentarse a un estudio. Esto consiste en desplazarnos hacia arriba y hacia abajo para localizar rápidamente las estructuras anatómicas, examinar la dependencia que existe entre unas y otras y poder centrarnos en el problema que tiene el paciente. No se puede concebir el trabajo actual de estos profesionales sin esta actividad tan cotidiana que permiten realizar los diferentes software hospitalarios.

A pesar de ser una herramienta indispensable son sorprendentemente escasos los recursos de divulgación de imagen diagnóstica que permiten esta función; ya sea, por ejemplo, las redes sociales, revistas específicas en imagen médica o sociedades especializadas. Como he expuesto anteriormente, el conocimiento disponible a nivel digital es sin lugar a dudas el de mayor utilidad en la formación de estos profesionales. El hecho de no disponer herramientas en la red que simulen verdaderamente la forma de trabajar de estos profesionales reduce ampliamente la utilidad de este conocimiento.

Debido a esto he querido introducir en la página web lo que se conoce en la jerga informática como un *scroll-container* que consiste en un *box-element* que tiene en su interior "x" número de imágenes de tal forma que haciendo uso de *JavaScript* y *CSS* se pueda ajustar para apilar las imágenes una debajo de otra, programando además la barra de desplazamiento para que simule perfectamente un visor de imágenes médicas.

De esta forma asociado a la información de texto tenemos una representación fiable del caso. En el anexo 4 expongo la sección del código más interesante a este respecto. Lo más importante llevado a cabo aquí es apilar las imágenes verticalmente una debajo de otra. Para ello creamos un script que utiliza básicamente el elemento **`document.querySelectorAll('.image-container')`**. Luego debemos de ir pasando sobre cada imagen "escuchando" el desplazamiento vertical que quiera realizar el usuario con **`scrollTop`** y **`e.deltaY`** y teniendo en cuenta el índice de diapositiva donde nos

encontremos gracias a **currentSlideIndex** y el número de imágenes que tenemos **scrollContainer.querySelectorAll('img').length**. A su vez hacemos un zoom que nos va a permitir centrarnos en la imagen en sí reduciendo la zona en negro del fondo, para ello calculamos las alturas de las imágenes introducidas (que en principio al pertenecer todas al mismo estudio van disponer de la misma altura) con **imageHeight**.

Ajustamos el desplazamiento para que sea de la forma más fluida y sin interrupciones con la siguiente línea de comandos **targetSlide.scrollToView ({block: 'nearest', inline: 'start', behavior: 'instant'})**.

La mayor dificultad en todo este proceso ha sido sobre todo la familiarización con el lenguaje **JavaScript**. El resto de código de la visualización de la entrada pública es relativamente sencillo porque extraemos la información de los modelos establecidos (si existiese). Así, por ejemplo, gran parte de los atributos de texto asociados al modelo Post es el siguiente:

*“Class Post (models.Model):*

*title = models.CharField (max\_length=100)*

*content = models.TextField ()*

*date\_posted = models.DateTimeField(auto\_now\_add=True)*

*author = models.ForeignKey (User, on\_delete=models.CASCADE)*

*category = models.CharField(max\_length=20, choices=POST\_CATEGORIES\_CHOICE, default='Abdomen')*

*fisiopathology = models.TextField(blank=True, null=True)*

*clinical\_case = models.TextField (blank=True, null=True)*  
*class Post(models.Model):*

*title = models.CharField(max\_length=100)*

*content = models.TextField() .....”*

Posteriormente en nuestra *template HTML* mediante bucles *If* comprobamos si existe información asociada y la presentamos asociada de forma responsiva.

## Capítulo 7: Imágenes DICOM

El formato de imágenes DICOM es el sistema más utilizado para el almacenaje de imágenes médicas asociadas a metadatos.

Su incorporación es casi tan antigua como el inicio de la investigación de la IA en el campo de las imágenes médicas comenzó en la década de 1960, cuando los investigadores exploraron el potencial del uso de algoritmos informáticos para ayudar a analizar las imágenes médicas (Kahn et al., 2011).

