



VNiVERSIDAD
D SALAMANCA

TRABAJO FIN DE GRADO DE INGENIERÍA INFORMÁTICA

ANEXO V.

DOCUMENTACIÓN DEL CÓDIGO FUENTE

Autor:

Rubén Fraile Sanchón

Tutores:

Andrea Vázquez Ingelmo

Alicia García Holgado

Francisco José García Peñalvo

Enero 2023

TABLA DE CONTENIDO

1.	INTRODUCCIÓN	4
2.	LENGUAJES DE PROGRAMACIÓN.....	5
2.1.	PYTHON	5
2.2.	HTML, CSS Y JAVASCRIPT	5
2.3.	BOOTSTRAP	5
2.4.	FONTAWESOME	6
2.5.	PYCHARM	6
2.6.	POSTGRESQL	6
3.	ESTRUCTURA DEL SISTEMA	8
4.	DOCUMENTACIÓN DEL CÓDIGO FUENTE	10
	BIBLIOGRAFÍA	13

ÍNDICE DE FIGURAS

<i>Figura 1. Estructura del proyecto en pycharm.....</i>	<i>8</i>
<i>Figura 2. Plantillas HTML del proyecto.....</i>	<i>9</i>
<i>Figura 3. Código CSS y JavaScript del proyecto.....</i>	<i>9</i>
<i>Figura 4. Administración de django del proyecto</i>	<i>10</i>
<i>Figura 5. Enlace a la documentación del código del proyecto</i>	<i>10</i>
<i>Figura 6. Documentación del proyecto.....</i>	<i>11</i>
<i>Figura 7. Modelos disponibles en la aplicación</i>	<i>11</i>
<i>Figura 8. Modelo de ejemplo de la gestión de imágenes dicom.....</i>	<i>12</i>

1. INTRODUCCIÓN

En este anexo se expondrán los diferentes lenguajes de código de programación que se han utilizado a lo largo de la realización del proyecto, la estructura del código y la especificación de la documentación con el objetivo de facilitar las labores de mantenimiento del código fuente en futuras líneas de trabajo.

2. LENGUAJES DE PROGRAMACIÓN

2.1. PYTHON

El lenguaje de programación utilizado para el desarrollo de la aplicación ha sido Python debido a que la aplicación base existente ya se encontraba desarrollada en dicho lenguaje. Este lenguaje dispone de numerosas librerías de ciencias de datos y matemáticas muy útiles a la hora de tratar imágenes de cualquier tipo, en nuestro caso, imágenes DICOM.

Este lenguaje se complementa con el *framework* web Django [1] debido a que permite un gran desarrollo rápido de sitios web seguros y con gran mantenibilidad.

2.2. HTML, CSS Y JAVASCRIPT

Para el desarrollo *front-end* del sitio web se han utilizado los lenguajes de programación HTML, CSS y JavaScript.

El lenguaje de programación HTML se ha usado para la definición de la estructura de la página, complementado con la definición de estilos que ofrece el lenguaje de programación CSS, separando de esta forma el contenido de la presentación, lo que permite obtener códigos mucho más limpios y sencillos.

Para añadir a la página atributos interactivos, como pueden ser la aparición y desaparición de iconos los cambios en el aspecto del menú, la creación de nodos de actividades o la aparición de los resultados de haber aplicado algoritmos a las imágenes se ha utilizado el lenguaje de programación JavaScript. JavaScript se trata de un lenguaje de programación que puede aplicarse a un documento HTML para ofrecerle el dinamismo que se ha mencionado anteriormente.

2.3. BOOTSTRAP

Para lograr el desarrollo de un diseño web adaptativo (*responsive design*) se ha utilizado la herramienta *Bootstrap*, que permite realizar este tipo de diseños a través

de componentes reutilizables y estructuras predefinidas como son las filas, las columnas o las tablas, además de ser una herramienta muy flexible pues permite la personalización de todas las estructuras y componentes [2].

2.4. FONTAWESONE

Se utiliza la biblioteca *FontAwesome* para la inserción y diseño de todos los iconos de la aplicación web.

Posee una gran cantidad de iconos disponibles, todos ellos personalizables mediante el lenguaje de programación CSS. La selección se hace de manera muy sencilla e intuitiva y su inserción se hace con tan solo una línea de código. Al tratarse de una biblioteca de código abierto, se tiene libertad de uso sobre cualquiera de sus diseños.

2.5. PYCHARM

En cuanto al entorno de desarrollo se utiliza *PyCharm*, IDE especializado en el lenguaje de programación *Python*. Se escoge este entorno debido a sus grandes características de finalización de código inteligente, inspecciones de código, errores automáticos con diversas soluciones sugeridas y funcionalidades de navegación muy completas.

En particular, se ha utilizado la versión profesional para estudiantes.

