



VNiVERSIDAD  
D SALAMANCA

TRABAJO FIN DE GRADO DE INGENIERÍA INFORMÁTICA

ANEXO III.

ANÁLISIS DEL SISTEMA SOFTWARE

**Autor:**

Rubén Fraile Sanchón

**Tutores:**

Andrea Vázquez Ingelmo

Alicia García Holgado

Francisco José García Peñalvo

Enero 2023

## TABLA DE CONTENIDO

---

<b>1.</b>	<b>INTRODUCCIÓN .....</b>	<b>4</b>
<b>2.</b>	<b>MODELO DE DOMINIO .....</b>	<b>5</b>
<b>3.</b>	<b>PAQUETES DE ANÁLISIS Y SERVICIO.....</b>	<b>7</b>
3.1.	CLASES DE ANÁLISIS.....	8
<b>4.</b>	<b>DESCRIPCIÓN DE LA ARQUITECTURA.....</b>	<b>10</b>
4.1.	CONTEXTO ARQUITECTURA C4.....	10
4.2.	CONTENEDOR ARQUITECTURA C4.....	11
<b>5.</b>	<b>DIAGRAMA ENTIDAD-RELACIÓN .....</b>	<b>12</b>
	<b>BIBLIOGRAFÍA .....</b>	<b>14</b>

## ÍNDICE DE FIGURAS

---

<i>Figura 1. Modelo del dominio del sistema software .....</i>	<i>6</i>
<i>Figura 2. Diagrama de Paquetes de Análisis .....</i>	<i>7</i>
<i>Figura 3. Paquete de análisis Gestión de Algoritmos .....</i>	<i>8</i>
<i>Figura 4. Paquete de análisis Gestión de Imágenes DICOM .....</i>	<i>8</i>
<i>Figura 5. Paquete de análisis Gestión de PACS .....</i>	<i>9</i>
<i>Figura 6. Arquitectura C4 .....</i>	<i>10</i>
<i>Figura 7. Contenedor C4.....</i>	<i>11</i>
<i>Figura 8. Diagrama entidad-relación .....</i>	<i>13</i>

# 1. INTRODUCCIÓN

---

Este anexo se corresponde con la documentación relacionada con el análisis, refinamiento y estructuración realizadas sobre los requisitos de la aplicación. El resultado de este análisis va a permitir una mayor comprensión mucho más precisa de la aplicación que ayudará a diseñar su estructura [1].

La estructura que se corresponde con este anexo es la siguiente:

- **Modelo de dominio:** Representación de las clases conceptuales significativas del dominio del problema.
- **Paquetes de análisis y de servicio:** Descomposición del sistema en partes mucho más manejables. En esta sección también identifica las clases de análisis que forman cada uno de los diferentes paquetes de la aplicación.
- **Descripción de la arquitectura:** Vista de la arquitectura del modelo de análisis.
- **Diagrama Entidad-Relación:** Muestra las diferentes entidades del sistema y, a su vez, cómo se relacionan también.

## 2. MODELO DE DOMINIO

---

El objetivo de la ingeniería del dominio consiste en identificar, construir, catalogar y diseminar un conjunto de componentes de software que tienen aplicación en el software actual y futuro dentro de un dominio de aplicación particular [2]

En concreto, el modelo del dominio es el resultado de las actividades que pertenecen al análisis del dominio [3]. Estas actividades son:

- Definir el dominio a investigar
- Categorizar los elementos extraídos del dominio
- Recoger una muestra representativa de las diferentes aplicaciones del dominio
- Analizar cada aplicación de las muestras recogidas
- Desarrollar un modelo de análisis para los objetos
- Definir un lenguaje del dominio, es decir, hacer posible la especificación y construcción posterior de aplicaciones dentro del dominio.

El modelo del dominio se centra en las abstracciones relevantes y en el vocabulario e información del dominio apoyándose en los requisitos de información descritos en el [Anexo I: Especificación de Requisitos Software](#).