Es un formato de imagen que presenta varias características adicionales sobre otros que lo convierten en específico para las imágenes médicas. En primer lugar es un formato seguro que permite el envío de datos protegiendo la privacidad. Permite asociar datos fundamentales clínicos del paciente. A su vez, también es una modalidad que soporta imágenes multidimensionales. Esto resulta esencial para poder investigar con pruebas como la TC o la RM ya que fácilmente vamos a poder reproducir los tres planos de estudio que se utilizan en radiología; axial, coronal y sagital. Esto junto con el scroll es esencial para llegar al diagnóstico adecuado debido a que se pueden localizar mucho mejor las estructuras anatómicas.

También, como hemos visto en uno de los módulos este curso, existe un protocolo DICOM de nivel superior binario (ULP, Upper Level Protocol) sobre TCP / IP, que permite buscar múltiples tipos de información.

Asimismo, existen varias librerías en python como *Pydicom* que permiten el procesamiento adecuado de las imágenes en formato DICOM y que permiten numerosas funciones, entre las cuales se encuentran la anonimización de los datos médicos de los estudios procesados. Por otra parte *Numpy* permite trabajar con los *arrays* de píxeles.

Debido a lo expuesto anteriormente, y otras funciones que comentaremos más adelante, he decidido guardar en la base de datos los estudios que se desean incluir en el estudio de IA en formato DICOM. Posteriormente el acceso a ellos junto con el etiquetado

correspondiente es relativamente sencillo, solamente comprobando si dicha entrada presenta en el campo de pregunta si el valor asociado es “Si”:

Formato:

JPEG/JPG

DICOM

Si tu estudio versa sobre patología hepática, ¿aceptas que usemos tus datos para la creación de un proyecto piloto de un algoritmo de IA de Imagen Médica? Al aceptar participar tu estudio se cargará en una base de datos que será utilizado en un futuro para el desarrollo de un algoritmo de inteligencia artificial sobre lesiones hepáticas. Te mantendremos informado con los resultados

Si

Etiquetado de patología según CIE-11

-----

Figura 2

A pesar de todo lo comentado anteriormente se me presentó un problema rápidamente y es que las imágenes DICOM no se pueden visualizar correctamente en los navegadores web. Para ello, he tenido que comprobar si el usuario sube las imágenes en formato DICOM. En caso afirmativo deben ser convertidas previamente a formato JPG. De esta forma, a la hora de la visualización, llamamos a las imágenes guardadas en este formato para mostrarlas en el *scroll-container*. A continuación muestro unas líneas de código de interés de este problema en cuestión. Se trata de una de las funciones más extensas del trabajo porque tenía que tener en cuenta bastantes variables. También en el anexo (6) se encuentra una imagen que expone parte del código de esta sección.

“if *image\_format* == 'DICOM':

*try*:

```
image_header_dicom = pydicom.dcmread(header_image)
```

```
image_header_array = image_header_dicom.pixel_array.astype(float)
```

```
new_image_header = Image.fromarray(image_header_array)
```

```
new_image_header = new_image_header.convert('L')
```

```
output_filename_header = header_image.name.split('.')[0] + '.jpg'
```

```

        output_path_image_header =
'media/images/user_{0}/post_{1}/{2}'.format(post_instance.author.username,
post_instance.pk, output_filename_header)

        os.makedirs(os.path.dirname(output_path_image_header),
exist_ok=True)

        new_image_header.save(output_path_image_header, format='JPEG')

        output_path_image_header =
output_path_image_header.replace('media/', '')

        if not [x for x in (axial_images, sagittal_images, coronal_images) if x]:

            FilePost.objects.create(post=post_instance, format=image_format,
header_image=output_path_image_header, accepted_data_use=accepted_data_use,
cie_11_tagging=cie_11_tagging, interested_region=interested_region,
other_interested_region=other_interested_region)

        for image_set in [axial_images, sagittal_images, coronal_images]:

            for image in image_set:

                dicom_data = pydicom.dcmread(image)

                image_array = dicom_data.pixel_array.astype(float)

                img = Image.fromarray(image_array)

                img = img.convert('L')

                output_filename = image.name.split('.')[0] + '.jpg'

                output_path_image =
'media/images/user_{0}/post_{1}/{2}'.format(post_instance.author.username,
post_instance.pk, output_filename)

                os.makedirs(os.path.dirname(output_path_image), exist_ok=True)

                if accepted_data_use == 'Si':

