



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

DSIC
DEPARTAMENT DE SISTEMES
INFORMÀTICS I COMPUTACIÓ

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Dpto. de Sistemas Informáticos y Computación

Desarrollo de una Plataforma Software Inteligente para el
Análisis & Gestión de Datos Clínicos-Genómicos sobre las
Enfermedades de la Retina-Mácula

Trabajo Fin de Máster

Máster Universitario en Ingeniería y Tecnología de Sistemas
Software

AUTOR/A: Carreño Bolufer, Jesús

Tutor/a: Casamayor Rodenas, Juan Carlos

Cotutor/a: Reyes Román, José Fabián

CURSO ACADÉMICO: 2022/2023

Desarrollo de una Plataforma Software Inteligente para el Análisis & Gestión de Datos Clínicos-Genómicos sobre las Enfermedades de la Retina-Mácula

Development of an Intelligent Software Platform for the Analysis & Management of Clinical-Genomic Data on Retina-Taint Diseases

Desenvolupament d'una Plataforma Software Intel·ligent per a l'Anàlisi & Gestió de Dades Clínic-Genòmics sobre les Malalties de la Retina-Màcula

Trabajo de Fin de Máster realizado por Jesús Carreño Bolufer en septiembre de 2023 para la obtención del título Máster Universitario en Ingeniería y Tecnología de Sistemas Software por la Universitat Politècnica de València

Tutores: Juan Carlos Casamayor Rodenas y José Fabián Reyes Román

Resumen

La mácula es una pequeña zona ubicada en la retina responsable de tareas cotidianas como la lectura, la conducción y la identificación de personas. Las enfermedades de la mácula impactan en una pérdida de la agudeza visual y actualmente, son la causa líder de ceguera legar en el mundo. La principal es la Degeneración de la Mácula Asociada a la Edad (DMAE), una enfermedad irreversible en la que la edad es un factor de riesgo. Además, el coste elevado de los fármacos para su tratamiento supone un problema serio de eficiencia económica en la red pública de hospitales.

La presente propuesta de TFM está enmarcada dentro del grupo «Genoma» del Centro de Investigación en Métodos de Producción de Software (PROS), miembro del Instituto Valenciano de Inteligencia Artificial (VRAIN) de la Universitat Politècnica de València (UPV), y tiene como objetivo principal extender una solución *software* para permitir el análisis y gestión de los datos (clínicos y genómicos) de enfermedades de la mácula. Mediante el desarrollo de este sistema de información (SI) se pretende: i) mejorar la gestión de los medicamentos de alto impacto económico, que precisan estas enfermedades, utilizados en los Servicios de Oftalmología (SO) de la Conselleria de Sanidad; y ii) integrar todas las variables clínicas de los y las pacientes para hacer estudios epidemiológicos (agrupados por género, enfermedad, sexo y edad), con la finalidad de mejorar la atención, optimización de las consultas, indicaciones de tratamiento, mejorar el coste-beneficio y realizar un seguimiento más personalizado (creando perfiles de pacientes).

Para el desarrollo de la nueva versión del SI, se hace necesario el análisis de requisitos y la aplicación de técnicas avanzadas de gestión en las que el modelado conceptual y la caracterización ontológica de la información manipulada, aseguren la consistencia, corrección y una adecuada explotación de las bases de datos diseñadas para almacenar dicha información. Para cumplir con las necesidades de los usuarios finales de la forma más eficaz posible, en este TFM se ha colaborado con un experto del Servicio de Oftalmología (SO) del Hospital Vega Baja de Orihuela, el cual participa durante la investigación del problema, así como en la validación de la solución.

Palabras clave: Sistema de Información; Retina-Mácula; Modelado Conceptual; Ciencia de Datos Genómicos; Datos Clínicos; Genómica; Ingeniería del Software

Resum

La màcula és una xicoteta zona situada en la retina responsable de tasques quotidianes com la lectura, la conducció i la identificació de persones. Les malalties de la màcula impacten en una pèrdua de l'agudesesa visual i actualment, són la causa líder de ceguesa llegar en el món. La principal és la Degeneració de la Màcula Associada a l'Edat (DMAE), una malaltia irreversible en la qual l'edat és un factor de risc. A més, el cost elevat dels fàrmacs per al seu tractament suposa un problema seriós d'eficiència econòmica en la xarxa pública d'hospitals.