En la *Figura 1* se muestra el Modelo de Dominio correspondiente a la aplicación que se está desarrollando.

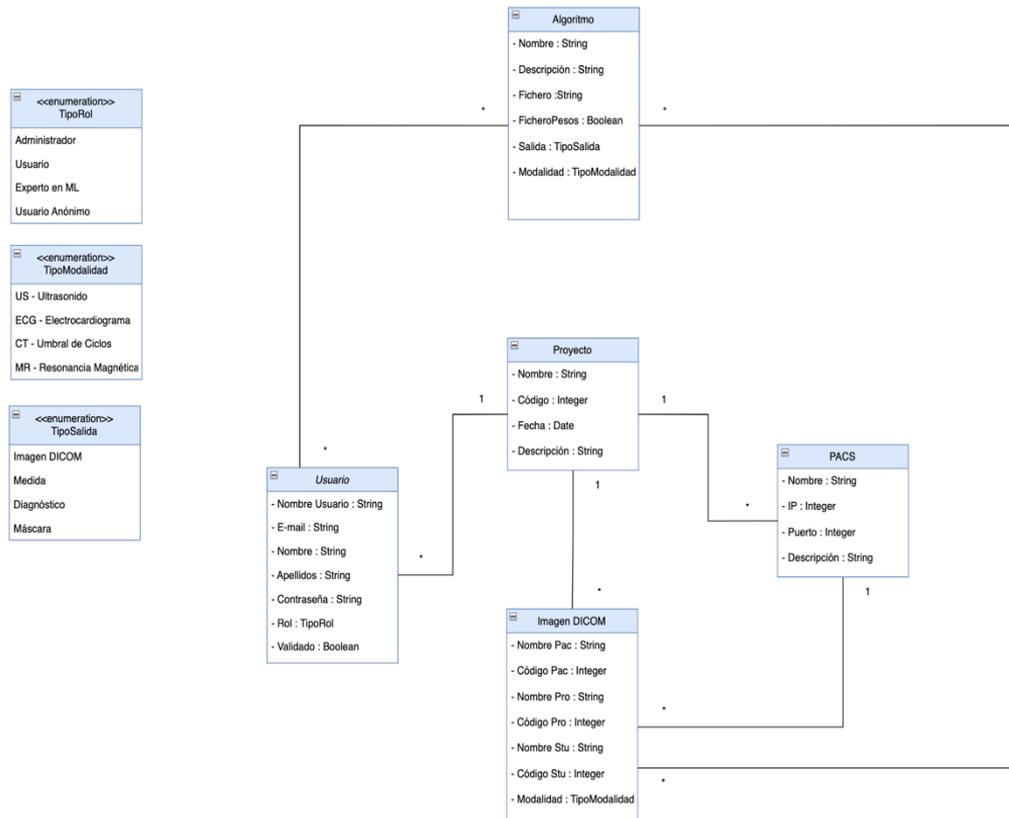


FIGURA 1. MODELO DEL DOMINIO DEL SISTEMA SOFTWARE

A continuación, se explicarán las diferentes clases que se pueden observar en el modelo de dominio mostrado:

- **Usuario**. Representa la información de las personas que tienen acceso a la información del sistema.
- **Proyecto**. Representa los diferentes proyectos del sistema.
- **Imagen DICOM**. Representa las imágenes que forman parte de un proyecto.
- **PACS**. Representa los PACS de los que se pueden extraer las diferentes imágenes.
- **Algoritmo**. Representa los algoritmos que se pueden aplicar a las imágenes.

### 3. PAQUETES DE ANÁLISIS Y SERVICIO

En este punto se realiza la descomposición del sistema en diferentes paquetes de análisis y paquetes de servicio para que se pueda realizar un proyecto mucho más manejable durante el ciclo de vida de éste. En la *Figura 2*, se muestra el diagrama de paquetes del análisis.