```

```

        output_path_dicom_image =
'media/images/user_{0}/post_{1}/{2}'.format(post_instance.author.username,
post_instance.pk, image.name)

        dicom_data.save_as(output_path_dicom_image)

img.save(output_path_image, format='JPEG')

output_path_image = output_path_image.replace('media/', '')

if image_set is axial_images:

    FilePost.objects.create(axial_images=output_path_image,
post=post_instance,                                format=image_format,
header_image=output_path_image_header,            axial_description=axial_description,
accepted_data_use=accepted_data_use,              cie_11_tagging=cie_11_tagging,
interested_region=interested_region,
other_interested_region=other_interested_region)

elif image_set is sagittal_images:

    FilePost.objects.create(sagittal_images=output_path_image,
post=post_instance,                                format=image_format,
header_image=output_path_image_header,            sagittal_description=sagittal_description,
accepted_data_use=accepted_data_use,              cie_11_tagging=cie_11_tagging,
interested_region=interested_region,
other_interested_region=other_interested_region)

else:

    FilePost.objects.create(coronal_images=output_path_image,
post=post_instance,                                format=image_format,
header_image=output_path_image_header,            coronal_description=coronal_description,
accepted_data_use=accepted_data_use,              cie_11_tagging=cie_11_tagging,
interested_region=interested_region,
other_interested_region=other_interested_region)

except:

```



*error = True*”

## Capítulo 8: Creación de un algoritmo

Como he dicho en secciones anteriores la intención del trabajo no era desarrollar un algoritmo de IA en sí, sino profundizar en todos los pasos previos a su desarrollo, que hemos ido conociendo en nuestro curso, creando un marco adecuado de extracción de datos de calidad y con información etiquetada asociada, para poder eso sí, utilizarlos en un futuro cercano en la creación de algoritmo de IA.

En mi opinión, son unos objetivos igualmente valiosos, ya que creo que son las principales barreras que existen cuando personal muy cualificado en el sector de la programación quiere desarrollar estrategias basadas en IA en este sector.

No obstante aquí presento unas líneas de código de lo que pudiera corresponder a las primeras etapas de extracción de datos almacenados y procesamiento.

En un futuro cercano, una vez creado este entorno, procederé a realizar más profundamente y a investigar al respecto en los aspectos necesarios para desarrollar un algoritmo de IA útil que pueda ser considerado como activo terapéutico de la clase IIb.

Antes de ponernos con el algoritmo debemos de acceder correctamente a los datos que van a ser utilizados. En nuestro caso dichos datos deben de cumplir una serie de características: 1. El usuario ha dado el permiso de que sean utilizados para generar un algoritmo de IA 2. Seleccionamos una patología en concreto dentro del espectro de patología de la CIE-11. Una vez cumplido con estos requisitos nos habremos asegurado que vamos a trabajar con las imágenes médicas correspondientes. Este sería el código empleado para ello:

```
from users.models import *
```

```
def algoritmo():
```

```
file_posts_permiso= FilePost.objects.filter(USE_DATA_CHOICES='Si')
```

```
file_post_cirrosis=file_posts_permiso.filter( CIE_11_TAGGING_CHOICES = 'DB93
Fibrosis o cirrosis hepática')
```

