



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería
Informática**

**título del TFG
Documentación Técnica**



Presentado por Jorge Martínez Martín
en Universidad de Burgos — 20 de mayo
de 2024

Tutor: Álgvar Arnaiz González

Índice general

Índice general	i
Índice de figuras	iii
Índice de tablas	iv
Apéndice A Plan de Proyecto Software	1
A.1. Introducción	1
A.2. Planificación temporal	2
A.3. Estudio de viabilidad	6
Apéndice B Especificación de Requisitos	7
B.1. Introducción	7
B.2. Objetivos generales	7
B.3. Catálogo de requisitos	7
B.4. Especificación de requisitos	9
Apéndice C Especificación de diseño	23
C.1. Introducción	23
C.2. Diseño de datos	23
C.3. Diseño procedimental	29
C.4. Diseño arquitectónico	29
Apéndice D Documentación técnica de programación	31
D.1. Introducción	31
D.2. Estructura de directorios	31
D.3. Manual del programador	31

D.4. Compilación, instalación y ejecución del proyecto	31
D.5. Pruebas del sistema	31
Apéndice E Documentación de usuario	33
E.1. Introducción	33
E.2. Requisitos de usuarios	33
E.3. Instalación	33
E.4. Manual del usuario	33
Apéndice F Anexo de sostenibilización curricular	35
F.1. Introducción	35
Bibliografía	37

Índice de figuras

B.1. Diagrama de casos de uso	9
C.1. Diagrama entidad-relación	24
C.2. Diagrama relacional de la aplicación	26

Índice de tablas

B.1. CU-1 Iniciar sesión.	10
B.2. CU-2 Finalizar sesión.	11
B.3. CU-3 Visualizar evolución.	12
B.4. CU-4.1 Dar de alta a usuarios.	13
B.5. CU-4.2 Dar de baja a usuarios.	14
B.6. CU-4.3 Modificar usuarios.	15
B.7. CU-4.4 Visualizar usuarios.	16
B.8. CU-5.1 Añadir vídeos.	17
B.9. CU-5.2 Eliminar vídeos.	18
B.10. CU-5.3 Modificar vídeos.	19
B.11. CU-5.4 Clasificar vídeos.	20
B.12. CU-6.1 Añadir paciente.	21
B.13. CU-6.2 Eliminar paciente.	22
C.1. Diccionario de datos: Admin	27
C.2. Diccionario de datos: Doctor	27
C.3. Diccionario de datos: Patient	27
C.4. Diccionario de datos: Video	28
C.5. Diccionario de datos: Medicine	28
C.6. Diccionario de datos: PatientMedicine	28

Apéndice A

Plan de Proyecto Software

A.1. Introducción

El propósito de este documento es establecer una guía de forma clara y organizado para la ejecución de un proyecto. En él se definen los objetivos y la forma en la que se van a alcanzar.

Scrum

Para la planificación del proyecto ha sido empleada la metodología ágil Scrum [2]. Scrum es una metodología ágil para la gestión de proyectos complejos en entornos cambiantes. Se basa en tres pilares: eventos, roles y artefactos, y se trabaja en sprints de una duración determinada, generalmente entre una semana y un mes. Los equipos de Scrum se autoorganizan y se comprometen a entregar resultados de alta calidad de manera eficiente y creativa.

Roles

En la metodología Scrum, se destacan tres roles principales:

- *Scrum Master*: Líder del equipo encargado de eliminar obstáculos y facilitar la auto-organización y coordinación del equipo.
- *Product Owner*: Representa la voz del cliente, lidera el desarrollo del producto y busca el valor para los usuarios.

- Equipo de desarrollo: Responsable de ejecutar las acciones previstas para el éxito del proceso

Eventos

Los eventos clave en Scrum son:

- *Sprint*: Reunión para planificar el trabajo del próximo sprint.
- Scrum diario: Breve reunión diaria para sincronizar actividades.
- Revisión del *sprint*: Revisión del producto al final del *sprint*.
- Retrospectiva del *sprint* : Análisis de los éxitos y fallos del *sprint*

Artefactos

Scrum utiliza tres artefactos principales:

- Pila de producto o *product backlog*: Inventario que contiene el trabajo pendiente en el producto.
- Pila del *sprint* o *sprint backlog*: Lista de tareas a realizar durante el *sprint*.
- Incremento: Versión mejorada y funcional del producto al final de cada *sprint*.

A.2. Planificación temporal

Planificación mediante *sprints*

Se ha decidido que la planificación del proyecto se haga mediante *sprints* debido al tamaño del equipo (un desarrollador).

Sprint 1

- **Objetivos**
 1. Configuración inicial: creación del repositorio y configuración de cuenta de ZenHub.
 2. Memoria: redacción de la introducción y familiarización con las aplicaciones para edición de archivos T_EX.