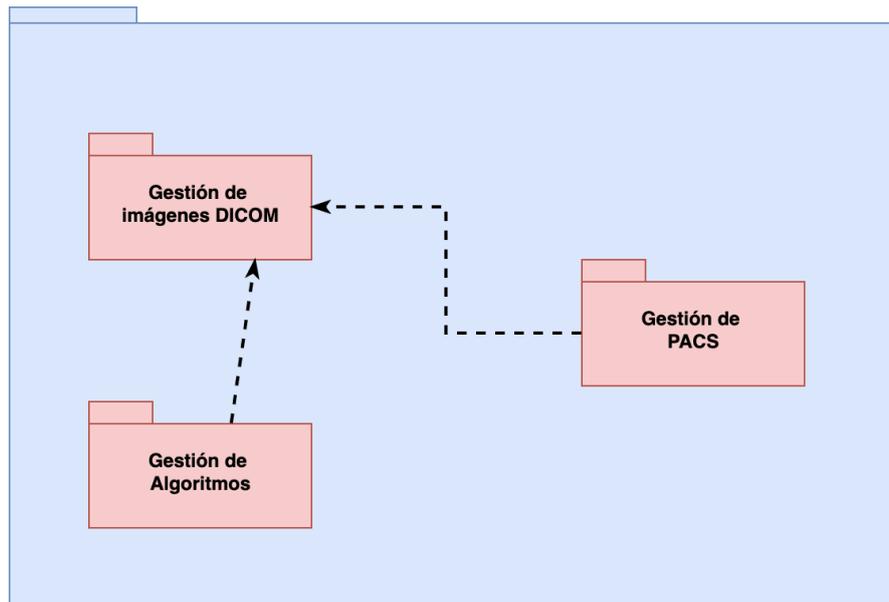


FIGURA 2. DIAGRAMA DE PAQUETES DE ANÁLISIS

A continuación, se pasa a explicar más concretamente cada uno de los paquetes que se pueden observar en el diagrama:

- **Gestión de imágenes DICOM.** Realiza la gestión de la información relacionada con las imágenes DICOM del sistema. Además, realiza tareas para poder aplicar algoritmos a dichas imágenes.
- **Gestión de Algoritmos.** Realiza la gestión de la información relacionada con los algoritmos que hay en el sistema.
- **Gestión de PACS.** Realiza la gestión de la información relacionada con las entidades PACS que hay en el sistema. Además, realiza las tareas de conexión del sistema con los diferentes PACS almacenados.

### 3.1. CLASES DE ANÁLISIS

En este apartado se van a representar una pequeña abstracción de una o varias clases y/o subsistemas del diseño del sistema. Se van a definir los siguientes 3 tipos de clases de análisis:

- **Clases de Entidad.** Se encargarán de modelar la información persistente del sistema, siendo normalmente, relacionada con las clases conceptuales.
- **Clases de Control.** Actúan como unión entre los objetos de la interfaz y de la entidad.
- **Clases de Interfaz.** Engloban toda la funcionalidad que depende directamente del entorno. Se encarga de modelar las interacciones que se producen entre el sistema y los actores. Los eventos de cambios de interfaz que no depende de la información que trata el sistema no se van a comunicar con las clases de control. Esto hace permitir que se pueda llegar a diseñar diferentes vistas sin modificar las clases de control.