Una vez que tenemos todos los estudios acumulados en nuestro servidor en donde hay hígados con aspecto, por imagen, de cirrosis procedemos a iterar sobre el conjunto y transformarlos en DICOM. Para ello debemos utilizar nuevamente las librerías Pydicom y PIL de python. Además añadimos un ID a cada uno de los DICOM creados. Este ID va a ser el ID del file\_post\_cirrosis:

```
for archivo in file_posts_cirrosis:
```

```
    axial_images = file_post.axial_images.all()
```

```
    for image in axial_images:
```

```
        if image.name.lower().endswith('.jpg'):
```

```
            image_data = image.read()
```

```
            image_pil = Image.open(BytesIO(image_data))
```

```
            ds = pydicom.Dataset()
```

```
            ds.PatientName = f"Paciente_{file_post.id}"
```

```
            pixel_array = image_pil.convert('L') # Convertir a escala de grises
```

```
            ds.PixelData = pixel_array.tobytes()
```

```
            ds.Rows, ds.Columns = pixel_array.size
```

```
            dicom_buffer = BytesIO()
```

```
            ds.save_as(dicom_buffer)
```

```
            dicom_content = ContentFile(dicom_buffer.getvalue())
```

```
            dicom_filename = f"dicom_{file_post.id}_{image.name}.dcm"
```

```
            file_post.dicom_images.create(image_dicom=dicom_content
```

Una vez que tenemos las imágenes de cada objeto, que versa sobre cirrosis hepática, de nuestro modelo, transformadas en formato DICOM podemos trabajar con ellas.

De aquí en adelante ya se puede proceder a intentar desarrollar un algoritmo de IA que permita identificar la cirrosis hepática.

Posteriormente y según lo aplicado en nuestro curso deberíamos introducir también estudios normales/o con otras patologías para el futuro entrenamiento de la IA. Más adelante deberíamos comprobar que el rendimiento de la IA es adecuado.

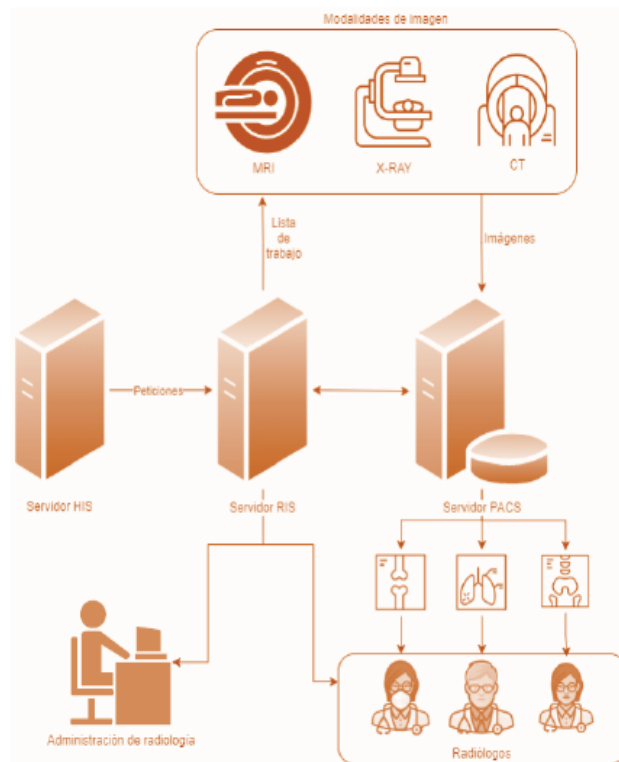
## Capítulo 9: Perspectiva de futuro

Como se ha podido apreciar de lo comentado en los apartados anteriores, el usuario es quien tiene que introducir manualmente las imágenes de un caso que le haya generado interés y también molestarse en subir información médica asociada.

Esto puede resultar interesante en un entorno de enseñanza, como puede ser el conjunto de residentes que componen un servicio de Radiodiagnóstico, pero supongo que puede resultar tedioso para personas ajenas a este proyecto y que no se encuentran motivadas con él.