3. Lectura de papers: «Supervised classification of bradykinesia in Parkinson's disease from smartphone videos» [4], «The discerning eye of computer vision: Can it measure Parkinson's finger tap bradykinesia?» [3] y «A computer vision framework for finger-tapping evaluation in Parkinson's disease» [1].
- **Periodo** Este *sprint* se desarrolló entre el 2 de octubre del 2023 y el 16 de octubre del 2023.
 - **Review** En la revisión se resolvieron dudas sobre la documentación y se decidió cambiar de herramienta para la realización de la metodología scrum del proyecto a zube.

Sprint 2

- **Objetivos**
 1. Documentación de trabajos previos: lectura del trabajo realizado por Catalin para ver la extracción de datos.
 2. Memoria: redacción del apartado de trabajos relacionados.
 3. Curso de flask: realización de los tutoriales del framework de flask para la realización del apartado web.
- **Periodo** Este *sprint* se desarrolló entre el 16 de octubre del 2023 y el 29 de octubre del 2023.
- **Review** Se resolvieron dudas sobre dónde encontrar la información dentro del proyecto de Catalin así como problemas a la hora de compilar los documentos de L^AT_EX.

Sprint 3

- **Objetivos**
 1. Documentación de trabajos previos: lectura de los documentos para la extracción de datos usando el paquete de python TSFresh.
 2. Memoria: corrección de errores en la documentación y añadida biografía.
 3. Curso de flask: continuada la realización de los tutoriales del framework de flask para la realización del apartado web.

4. Repositorio: añadido el archivo gitignore para evitar la subida de archivos temporales.
- **Periodo** Este *sprint* se desarrolló entre el 30 de octubre del 2023 y el 12 de noviembre del 2023.
 - **Review** Problemas a la hora de la realización de los objetivos propuestos agregados en el siguiente *sprint*.

Sprint 4

- **Objetivos**
 1. Creación del *mockup* de la aplicación.
 2. Documentación de los anexos.
- **Periodo** Este *sprint* se desarrolló entre el 13 de noviembre del 2023 y el 30 de diciembre del 2023.
- **Review** Se revisó el *mockup* y se propuso una modificación del mismo.

Sprint 5

- **Objetivos**
 1. Finalización del curso de flask.
 2. Modificación del *mockup*.
 3. Creación de diagramas entidad relación y diagramas de casos de uso.
- **Periodo** Este *sprint* se desarrolló entre el 30 de noviembre del 2023 y el 12 de diciembre del 2023.
- **Review** Se revisaron tanto los diagramas como el *mockup* y se propusieron unos cambios en los diagramas.

Sprint 6

- **Objetivos**
 1. Documentación general añadiendo técnicas y herramientas.
 2. Búsqueda de trabajos parecidos o aplicaciones médicas similares.

3. Modificación de los diagramas de casos de uso y entidad-relación.

- **Periodo** Este *sprint* se desarrolló entre el 13 de diciembre del 2023 y el 19 de enero de 2024.
- **Review** Se cumplieron todos los objetivos del sprint.

Sprint 7

- **Objetivos**

1. Documentación general añadiendo técnicas y herramientas.
2. Búsqueda de trabajos parecidos o aplicaciones médicas similares.
3. Modificación del diagramas de casos de uso y entidad-relación.

- **Periodo** Este *sprint* se desarrolló entre el 20 de enero de 2024 y el 15 de febrero de 2024.
- **Review** Se cumplieron todos los objetivos del sprint. Se detectaron ciertos fallos en la documentación que se añadieron como tarea para el siguiente sprint.

Sprint 8

- **Objetivos**

1. Comienzo de la página web.
2. Conceptos teóricos de la aplicación.
3. Solución de errores detectados en la anterior review.
4. Conexión con la base de datos.

- **Periodo** Este *sprint* se desarrolló entre el 16 de febrero del 2024 y el 29 de febrero de 2024.
- **Review** Se cumplieron todos los objetivos del sprint.

Sprint 9

- **Objetivos**

1. Continuación de los conceptos teóricos.

2. Creación de la página de registro.

- **Periodo** Este *sprint* se desarrolló entre el 1 de marzo del 2024 y el 14 de marzo de 2024.
- **Review** No se cumplió el objetivo de la documentación que se añadió al siguiente sprint.

A.3. Estudio de viabilidad

Viabilidad económica

Viabilidad legal

Apéndice B

Especificación de Requisitos

B.1. Introducción

Previo a la creación de un programa se debe especificar las características concretas que ha de tener. En este caso caso ha definido estas características utilizando UML como estándar.

B.2. Objetivos generales

El objetivo principal de esta aplicación es mostrar, tanto a los pacientes como a sus respectivos médicos, su evolución mediante el uso de inteligencia artificial.