La present proposta de TFM està emmarcada dins del grup «Genoma» del Centre d'Investigació en Mètodes de Producció de Programari (PROS), membre de l'Institut Valencià d'Intel·ligència Artificial (VRAIN) de la Universitat Politècnica de València (UPV), i té com a objectiu principal estendre una solució programari per a permetre l'anàlisi i gestió de les dades (clínic i genòmic) de malalties de la màcula. Mitjançant el desenvolupament d'aquest sistema d'informació (SI) es pretén: i) millorar la gestió dels medicaments d'alt impacte econòmic, que precisen aquestes malalties, utilitzats en els Serveis d'Oftalmologia (SO) de la Conselleria de Sanitat; i ii) integrar totes les variables clíniques dels i les pacients per a fer estudis epidemiològics (agrupats per gènere, malaltia, sexe i edat), amb la finalitat de millorar l'atenció, optimització de les consultes, indicacions de tractament, millorar el cost-benefici i realitzar un seguiment més personalitzat (creant perfils de pacients).

Per al desenvolupament de la nova versió del SI, es fa necessari l'anàlisi de requisits i l'aplicació de tècniques avançades de gestió en les quals el modelatge conceptual i la caracterització ontològica de la informació manipulada, asseguren la consistència, correcció i una adequada explotació de les bases de dades dissenyades per a emmagatzemar aquesta informació. Per a complir amb les necessitats dels usuaris finals de la forma més eficaç possible, en aquest TFM s'ha col·laborat amb un expert del Servei d'Oftalmologia (SO) de l'Hospital Vega Baixa d'Oriola, el qual participa durant la investigació del problema, així com en la validació de la solució.

Paraules clau: Sistema d'Informació; Retina-Màcula; Modelat Conceptual; Ciència de Dades Genòmiques; Dades Clíniques; Genòmica; Enginyeria del Software

Abstract

The macula is a small area in the retina responsible for everyday tasks such as reading, driving and identifying people. Diseases of the macula result in a loss of visual acuity and are currently the leading cause of long-term blindness in the world. The leading one is Age-Related Macular Degeneration (AMD), an irreversible disease in which age is a risk factor. Moreover, the high cost of drugs for its treatment is a serious problem of economic efficiency in the public hospital network.

This TFM proposal is part of the "Genome" group of the Centre for Research in Software Production Methods (PROS), a member of the Valencian Institute of Artificial Intelligence (VRAIN) of the Universitat Politècnica de València (UPV), and its main objective is to extend a software solution to enable the analysis and management of data (clinical and genomic) on macular diseases. The development of this information system (IS) aims to: i) improve the management of medicines with a high economic impact, which these diseases require, used in the Ophthalmology Services (OS) of the Regional Ministry of Health; and ii) integrate all the clinical variables of the patients to carry out epidemiological studies (grouped by gender, disease, sex and age), with the aim of improving care, optimizing consultations, treatment indications, improving cost-benefit and carrying out a more personalized follow-up (creating patient profiles).

For the development of the new version of the IS, it is necessary to analyse requirements and apply advanced management techniques in which conceptual modelling and ontological characterisation of the information handled ensure consistency, correctness and adequate exploitation of the databases designed to store this information. In order to meet the needs of end users as effectively as possible, in this TFM we have collaborated with an expert from the Ophthalmology Service (SO) of the Vega Baja Hospital in Orihuela, who participates during the investigation of the problem, as well as in the validation of the solution.

Keywords: Information System; Retina-Macula; Conceptual Modeling; Genomic Data Science; Clinical Data; Genomics; Software Engineering

Índice general

Motivación	1
1.1. Problema de investigación	1
1.2. Objetivos	2
1.3. Metodología de Investigación	3
1.4. Estructura del proyecto	6
Estado del arte	7
2.1. Marco conceptual	7
2.2. Trabajos relacionados	13
Obtención de Variaciones Relevantes asociadas a DMAE: Uso Práctico del Oráculo	15
3.1. Introducción	15
3.2. Materiales y métodos usados	16
3.3. Resultados	17
3.4. Discusión	19
3.5. Conclusiones	20
Especificación de requisitos	21
4.1. Requisitos funcionales	21
4.2. Requisitos no funcionales	23
4.3. Diagramas de casos de uso	24
4.4. Mockups	39
4.5. Funcionalidad clínico-genómica: gestión de variaciones y relevancia clínica	47
Modelo Conceptual de la Patología Macular (Extensión)	51
5.1. Primer modelo	51
5.2. Modelo extendido	52
5.3. Modelo extendido con la Vista Genómica	56
Diseño e implementación de la solución	60
6.1. Diseño de la arquitectura	60
6.2. Diseño de la parte del cliente (frontend)	61
6.3. Diseño de la parte del servidor (backend)	63
Despliegue	68