Debido a ello considero varias opciones que pueden resultar mucho más productivas y eficientes en un futuro cercano. Aquí quería introducir otro concepto común a las unidades de Radiodiagnóstico y que también hemos podido estudiar en el Máster. Se trata de los PACS (Picture Archiving and Communication System) de imágenes médicas. El PACS es el sistema que se encarga de gestionar todas las imágenes médicas generadas en los diferentes centros de salud junto con la comunicación correcta que permita identificar a cada estudio con su correspondiente paseante.

Por ejemplo, en el caso del funcionamiento de la radiología en el SAS (Servicio Andaluz de Salud), el PACS seleccionado fue proporcionado por CARESTREAM, empresa mundial canadiense. Su instauración allá por el año 2005 supuso la creación de uno de los PACS más grande de Europa al dar servicio a una población de aproximadamente 8,5 millones de personas.



**Ilustración 1: Arquitectura de un sistema integrado RIS/PACS, donde el servidor RIS administra la gestión del departamento mientras que el servidor PACS actúa como servidor de imágenes**

**Figura 3**

Las posibilidades de este servicio en IA son enormes, al disponer de una cantidad de ingente de datos perfectamente guardados y con información detallado asociada al respecto. Por si fuera poco, además de esto, el GSIR (sistema de gestión de información radiológica) permite rescatar cualquier tipo de estudio almacenado en el PACS introduciendo una serie de palabras clave. De esta forma podemos buscar todos los estudios, de TC, RM.... en el hospital donde el usuario esté trabajando, haciendo referencia a un periodo de tiempo en concreto, en los que en el informe radiológico asociado posea por ejemplo las palabras clave.: “cirrosis hepática”.

Para cierto tipo de patologías que son bastantes frecuentes entre la población el número de casos que puede aportar un único hospital de tercer nivel pueden llegar a ser suficientes para desarrollar algoritmos de IA. Por ejemplo, términos como “cirrosis hepática” o incluso mejor “esteatosis hepática” ofrecen una cantidad de estudios suficientes como para generar un algoritmo con rendimiento adecuado. Para ello, tal vez solo debemos acceder a los informes radiológicos en los que aparezcan dichas palabras claves, con sus correspondientes estudios asociados localizados en el PACS durante un periodo de 5 años.

Con esto quiero decir que se podría desarrollar otro proyecto diferente a este pero siguiendo un esquema parecido para aprovechar al máximo dichas funcionalidades del sistema actual de imágenes médicas, en donde los radiólogos desde remoto pudieran participar en un proyecto de investigación de forma cómoda sin tener que desplazarse como sucede en muchos sitios hasta estaciones de trabajo dedicadas a ello y realizando un trabajo más manual de lo deseado.

Serían necesarios algunos cambios. Entre ellos el más importante sería el poder establecer una aplicación que permitiese tanto la conexión y el enlace a GSIR como a CARESTREAM. En el presente trabajo he utilizado, como expongo en el siguiente capítulo, un sistema de API (Application Programming Interface) que permite la comunicación entre el proyecto Radiologymodels y el actual trabajo *Portal*. De forma más o menos parecida y teniendo en cuenta las medidas adicionales de seguridad que deben de tener estos protocolos de interacción entre diferentes software se podría extraer las imágenes en una base de datos externa dedicada a la investigación según la descripción introducida en el sistema de búsqueda del GSIR y mediante previa comprobación del estudio por parte del radiólogo para asegurarse de la fiabilidad de la descripción dada en el informe radiológico previo. Esta API debería poder, como conocemos, acceder mediante su IP y AET a los archivos DICOM.