Además se busca que el usuario médico sea capaz de realizar comparaciones de múltiples pacientes de manera simultánea.

B.3. Catálogo de requisitos

A continuación se presentan los requisitos funcionales y no funcionales de la aplicación.

Requisitos funcionales

RF1: El sistema debe permitir diferenciar entre tres tipos de roles: administrador, paciente y médico.

RF2: El usuario con rol de paciente ha de ser capaz de agregar vídeos.

- RF3:** El usuario con rol de paciente ha de ser capaz de eliminar sus vídeos.
- RF4:** El usuario con rol de paciente ha de ser capaz de modificar sus vídeos.
- RF5:** El paciente y el médico han de ser capaces de clasificar los vídeos.
- RF6:** Los usuarios han de ser capaces de iniciar sesión con sus correspondientes credenciales.
- RF7:** Los usuarios deben ser capaces de poder finalizar sesión.
- RF8:** El administrador ha de ser capaz de añadir usuarios.
- RF9:** El administrador ha de ser capaz de eliminar usuarios.
- RF10:** El administrador ha de ser capaz de modificar usuarios.
- RF11:** El administrador ha de ser capaz de visualizar a los usuarios.
- RF12:** Los médicos han de ser capaces de agregarse pacientes asociados.
- RF13:** Los médicos han de ser capaces de eliminar sus pacientes asociados.
- RF14:** El paciente y el médico han de ser capaces de ver la evolución del paciente.

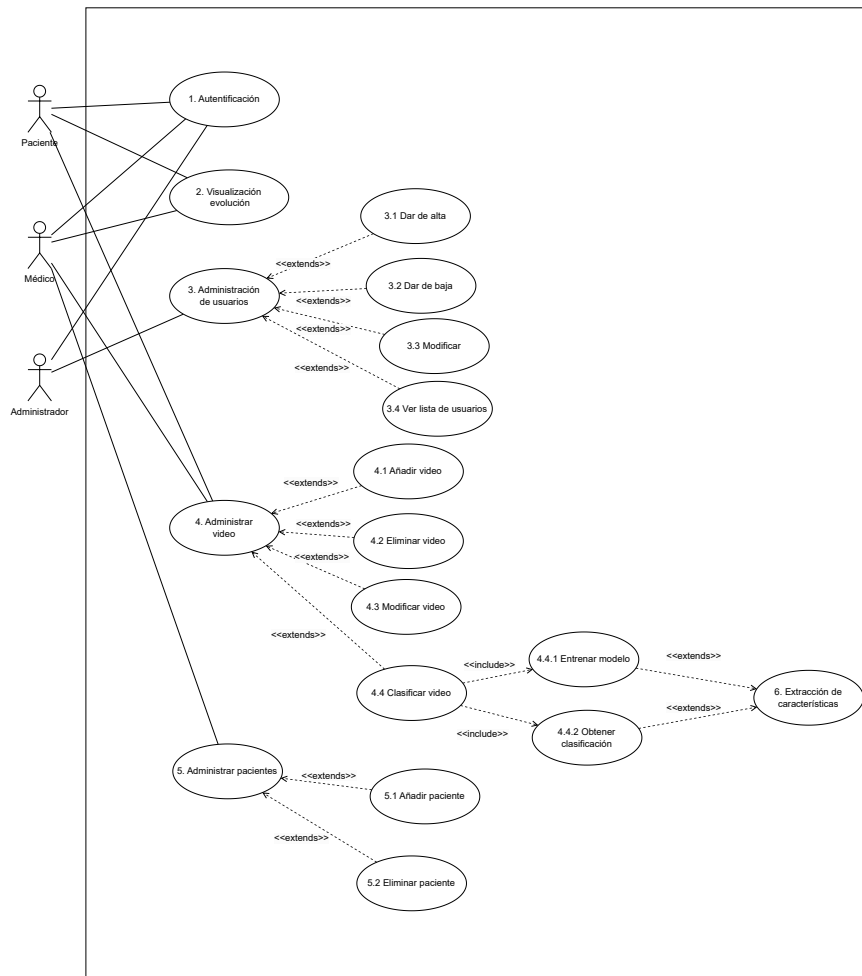


Figura B.1: Diagrama de casos de uso

B.4. Especificación de requisitos

En este apartado se especifican los casos de uso correspondientes a los requisitos funcionales previamente expuestos.