2.6. POSTGRESQL

Esta herramienta es utilizada como sistema gestor de la base de datos. Se trata de una herramienta de código libre considerada en la actualidad como uno de los más robustos y potentes.

Se escogió esta base de datos estructurada debido a su fácil instalación que permite un sistema multiplataforma. Por otra parte, es capaz de ajustarse al número de CPU

y a la cantidad de memoria disponible por lo que se podrán recoger múltiples peticiones de manera simultánea.

3. ESTRUCTURA DEL SISTEMA

La *Figura 1* muestra la estructura de la aplicación desarrollada en el IDE Pycharm.

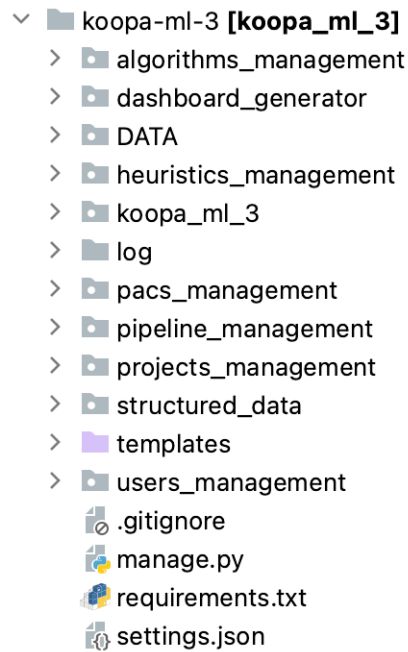


FIGURA 1. ESTRUCTURA DEL PROYECTO EN PYCHARM

En ella se detallan los siguientes aspectos:

- Las carpetas representadas con un punto son aquellas que se conocen como aplicaciones en Django. Cada una de ellas representa un apartado diferente de la aplicación entre los que se encuentran los que ya estaban en el proyecto y los añadidos por el desarrollador actual:
 - Gestión de Algoritmos (añadido por el desarrollador)
 - Generador del Tablero
 - Gestión de Heurísticas
 - Gestión de Entidades PACS (añadido por el desarrollador)
 - Gestión de Pipelines
 - Gestión de Proyectos DICOM (añadido por el desarrollador)
 - Gestión de Imágenes DICOM (añadido por el desarrollador)
 - Gestión de Usuarios

- La aplicación principal será la carpeta *koopa_ml_3* con el mismo icono que las carpetas anteriores.
- La carpeta *templates* representa todas las plantillas HTML que han sido utilizadas a lo largo del proyecto y que se encuentran divididas en secciones de la aplicación como se puede observar en la *Figura 2*.

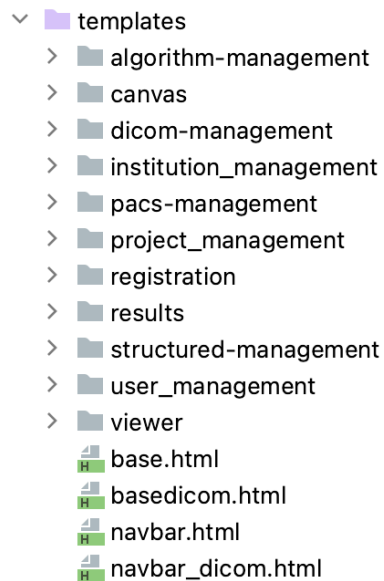


FIGURA 2. PLANTILLAS HTML DEL PROYECTO

- El código relacionado con los lenguajes CSS y JavaScript se encuentra dentro de la carpeta de la aplicación principal. Este código se puede observar en la *Figura 3*.

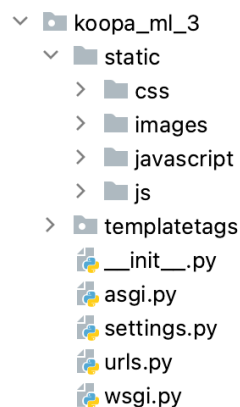


FIGURA 3. CÓDIGO CSS Y JAVASCRIPT DEL PROYECTO

4. DOCUMENTACIÓN DEL CÓDIGO FUENTE

Para la documentación del código fuente se ha seguido los patrones indicados en la documentación del *framework* Django [1]. De esta forma se ha conseguido generar una documentación de forma sencilla y automática y, es posible acceder a ella a través del administrador de Django.

Si se quiere observar la documentación del código fuente, únicamente habrá que acceder al administrador a través del siguiente enlace:

<https://koopaml.grial.eu/admin/>

Como podemos observar en la *Figura 4*



FIGURA 4. ADMINISTRACIÓN DE DJANGO DEL PROYECTO

Para revisar la documentación se ha de acceder al enlace que se encuentra en la esquina superior derecha cuyo nombre es "Documentación" y que se encuentra señalado en la *Figura 5*



FIGURA 5. ENLACE A LA DOCUMENTACIÓN DEL CÓDIGO DEL PROYECTO

Como segunda opción, dicha documentación se podrá encontrar en `/admin/doc`, como se puede observar en la *Figura 6*, si pulsamos sobre cada uno de los diferentes apartados se devolverá una descripción, métodos o parámetros de cada uno de ellos.