Esto es un ejemplo de las posibilidades que ofrecen los servicios basados en API. He podido buscar información al respecto y comprobar que el SAS tiene ya creado una plataforma que define los requisitos necesarios para desarrollar una aplicación externa que permita dar solución a un problema sanitario ya existente, teniendo en cuenta aspectos sobre la seguridad de las API junto con la documentación necesaria para acceder a la información médica. Para ello, es necesario presentar un plan de trabajo,

que resulte interesante, y como digo, presente algún servicio o ayuda tanto a los profesionales médicos como a los pacientes. Más información al respecto se encuentra en la página web: <https://ws001.sspa.juntadeandalucia.es/confluence/display/SERVCGESP/Us+de+la+API>

## Capítulo 10: Escalabilidad del proyecto

Cuando plantee el proyecto (*Portal*) para hacerlo de la forma más seria posible, la futura escalabilidad del mismo fue un tema que consideré importante, no porque pensara que fuera a ser utilizado rápidamente si no que resultaba que era la manera en la que iba a aprender más haciéndolo, lo más realista posible y siguiendo las buenas prácticas. Para ello lo cree como servicio independiente de la página web, en una máquina virtual aparte y estableciendo una comunicación entre ambos trabajos mediante un servicio de APIS. A continuación, muestro parte del código: Utilizo como es lógico Django REST Framework para devolver una vista de API en formato JSON. Primeramente tuve que descargar los decoradores que me iban a hacer falta. En mi caso solo se iban a hacer solicitudes mediante el método “**GET**” y mediante autenticación previa del usuario, para ello empleamos, por ejemplo, **api\_view**, o **authentication\_classes**. Posteriormente también me di cuenta que era necesario aplicar Tokens de autenticación (módulo **rest\_framework.authentication**). Una vez que obtenemos los datos serializados, creamos un diccionario, en donde lo más importante es que va a tener el nombre de la **vista** junto con el “**pk**” correspondiente. Esto nos llevará expresamente a la publicación que queremos visualizar (más información en el **anexo correspondiente (5)**).

De esta forma junto con la información que proporciono en el anexo, creamos un enlace entre ambos proyectos de una forma igualmente segura mediante autenticación basada en Tokens.

Esto lo realicé de esta forma porque las necesidades de ambos sitios son muy diferentes. En el caso de que fuera necesario ampliar las características de la máquina virtual para incrementar la memoria RAM o cambiar a una con un disco duro de mayor tamaño esta forma de trabajar ofrece una mayor sencillez.

Debido a que en un futuro cercano quiero añadir las opciones de poder subir mayor número de imágenes (ejemplo actual: **coronal\_description = models.CharField (max\_length=500, blank=True, null=True)**) y además quiero que haya más campos disponibles para poder simular un estudio completo de RM (con todos los campos de imagen/secuencias necesarias que ello implica), puede llegar a ser necesario como digo implementar una mejora en la máquina virtual, cosa que Microsoft Azure permite muy fácilmente.

## Agradecimientos

Para terminar quería resaltar el hecho de lo que ha significado este Máster y trabajo para mí. Durante este curso he podido aprender de una forma muy transversal los diferentes aspectos que hay que tener en cuenta para desarrollar estrategias de IA. Además he intentado compaginar dichos estudios con el interés por la programación que tengo para aunar dos mundos a primera vista muy distantes como pueden ser la medicina y la informática.

Quería agradecer igualmente al profesorado, sobre todo por las explicaciones sencillas que se dan en las clases virtuales y por las secciones donde se profundizaba en el lenguaje programación de python, que para mí, ha resultado la parte que más provechosa me ha resultado. Esto es debido a que también es la sección que más me interesaba previamente a inscribirme en este curso.