CU-1	Iniciar sesión
Versión	1.0
Autor	Jorge Martínez Martín
Requisitos asociados	RF-1,RF-6
Descripción	Los usuarios deben ser capaces de iniciar sesión
Precondición	<ul style="list-style-type: none"> ■ El usuario debe estar dado de alta en la base de datos. ■ Estar en la página de inicio de sesión.
Acciones	<ul style="list-style-type: none"> ■ El usuario introduce su nombre de usuario. ■ El usuario introduce su contraseña. ■ El usuario hace clic en el botón de iniciar sesión.
Postcondición	El usuario será redirijo a su página de perfil correspondiente
Excepciones	<ul style="list-style-type: none"> ■ Las credenciales no son correctas. ■ El usuario no está dado de alta.
Importancia	Alta

Tabla B.1: CU-1 Iniciar sesión.

CU-2	Finalizar sesión
Versión	1.0
Autor	Jorge Martínez Martín
Requisitos asociados	RF-1,RF-7
Descripción	Los usuarios deben ser capaces de iniciar sesión
Precondición	<ul style="list-style-type: none">■ El usuario debe tener la sesión iniciada.
Acciones	<ul style="list-style-type: none">■ El usuario hace clic en el botón de finalizar sesión.
Postcondición	El usuario será redirijo a su página de perfil correspondiente
Excepciones	
Importancia	Alta

Tabla B.2: CU-2 Finalizar sesión.

CU-3	Visualizar evolución
Versión	1.0
Autor	Jorge Martínez Martín
Requisitos asociados	RF-14
Descripción	El paciente y el médico han de ser capaces de visualizar la evolución del paciente
Precondición	<ul style="list-style-type: none"> ■ El paciente debe estar dado de alta en la base de datos. ■ El paciente debe de tener al menos 2 vídeos para ver la gráfica.
Acciones	<ul style="list-style-type: none"> ■ El usuario accede a su perfil. ■ Se visualiza la gráfica de la evolución del paciente.
Postcondición	Se muestra al usuario su gráfico correspondiente
Excepciones	
Importancia	Alta

Tabla B.3: CU-3 Visualizar evolución.

CU-4.1	Dar de alta a usuarios
Versión	1.0
Autor	Jorge Martínez Martín
Requisitos asociados	RF-8
Descripción	El administrador debe poder dar de alta a nuevos usuarios.
Precondición	<ul style="list-style-type: none"> ■ El usuario no debe estar dado de alta.
Acciones	<ul style="list-style-type: none"> ■ El administrador pulsa el botón de «añadir usuario». ■ El administrador introduce los datos del paciente. ■ El administrador confirma los datos y pulsa el botón «añadir».
Postcondición	Se muestra el nuevo usuario en la lista de usuarios
Excepciones	<ul style="list-style-type: none"> ■ El usuario ya existe. ■ Los datos introducidos son erróneos.
Importancia	Alta

Tabla B.4: CU-4.1 Dar de alta a usuarios.

CU-4.2	Dar de baja a usuarios
Versión	1.0
Autor	Jorge Martínez Martín
Requisitos asociados	RF-9
Descripción	El administrador debe poder dar de baja a usuarios.
Precondición	<ul style="list-style-type: none"> ■ El usuario debe estar dado de alta.
Acciones	<ul style="list-style-type: none"> ■ El administrador pulsa el botón de «eliminar usuario». ■ El administrador confirma el usuario y pulsa el botón «eliminar».
Postcondición	Ya no se muestra al usuario en la lista de usuarios
Excepciones	<ul style="list-style-type: none"> ■ El usuario no existe.
Importancia	Alta

Tabla B.5: CU-4.2 Dar de baja a usuarios.

CU-4.3	Modificar usuarios
Versión	1.0
Autor	Jorge Martínez Martín
Requisitos asociados	RF-10
Descripción	El administrador debe poder modificar usuarios.
Precondición	<ul style="list-style-type: none">■ El usuario debe estar dado de alta.
Acciones	<ul style="list-style-type: none">■ El administrador pulsa el botón de «modificar usuario».■ El administrador realiza los cambios del usuario y pulsa el botón «confirmar».
Postcondición	Se muestra la modificación del usuario.
Excepciones	<ul style="list-style-type: none">■ El usuario modificado coincide con otro que ya existe.
Importancia	Alta

Tabla B.6: CU-4.3 Modificar usuarios.

CU-4.4	Visualizar usuarios
Versión	1.0
Autor	Jorge Martínez Martín
Requisitos asociados	RF-11
Descripción	El administrador debe poder visualizar la vista de usuarios.
Precondición	<ul style="list-style-type: none">■ Debe existir al menos 1 usuario.
Acciones	<ul style="list-style-type: none">■ El administrador visualiza la lista de usuarios.
Postcondición	Se muestra la lista de usuarios.
Excepciones	
Importancia	Media

Tabla B.7: CU-4.4 Visualizar usuarios.