7.1. Despliegue del prototipo AGC-MAC	68
Pruebas y validación de la solución	70
8.1. Prueba de usabilidad	70
8.2. Evaluación	72
Resultados	74
9.1. Representación visual de la solución	74
9.2. Experiencia en el desarrollo	84
Conclusiones	86
10.1. Contribuciones principales	86
10.2. Trabajo futuro	86
Bibliografía	88
Anexo I – Prototipo G-MAC	92
Anexo II – Instrucciones de la prueba de usabilidad	97
Anexo III – Fragmentos de código	99

Índice de figuras

Ilustración 1 Iteración entre dos actividades de la metodología Design Science [21]	3
Ilustración 2 Framework para Design Science aplicado al proyecto de investigación	4
Ilustración 3 Ciclo de ingeniería de la metodología Design Science	5
Ilustración 4 Nueva clasificación clínica de la DMAE (Imagen de la Sociedad Española de Retina y Vítreo) [13].....	8
Ilustración 5 Manifestaciones de la DMAE observadas en la fotografía en color del fondo de ojo [4].....	9
Ilustración 6 Estrategia de diagnóstico, tratamiento y seguimiento del paciente con DMAE precoz (Imagen de la Sociedad Española de Retina y Vítreo) [13].....	11
Ilustración 7 Estrategia de diagnóstico, tratamiento y seguimiento del paciente con DMAE intermedia (Imagen de la Sociedad Española de Retina y Vítreo)[13].....	12
Ilustración 8 Variaciones clasificadas por cromosoma.....	17
Ilustración 9 Variaciones clasificadas por tipo	18
Ilustración 10 Variaciones clasificadas por relevancia clínica	18
Ilustración 11 Variaciones del fenotipo visualizadas en Sibila.....	19
Ilustración 12 Diagrama de CU - Acceso a la aplicación y gestión de médicos y hospitales	25
Ilustración 13 Diagrama de CU - Gestión de pacientes	30
Ilustración 14 Diagrama de CU - Gráficas de población de pacientes.....	38
Ilustración 15 Mockup Resumen del paciente y acciones rápidas.....	41
Ilustración 16 Mockup Revisiones del paciente	41
Ilustración 17 Mockup Nueva revisión colapsado.....	42
Ilustración 18 Mockup Ventana modal para la creación de objetos	42
Ilustración 19 Mockup Nueva revisión desplegado.....	43
Ilustración 20 Mockup Gráficas del paciente	43
Ilustración 21 Mockup enfermedades del paciente.....	44
Ilustración 22 Mockup Antecedentes médicos del paciente	44
Ilustración 23 Mockup Datos antropométricos del paciente.....	45
Ilustración 24 Mockup Gráficas de la población de pacientes del médico	46
Ilustración 25 Mockup Detalle de revisión del paciente.....	47
Ilustración 26 Diagrama de CU con gestión de variaciones genómicas y relevancia clínica (Parcial).....	48
Ilustración 27 Mockup Gestión de variaciones genómicas e información de la relevancia clínica	50
Ilustración 28 MCPM del prototipo G-MAC [1].....	52
Ilustración 29 Modelo extendido MCPM	56
Ilustración 30 Modelo Conceptual del Genoma (CSG) para una mejor representación de la relevancia clínica (García Simón, 2022).....	58
Ilustración 31 Unión del MCPM y del CSG para la relevancia clínica	59
Ilustración 32 Arquitectura del prototipo G-MAC (Imagen del TFM de Alejandro Marco [1])	60
Ilustración 33 Arquitectura del prototipo AGC-MAC.....	61
Ilustración 34 Estructura del proyecto de frontend.....	63
Ilustración 35 AGC-MAC Esquema de la base de datos relacional	66
Ilustración 36 Resultado de los tests de integración del backend	67
Ilustración 37 Despliegue del prototipo AGC-MAC en los servidores PROS/UPV.....	69
Ilustración 38 Cuestionario Sistem Usability Scale (SUS) (J. Brook, 1995).....	71
Ilustración 39 AGC-MAC Login o inicio de sesión	75
Ilustración 40 AGC-MAC Ventana Gráficas de la población de pacientes	76
Ilustración 41 AGC-MAC Ventana Pacientes	76
Ilustración 42 AGC-MAC Formulario de creación de un paciente	77
Ilustración 43 AGC-MAC Ventana Pacientes inhabilitados.....	77
Ilustración 44 AGC-MAC Ventana Resumen de paciente	78
Ilustración 45 AGC-MAC Ventana Revisiones de paciente.....	78
Ilustración 46 AGC-MAC Ventana Gráficas de paciente.....	79