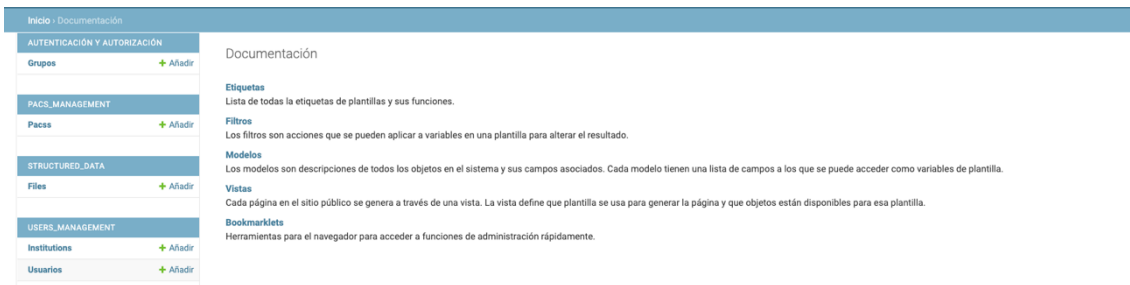


FIGURA 6. DOCUMENTACIÓN DEL PROYECTO

Destacamos el contenido del apartado de modelos de la Figura, este apartado se corresponde con descripciones de todos los objetos del sistema y sus campos asociados. Cada uno de los modelos tiene una lista de campos a los que se puede acceder como variables de plantilla. En la *Figura 7* se pueden observar los modelos que se encuentran disponibles en la aplicación

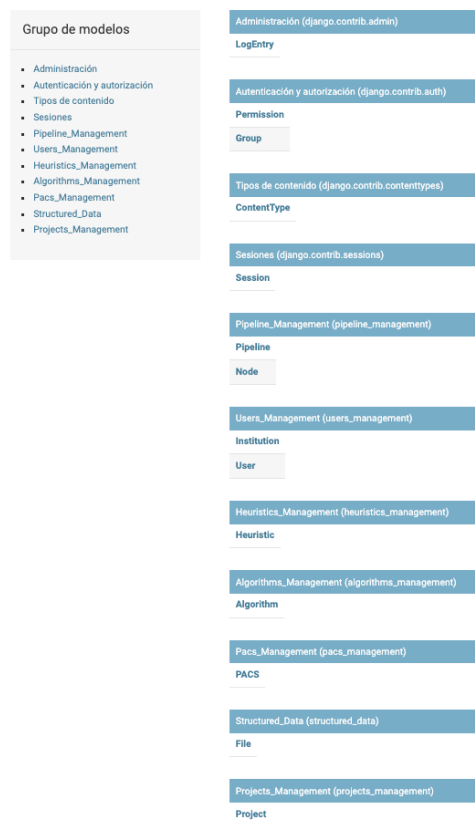


FIGURA 7. MODELOS DISPONIBLES EN LA APLICACIÓN

A modo de ejemplo, en la *Figura 8* se mostrará el contenido del modelo File, correspondiente con las imágenes DICOM de la aplicación.

structured_data.File

File(id, project, name, file, name_patient, name_study, type, position_x, position_y, position_z, sort_index, uploaded_by)

Campos

CAMPO	TIPO	DESCRIPCIÓN
file	Archivo	file
id	Entero grande (8 bytes)	ID
name	Cadena (máximo 200)	name
name_patient	Cadena (máximo 200)	name patient
name_study	Cadena (máximo 200)	name study
position_x	Número en coma flotante	position x
position_y	Número en coma flotante	position y
position_z	Número en coma flotante	position z
project	Project	el objeto relacionado projects_management.Project
sort_index	Número en coma flotante	sort index
type	Cadena (máximo 200)	type
uploaded_by	User	el objeto relacionado users_management.User

FIGURA 8. MODELO DE EJEMPLO DE LA GESTIÓN DE IMÁGENES DICOM

Bibliografía

- [1] Django, «The Django Admin Documentation Generator,» [En línea]. Available: <https://docs.djangoproject.com/en/3.2/ref/contrib/admin/admindocs/>.
- [2] T. Bacinger, «What is Bootstrap? A Short Bootstrap Tutorial on the What, Why, and How. Retrieved from Developers,» [En línea]. Available: <https://www.toptal.com/front-end/what-is-bootstrap-a-short-tutorial-on-the-what-why-and-how>. [Último acceso: 1 Diciembre 2022].