CU-5.1	Añadir vídeos
Versión	1.0
Autor	Jorge Martínez Martín
Requisitos asociados	RF-2
Descripción	Tanto los médicos como los pacientes han de ser capaces de añadir nuevos vídeos.
Precondición	<ul style="list-style-type: none"> ■ Debe existir el paciente al que se le quiere añadir el vídeo.
Acciones	<ul style="list-style-type: none"> ■ El usuario pulsa el botón «añadir vídeo». ■ Se rellenan los datos correspondientes del vídeo como la mano. ■ Se añade el fichero con el vídeo que se quiere subir. ■ Se pulsa el botón «añadir».
Postcondición	El nuevo vídeo debe estar en la lista de vídeos del paciente correspondiente.
Excepciones	<ul style="list-style-type: none"> ■ No se han rellenado los datos correctamente. ■ El fichero subido no corresponde con formato solicitado.
Importancia	Alta

Tabla B.8: CU-5.1 Añadir vídeos.

CU-5.2	Eliminar vídeos
Versión	1.0
Autor	Jorge Martínez Martín
Requisitos asociados	RF-3
Descripción	Tanto los médicos como los pacientes han de ser capaces de eliminar vídeos.
Precondición	<ul style="list-style-type: none"> ■ Debe existir el paciente al que se le quiere eliminar el vídeo. ■ El paciente debe tener al menos un vídeo.
Acciones	<ul style="list-style-type: none"> ■ El usuario pulsa el botón «eliminar vídeo» al lado del vídeo que desea eliminar. ■ Se pulsa el botón «confirmar».
Postcondición	El vídeo ya no está en la lista de vídeos del paciente correspondiente.
Excepciones	
Importancia	Media

Tabla B.9: CU-5.2 Eliminar vídeos.

CU-5.3	Modificar vídeos
Versión	1.0
Autor	Jorge Martínez Martín
Requisitos asociados	RF-4
Descripción	Tanto los médicos como los pacientes han de ser capaces de modificar datos de los vídeos del paciente.
Precondición	<ul style="list-style-type: none"> ■ Debe existir el paciente al que se le quiere modificar el vídeo. ■ El paciente debe tener al menos un vídeo.
Acciones	<ul style="list-style-type: none"> ■ El usuario pulsa el botón «editar vídeo» al lado del vídeo que desea modificar. ■ El usuario modifica los datos que sean necesarios del vídeo seleccionado. ■ El usuario pulsa el botón «confirmar».
Postcondición	El vídeo debe aparecer con las modificaciones realizadas en la lista de vídeos del paciente correspondiente.
Excepciones	
Importancia	Baja

Tabla B.10: CU-5.3 Modificar vídeos.

CU-5.4	Clasificar vídeos
Versión	1.0
Autor	Jorge Martínez Martín
Requisitos asociados	RF-5
Descripción	Tanto los médicos como los pacientes han de ser capaces de clasificar los vídeos del paciente.
Precondición	<ul style="list-style-type: none"> ■ Debe existir el paciente al que se le quiere clasificar el vídeo. ■ El paciente debe tener al menos un vídeo.
Acciones	<ul style="list-style-type: none"> ■ El usuario pulsa el botón «clasificar vídeo». ■ Se clasificará el vídeo correspondiente.
Postcondición	Aparecerá la clasificación del vídeo.
Excepciones	
Importancia	Alta

Tabla B.11: CU-5.4 Clasificar vídeos.

CU-6.1	Añadir paciente
Versión	1.0
Autor	Jorge Martínez Martín
Requisitos asociados	RF-12
Descripción	El médico a de ser capaz de añadir pacientes a su lista de pacientes.
Precondición	<ul style="list-style-type: none"> ■ Debe existir el paciente que se desea añadir.
Acciones	<ul style="list-style-type: none"> ■ El médico pulsa el botón «añadir paciente». ■ Se deberá rellenar los datos del paciente que se desea agregar. ■ El paciente se agregará a la lista de pacientes.
Postcondición	Aparecerá el paciente en la lista del médico.
Excepciones	<ul style="list-style-type: none"> ■ El paciente no existe. ■ El paciente ya se encuentra en la lista del médico.
Importancia	Alta

Tabla B.12: CU-6.1 Añadir paciente.

CU-6.2	Eliminar paciente
Versión	1.0
Autor	Jorge Martínez Martín
Requisitos asociados	RF-13
Descripción	El médico a de ser capaz de eliminar pacientes a su lista de pacientes.
Precondición	<ul style="list-style-type: none"> ■ Debe existir el paciente que se desea eliminar. ■ El paciente debe estar incluido en la lista de pacientes del médico.
Acciones	<ul style="list-style-type: none"> ■ El médico pulsa el botón «eliminar paciente». ■ Se selecciona el paciente que se desea eliminar. ■ Se confirma que se desea eliminar a ese paciente.
Postcondición	Ya no aparecerá el paciente en la lista del médico.
Excepciones	<ul style="list-style-type: none"> ■ El paciente no existe. ■ El paciente no se encuentra en la lista del médico.
Importancia	Media

Tabla B.13: CU-6.2 Eliminar paciente.