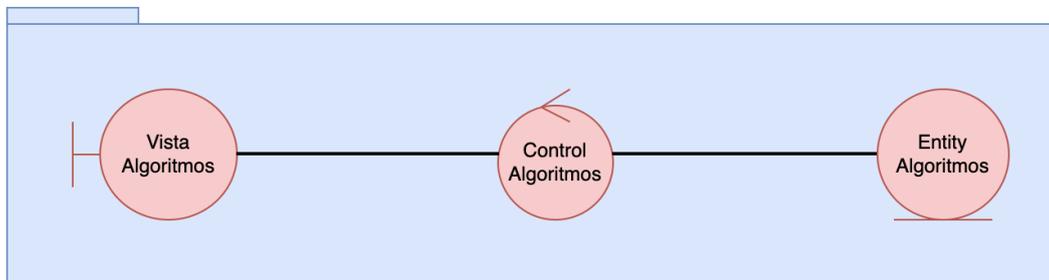


FIGURA 3. PAQUETE DE ANÁLISIS GESTIÓN DE ALGORITMOS

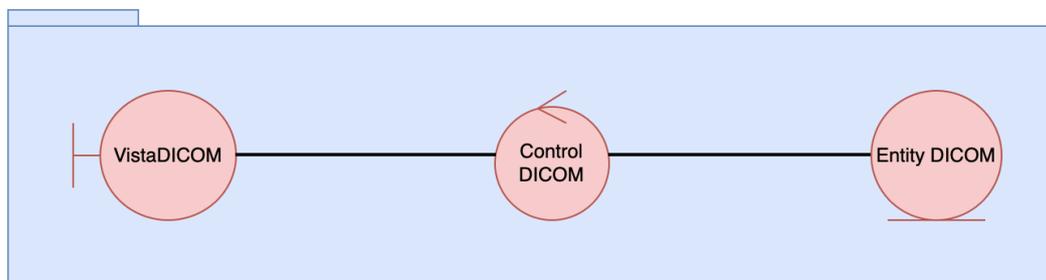
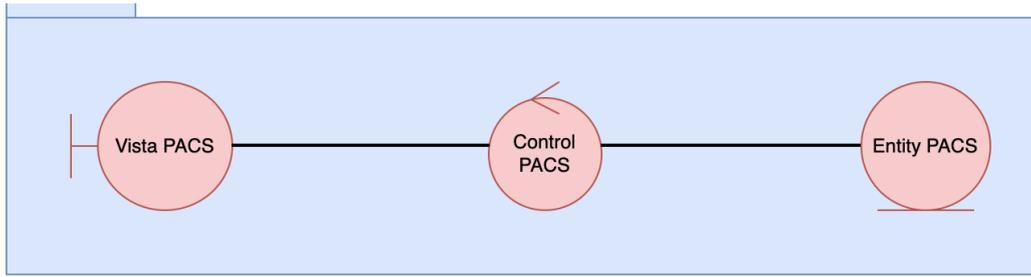


FIGURA 4. PAQUETE DE ANÁLISIS GESTIÓN DE IMÁGENES DICOM



*FIGURA 5. PAQUETE DE ANÁLISIS GESTIÓN DE PACS*

## 4. DESCRIPCIÓN DE LA ARQUITECTURA

El modelo de arquitectura que se ha utilizado es el modelo de arquitectura C4 propuesto por Simon Brown que nos permite diseñar el sistema utilizando cuatro niveles de abstracción, lo que establece un lenguaje común, facilitando la comunicación entre los integrantes de un equipo [4]

Para este sistema sólo se ha necesitado utilizar y establecer los dos primeros niveles.

### 4.1. CONTEXTO ARQUITECTURA C4

En este nivel se representa el sistema objetivo y la interacción de éste con las personas u otros sistemas [4]. Se trata de un modelo pensado para ser comprendido por usuarios sin conocimientos técnicos.

En este caso, el sistema software *KoopaML* va a ser utilizado por usuarios (representados en el diagrama de actores) lo que les va a permitir, por un lado, visualizar y explorar datos, entrenar algoritmos de *Machine Learning* y aprender sobre este proceso y, por otro lado, les permitirá recibir, gestionar y almacenar imágenes DICOM.

La *Figura 6* nos representa el modelo de arquitectura C4 a nivel de contexto.