## Bibliografía

- Hsu, W., & Hoyt, A. C. (2019). Using Time as a Measure of Impact for AI Systems: Implications in Breast Screening. *Radiology*, *1*(4), e190107. <https://doi.org/10.1148/ryai.2019190107>.
- Balsera, A. (2022, 22 abril). El Imibic impulsa en Córdoba la inteligencia artificial en el diagnóstico y cribado del cáncer de mama. El Día de Córdoba. [https://www.eldiadicordoba.es/cordoba/Imibic-Cordoba-diagnostico-cancer-mama\\_0\\_1676832646.html](https://www.eldiadicordoba.es/cordoba/Imibic-Cordoba-diagnostico-cancer-mama_0_1676832646.html)
- Fariña, E., & Fariña, E. (2022). SerAM alerta de la falta de radiólogos en España. El médico interactivo. <https://elmedicointeractivo.com/seram-alerta-falta-de-radiologos/>
- Maleck, M., Fischer, M. R., Kammer, B., Zeiler, C., Mangel, E., Schenk, F., & Pfeifer, K. (2001). Do Computers Teach Better? A media comparison study for case-based teaching in radiology. *Radiographics*, *21*(4), 1025-1032. <https://doi.org/10.1148/radiographics.21.4.g01jl091025>
- Willemink, M. J., Koszek, W., Hardell, C., Wu, J., Fleischmann, D., Harvey, H., Folio, L. R., Zhang, D., Rubin, D. L., & Lungren, M. P. (2020). Preparing medical imaging data for machine learning. *Radiology*, *295*(1), 4-15. <https://doi.org/10.1148/radiol.2020192224>
- Kahn, C. E., Langlotz, C. P., Channin, D. S., & Rubin, D. L. (2011). Informatics in Radiology: An information model of the DICOM Standard. *RadioGraphics*, *31*(1), 295-304. <https://doi.org/10.1148/rg.311105085>.
- En España se realizan más de 40 millones de pruebas de imagen al año, según la Sociedad Española de Radiología Médica (SERAM) - SERAM", SERAM, 8 de octubre de 2022, <https://seram.es/en-espana-se-realizan-mas-de-40-millones-de-pruebas-de-imagen-al-ano-segun-la-sociedad-espanola-de-radiologia-medica-seram/>.

# Anexos

## Anexo 1:

```
class UserProfile(models.Model):
    academic_year_list = ['Estudiante', 'R0', 'R1', 'R2', 'R3', 'R4', 'R5', 'Adjunto']
    medical_specialty_list = ['Radiología pediátrica', 'Radiología ginecológica', 'Radiología de abdomen', 'Neurorradiología', 'Radiología intervencionista', 'Radiología musculoesquelética', 'Radiología general']

    user = models.OneToOneField(User, on_delete=models.CASCADE)
    country = models.CharField(max_length=100, choices=[(country.name, country.name) for country in pycountry.countries], blank=True, null=True)
    academic_year = models.CharField(max_length=100, choices=[(academic_year, academic_year) for academic_year in academic_year_list], blank=True, null=True)
    work_location = models.CharField(max_length=100, blank=True, null=True)
    medical_specialty = models.CharField(max_length=100, choices=[(medical_specialty, medical_specialty) for medical_specialty in medical_specialty_list], blank=True, null=True)
```

## Anexo 2:

```
subject = render_to_string('users/activation_email_subject.txt')
message = render_to_string('users/activation_email_message.txt', {
    'user': user,
    'domain': current_site.domain,
    'uid': urlsafe_base64_encode(force_bytes(user.pk)),
    'token': account_activation_token.make_token(user),
})
```

## Anexo 3:

```
if user and user.check_password(form.cleaned_data['password']):
    if user.is_active:
        auth_login(request, user)
        return redirect(reverse('users:user_dashboard'))
    else:
        return render(request, 'users/login.html', {'form': form, 'error_message': 'Tu cuenta no está activa. Revisa tu correo electrónico para activarla.})
else:
    return render(request, 'users/login.html', {'form': form, 'error_message': 'Correo o contraseña incorrectos.'})
```