Apéndice C

Especificación de diseño

C.1. Introducción

En el siguiente apartado se describen los detalles de la estructura del *software*: el comportamiento y la interacción entre los distintos componentes.

C.2. Diseño de datos

Para implementar la persistencia de los datos se ha hecho uso de una base de datos SQLite debido a que no requiere instalación separada y es fácil de integrar con el framework de Flask. Este diseño se ha dividido en varias fases: diseño del diagrama entidad-relación, diseño del diagrama relacional y diseño del diccionario de datos.

Modelo entidad-relación

Para representar gráficamente las relaciones entre los datos, se ha creado un diagrama E-R (ver figura C.1). En el se puede observar una ISA exclusiva total. Esto se debe a que, tras consultarlo con el profesor Jesús Maudes, se llegó a la conclusión de que crear una ISA exclusiva total era una solución válida debido a que se quiere evitar que un mismo usuario tenga acceso a varias funcionalidades. En caso de que esto se requiriese (si un doctor necesita permisos de administración) se crearía otra cuenta de forma más controlada.

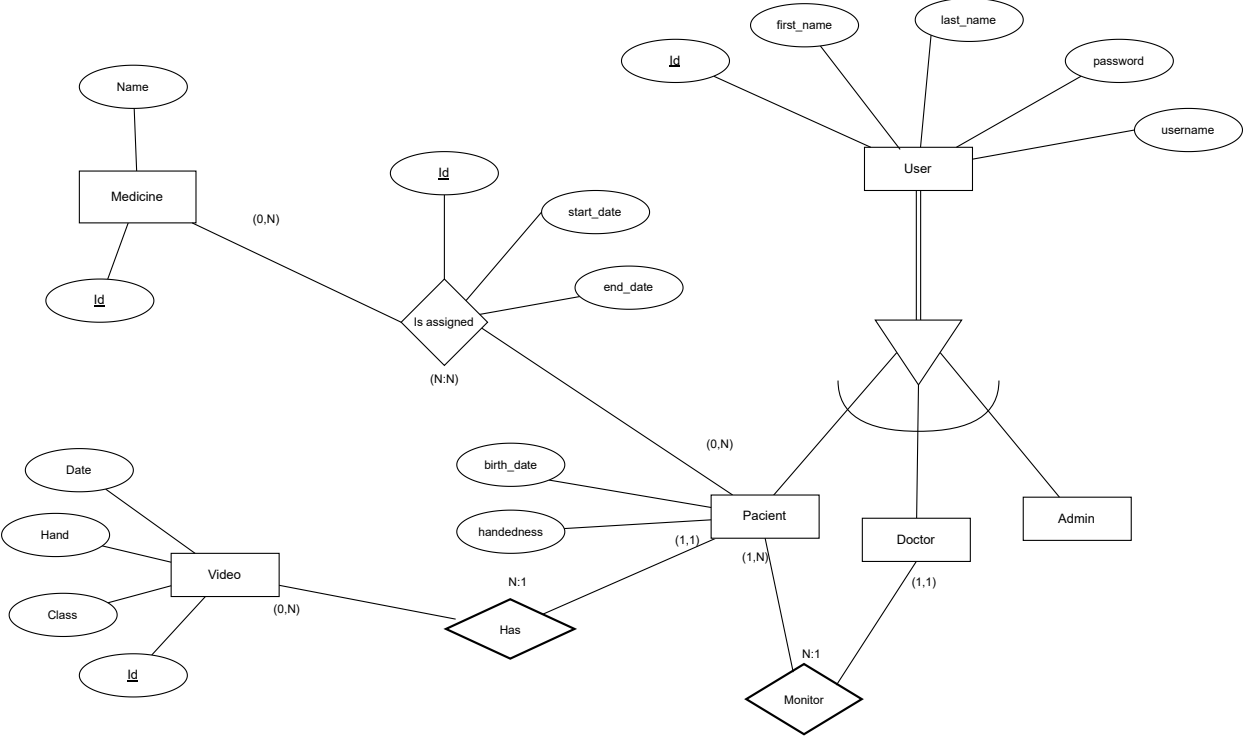


Figura C.1: Diagrama entidad-relación

Modelo relacional

Partiendo del modelo entidad-relación (ver figura C.1) se ha creado el diagrama relacional (ver figura C.2). A continuación se mencionan algunas de las decisiones más importantes que se han decidido tomar a la hora de crear este diagrama. Con respecto a la ISA se ha decidido dividir en tres tablas distintas ya que en un principio un usuario solo va a poder tener uno de los tres roles asignados y en caso de que se necesitase que un doctor tuviera el rol de administrador, que es el un caso excepcional, se le otorgaría otra cuenta distinta con los permisos de administración. De esta manera podremos controlar los accesos de cada usuario de manera más limpia y exclusiva. El rol de administrador es un rol que pese a que no se relacione con ninguna otra entidad, se ha visto que es necesario para poder permitir el acceso a los controles de administrador.