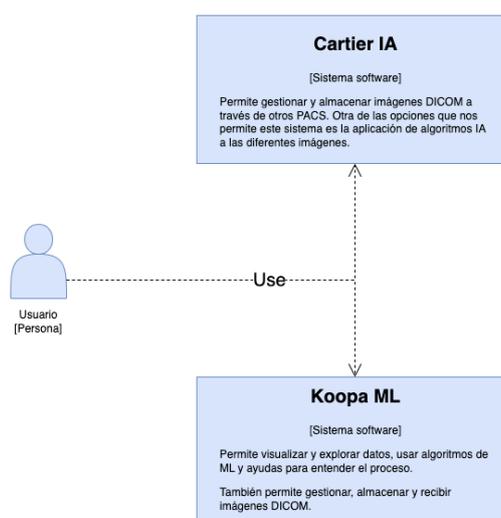


FIGURA 6. ARQUITECTURA C4

## 4.2. CONTENEDOR ARQUITECTURA C4

En este nivel se muestra más detalladamente la información del nivel anterior, haciendo referencia a las aplicaciones, almacenamiento de datos y otros microservicios [4].

El modelo de esta aplicación a este nivel nos muestra cómo el usuario se comunicará con el sistema; el usuario utilizará su navegador (*Front-end*) y será éste el que se comunique con el servidor (*Back-end*) y la API (Interfaz de Programación de Aplicaciones). El servidor va a ser quien se comunique con la base de datos que, en nuestro caso, es una base de datos relacional.

En la *Figura 7* podemos observar el modelo que se acaba de explicar.

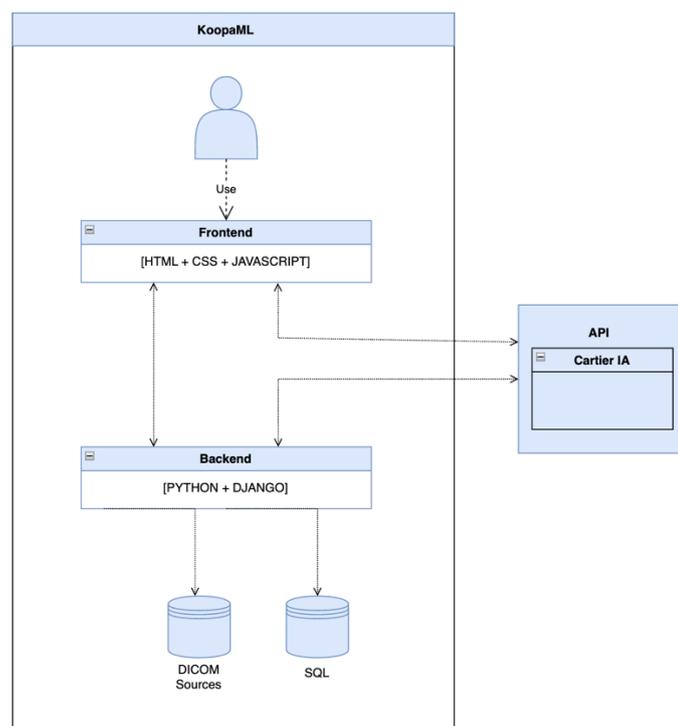


FIGURA 7. CONTENEDOR C4

Siendo, el *front-end* es la parte del software que interactúa con los usuarios y el *back-end* la parte que procesa la entrada desde el *front-end*. La separación del sistema en ambas partes es un tipo de abstracción que ayuda a mantener las diferentes partes del sistema separadas.

## 5. DIAGRAMA ENTIDAD-RELACIÓN

---

Un diagrama entidad-relación nos muestra la forma en la que se relacionan las distintas entidades del sistema.

En este caso, el sistema presenta únicamente cinco entidades, la entidad correspondiente a los usuarios que harán uso de nuestro sistema es Usuario, la que se corresponde con los proyectos DICOM que crea el usuario, la que corresponde a las imágenes DICOM se llama DICOM, la que corresponde con las entidades PACS a las que se puede conectar la aplicación para la extracción de las imágenes se llama PACS y, por último, la entidad que se corresponde con los algoritmos que se van a aplicar a las diferentes imágenes de los proyectos DICOM.

La entidad Usuario almacena atributos como el nombre único de usuario, el email, el nombre, los apellidos, la contraseña, el rol y si el usuario se encuentra o no validado (aunque en este último caso no podrá realizar ninguna de las funciones relacionadas con imágenes DICOM).