## Anexo 4:

```
<script>
  document.querySelectorAll('.image-container').forEach(function(scrollContainer) {
    scrollContainer.addEventListener('scroll', function(e) {
      e.preventDefault();

      const scrollTop = scrollContainer.scrollTop;
      const containerHeight = scrollContainer.clientHeight;
      const imageHeight = scrollContainer.querySelector('img').clientHeight;
      const diapositiva = (containerHeight / imageHeight);
      const imageCount = scrollContainer.querySelectorAll('img').length;
      const currentSlideIndex = Math.round(scrollPosition / imageHeight);

      let scrollTargetIndex = currentSlideIndex;

      const sensitivityFactor = 2;

      if (e.deltaY > 0) {
        scrollTargetIndex = scrollTargetIndex + sensitivityFactor;
      } else if (e.deltaY < 0) {
        scrollTargetIndex = scrollTargetIndex - sensitivityFactor;
      }

      const targetSlide = scrollContainer.querySelectorAll('img')[scrollTargetIndex];
      targetSlide.scrollIntoView({ block: 'nearest', inline: 'start', behavior: 'instant' });

      const targetScrollPosition = scrollTargetIndex * imageHeight;
      scrollContainer.scrollTop = targetScrollPosition;
    });
  });
</script>
```

## Anexo 5:

```
@api_view(['GET'])
@authentication_classes([TokenAuthentication])
@permission_classes([IsAuthenticated])
def post_urls(request):
    posts = Post.objects.all()
    post_data = []

    for post in posts:
        user = post.author
        user_data = UserSerializer(user).data
        post_dict = {
            'titulo': post.title,
            'autor': user_data,
            'categoria': post.category,
            'url': request.build_absolute_uri(reverse('blog:blog_page', args=[post.title, post.pk]))
        }
        post_data.append(post_dict)

    return Response(post_data)
```

## Anexo 6:

```
if image_format == 'DICOM':
    try:
        image_header_dicom = pydicom.dcmread(header_image)
        image_header_array = image_header_dicom.pixel_array.astype(float)
        new_image_header = Image.fromarray(image_header_array)
        new_image_header = new_image_header.convert('L')
        output_filename_header = header_image.name.split('.')[0] + '.jpg'
        output_path_image_header = 'media/images/user_{0}/post_{1}/{2}'.format(post_instance.author.username, post_instance.pk, output_filename_header)
        os.makedirs(os.path.dirname(output_path_image_header), exist_ok=True)
        new_image_header.save(output_path_image_header, format='JPEG')
        output_path_image_header = output_path_image_header.replace('media/', '')

    if not [x for x in (axial_images, sagittal_images, coronal_images) if x]:
        FilePost.objects.create(post=post_instance, format=image_format, header_image=output_path_image_header, accepted_data_use=accepted_data_use, cie_11_tagging=cie_11_tagging, interested_region=interested_region)

    for image_set in [axial_images, sagittal_images, coronal_images]:
        for image in image_set:
            dicom_data = pydicom.dcmread(image)
            image_array = dicom_data.pixel_array.astype(float)
            img = Image.fromarray(image_array)
            img = img.convert('L')
            output_filename = image.name.split('.')[0] + '.jpg'
            output_path_image = 'media/images/user_{0}/post_{1}/{2}'.format(post_instance.author.username, post_instance.pk, output_filename)
            os.makedirs(os.path.dirname(output_path_image), exist_ok=True)
            if accepted_data_use == 'S1':
                output_path_dicom_image = 'media/images/user_{0}/post_{1}/{2}'.format(post_instance.author.username, post_instance.pk, image.name)
                dicom_data.save_as(output_path_dicom_image)
                img.save(output_path_image, format='JPEG')
                output_path_image = output_path_image.replace('media/', '')

            if image_set is axial_images:
                FilePost.objects.create(axial_images=output_path_image, post=post_instance, format=image_format, header_image=output_path_image_header, axial_description=axial_description, accepted_data_use=accepted_data_use)
            elif image_set is sagittal_images:
                FilePost.objects.create(sagittal_images=output_path_image, post=post_instance, format=image_format, header_image=output_path_image_header, sagittal_description=sagittal_description, accepted_data_use=accepted_data_use)
            else:
                FilePost.objects.create(coronal_images=output_path_image, post=post_instance, format=image_format, header_image=output_path_image_header, coronal_description=coronal_description, accepted_data_use=accepted_data_use)
        except:
            error = True
```