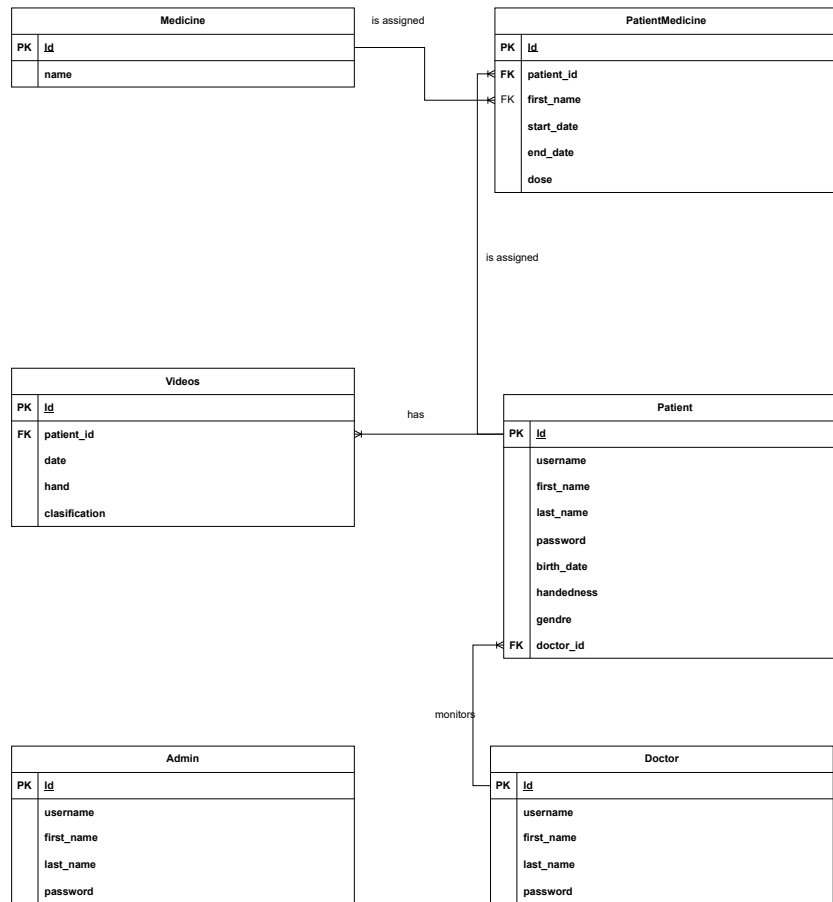


Figura C.2: Diagrama relacional de la aplicación

Diccionario de datos

Se adjunta a continuación, para cada tabla, el diccionario de datos correspondiente.

- **Administradores:** la tabla **C.1** contiene la información correspondiente a los administradores de la aplicación.
- **Doctores:** la tabla **C.2** contiene la información correspondiente a los doctores.
- **Pacientes:** la tabla **C.3** contiene la información correspondiente a los pacientes.

Nombre	Tipo	Columna	Descripción
id	INTEGER	PK	Identificador único del administrador.
username	VARCHAR(100)	UNIQUE NOT NULL	Nombre de usuario del administrador. Debe ser único.
password	VARCHAR(100)	NOT NULL	Contraseña del administrador, cifrada usando sha-256.
first_name	VARCHAR(100)	NOT NULL	Primer nombre del administrador.
last_name	VARCHAR(100)	NOT NULL	Apellidos del administrador.

Tabla C.1: Diccionario de datos. Tabla correspondiente a la clase **Admin**.

Nombre	Tipo	Columna	Descripción
id	INTEGER	PK	Identificador único del doctor.
username	VARCHAR(100)	UNIQUE NOT NULL	Nombre de usuario del doctor. Debe ser único.
password	VARCHAR(100)	NOT NULL	Contraseña del doctor, cifrada usando sha-256.
first_name	VARCHAR(100)	NOT NULL	Primer nombre del doctor.
last_name	VARCHAR(100)	NOT NULL	Apellidos del doctor.

Tabla C.2: Diccionario de datos. Tabla correspondiente a la clase **Doctor**.