La entidad Proyecto almacena atributos como el código, el identificador del usuario que ha creado ese proyecto, un nombre que se le asigna automáticamente al proyecto siendo este Nuevo Proyecto DICOM – Código. A parte, esta entidad almacena la fecha en la que se creó y una descripción que el usuario le quiere dar a la hora de su creación.

La entidad DICOM almacena atributos como nombre e id del paciente, del proyecto y del estudio al que pertenecen, modalidad de la imagen, su propio identificador y la dirección IP del PACS al que pertenece.

La entidad PACS almacena atributos como el título de la entidad, dirección IP, puerto por el que se establece la conexión y una breve descripción.

La entidad Algoritmo almacena atributos como su nombre, la descripción, la fecha en la que se añadió a la plataforma, un fichero que contiene el código de ejecución de dicho algoritmo, si aplica unos ficheros de pesos para que se pueda aplicar de una mejor manera el algoritmo y el tipo de salida (que en este caso siempre serán imágenes) y las modalidades a las que se puede aplicar este algoritmo.

Las relaciones existentes son las siguientes:

- Un Usuario puede ser el autor de 0 o más imágenes DICOM, pero, una imagen solo puede pertenecer a un usuario.
- Un Usuario puede crear 0 o más proyectos DICOM, pero un solo Proyecto puede pertenecer a una persona, es decir, sólo tiene un creador.
- Un Proyecto contiene 0 o más imágenes DICOM y, a su vez, una imagen DICOM puede pertenecer a 1 o más proyectos.
- Un PACS contiene 0 o más imágenes DICOM, pero un solo PACS puede pertenecer a una imagen.
- Un Algoritmo puede ser aplicado a 0 o más imágenes y, a su vez, a una imagen DICOM se le pueden aplicar 0 o más algoritmos.

A continuación, podemos observar lo explicado anteriormente con el diagrama entidad-relación del sistema.

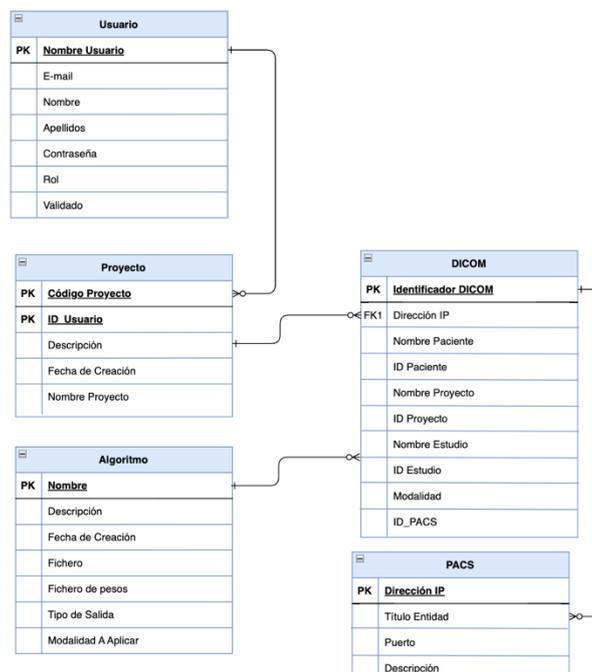


FIGURA 8. DIAGRAMA ENTIDAD-RELACIÓN

## Bibliografía

---

- [1] G. P. F. J., G. H. Alicia y V. I. Alicia, «Transparencias Ingeniería del Software I,» Ingeniería del Software I, 2020.
- [2] R. S. Pressman, Ingeniería del Software. Un Enfoque Práctico. 7ª Edición, Mc Graw Hill, 2010.
- [3] G. P. F. José, G. H. Alicia y V. I. Andrea, «Tema 3. Modelos de Proceso (Ingeniería del Software I),» 2020.
- [4] Mauro, «Modelo de arquitectura C4,» 7 Marzo 2020. [En línea]. Available: <https://mauro.ec/2020/07/04/modelo-de-arquitectura-c4/>.