Nombre	Tipo	Columna	Descripción
id	INTEGER	PK	Identificador único del paciente.
username	VARCHAR(100)	UNIQUE NOT NULL	Nombre de usuario del paciente. Debe ser único.
password	VARCHAR(100)	NOT NULL	Contraseña del paciente, cifrada usando sha-256.
first_name	VARCHAR(100)	NOT NULL	Primer nombre del paciente.
last_name	VARCHAR(100)	NOT NULL	Apellidos del paciente.
birth_date	DATE	NOT NULL	Fecha de nacimiento del paciente.
handedness	ENUM('right', 'left')	NOT NULL	Mano dominante del paciente (derecha o izquierda).
gender	ENUM('M', 'F')	NOT NULL	Género del paciente (masculino o femenino).
doctor_id	INTEGER	FK(doctor.id)	Clave foránea a la tabla Doctor. Indica el doctor a cargo del paciente.

Tabla C.3: Diccionario de datos. Tabla correspondiente a la clase **Patient**.

Nombre	Tipo	Columna	Descripción
id	INTEGER	PK	Identificador único del video.
patient_id	INTEGER	FK(patient.id)	Clave foránea a la tabla Paciente. Indica el paciente asociado al video.
hand	ENUM('left', 'right')	NOT NULL	Mano usada en el vídeo (izquierda o derecha).
date	DATE	NOT NULL	Fecha de grabación del video.
video_data	LargeBinary	NOT NULL	Datos binarios del vídeo.
classification	VARCHAR(100)	NULL	Clasificación del vídeo (opcional).

Tabla C.4: Diccionario de datos. Tabla correspondiente a la clase **Video**.

Nombre	Tipo	Columna	Descripción
id	INTEGER	PK	Identificador único del medicamento.
name	VARCHAR(100)	NOT NULL	Nombre del medicamento.

Tabla C.5: Diccionario de datos. Tabla correspondiente a la clase **Medicine**.

Nombre	Tipo	Columna	Descripción
id	INTEGER	PK	Identificador único de la asociación entre paciente y medicamento.
patient_id	INTEGER	FK(patient.id)	Clave foránea a la tabla Paciente. Indica el paciente que toma el medicamento.
medicine_id	INTEGER	FK(medicine.id)	Clave foránea a la tabla Medicamento. Indica el medicamento que toma el paciente.
dosage	VARCHAR(100)	NOT NULL	Dosis del medicamento para el paciente.
start_date	DATE	NOT NULL	Fecha de inicio del tratamiento con el medicamento.
end_date	DATE	NULL	Fecha de fin del tratamiento con el medicamento (opcional).

Tabla C.6: Diccionario de datos. Tabla correspondiente a la clase **PatientMedicine**.

- **Vídeos:** la tabla [C.4](#) contiene la información correspondiente a los vídeos de los pacientes.
- **Medicinas:** la tabla [C.5](#) contiene la información de las medicinas.
- **PacienteMedicinas:** la tabla [C.6](#) contiene la información de las medicinas administradas a cada paciente incluyendo la fecha de inicio del tratamiento así como la fecha de finalización.

C.3. Diseño procedimental

C.4. Diseño arquitectónico

Apéndice D

Documentación técnica de programación

- D.1. Introducción
- D.2. Estructura de directorios
- D.3. Manual del programador
- D.4. Compilación, instalación y ejecución del proyecto
- D.5. Pruebas del sistema

Apéndice E

Documentación de usuario

- E.1. Introducción
- E.2. Requisitos de usuarios
- E.3. Instalación
- E.4. Manual del usuario

Apéndice F

Anexo de sostenibilización curricular

F.1. Introducción

Este anexo incluirá una reflexión personal del alumnado sobre los aspectos de la sostenibilidad que se abordan en el trabajo. Se pueden incluir tantas subsecciones como sean necesarias con la intención de explicar las competencias de sostenibilidad adquiridas durante el alumnado y aplicadas al Trabajo de Fin de Grado.

Más información en el documento de la CRUE https://www.crue.org/wp-content/uploads/2020/02/Directrices_Sostenibilidad_Crue2012.pdf.

Este anexo tendrá una extensión comprendida entre 600 y 800 palabras.

Bibliografía

- [1] Taha Khan, Dag Nyholm, Jerker Westin, and Mark Dougherty. A computer vision framework for finger-tapping evaluation in parkinson's disease. *Artificial intelligence in medicine*, 60(1):27–40, 2014.
- [2] Marta Palacio. *Scrum Master. Temario troncal 1*. 02 2022.
- [3] Stefan Williams, Samuel D Relton, Hui Fang, Jane Alty, Rami Qahwaji, Christopher D Graham, and David C Wong. Supervised classification of bradykinesia in parkinson's disease from smartphone videos. *Artificial Intelligence in Medicine*, 110:101966, 2020.
- [4] Stefan Williams, Zhibin Zhao, Awais Hafeez, David C Wong, Samuel D Relton, Hui Fang, and Jane E Alty. The discerning eye of computer vision: Can it measure parkinson's finger tap bradykinesia? *Journal of the Neurological Sciences*, 416:117003, 2020.