Ilustración 47 AGC-MAC Ventana Enfermedades de paciente.....	79
Ilustración 48 AGC-MAC Eliminación de una enfermedad.....	80
Ilustración 49 AGC-MAC Ventana Antecedentes médicos de paciente.....	80
Ilustración 50 AGC-MAC Ventana Datos antropométricos de paciente	81
Ilustración 51 AGC-MAC Ventana Revisión	81
Ilustración 52 AGC-MAC Ventana médicos.....	82
Ilustración 53 AGC-MAC Ventana Médicos inhabilitados	82
Ilustración 54 AGC-MAC Ventana Médico	83
Ilustración 55 AGC-MAC Ventana Hospitales	83
Ilustración 56 AGC-MAC Ventana Hospital.....	84
Ilustración 57 AGC-MAC Ventana Perfil	84
Ilustración 58 G-MAC Login o inicio de sesión.....	92
Ilustración 59 G-MAC Ventana hospitales.....	92
Ilustración 60 G-MAC Ventana médicos	93
Ilustración 61 G-MAC Ventana Pacientes.....	93
Ilustración 62 G-MAC Ventana detalle del paciente	93
Ilustración 63 G-MAC Ventana Gestión de enfermedades de un paciente.....	94
Ilustración 64 G-MAC Listado de revisiones	94
Ilustración 65 G-MAC Ventana detalle de una revisión	94
Ilustración 66 G-MAC Gestión de escáneres de retina.....	95
Ilustración 67 G-MAC Gestión de mediciones.....	95
Ilustración 68 G-MAC Visualización de gráficas.....	96

Índice de tablas

Tabla 1 Variaciones obtenidas con el módulo Hermes (Mayo de 2023)	17
Tabla 2 Variaciones clasificadas por evidencia por Ulises	18
Tabla 3 Requisitos funcionales	23
Tabla 4 Requisitos no funcionales del AGC-MAC	24
Tabla 5 CU Listar hospitales	26
Tabla 6 CU Buscar hospitales.....	26
Tabla 7 CU Ver hospital	26
Tabla 8 CU Listar médicos asociados al hospital	26
Tabla 9 CU Dar de alta un hospital.....	27
Tabla 10 CU Listar médicos	27
Tabla 11 CU Listar médicos inhabilitados.....	27
Tabla 12 CU Dar de alta un médico	28
Tabla 13 CU Ver médico	28
Tabla 14 CU Inhabilitar médico	28
Tabla 15 CU Habilitar médico.....	29
Tabla 16 CU Editar médico	29
Tabla 17 CU Listar pacientes al cargo	30
Tabla 18 CU Buscar pacientes al cargo	30
Tabla 19 CU Crear paciente	31
Tabla 20 CU Añadir paciente existente	31
Tabla 21 CU Ver paciente	31
Tabla 22 CU Resumen del paciente y acciones rápidas.....	32
Tabla 23 CU Editar datos personales del paciente.....	32
Tabla 24 CU Gráficas del paciente	33
Tabla 25 CU Listar enfermedades del paciente	33
Tabla 26 CU Gestionar enfermedades del paciente	33
Tabla 27 CU Listar antecedentes médicos del paciente.....	34
Tabla 28 CU Gestionar antecedentes médicos del paciente.....	34
Tabla 29 CU Listar datos antropométricos del paciente	35
Tabla 30 CU Gestionar datos antropométricos del paciente	35
Tabla 31 CU Listar revisiones del paciente	35
Tabla 32 CU Ver revisión.....	35
Tabla 33 CU Crear revisión.....	36
Tabla 34 CU Gestionar revisión	36
Tabla 35 CU Listar todos los pacientes	37
Tabla 36 CU Listar pacientes inhabilitados	37
Tabla 37 CU Buscar pacientes.....	37
Tabla 38 CU Ver paciente con vista administrador	37
Tabla 39 CU Inhabilitar paciente.....	38
Tabla 40 CU Habilitar paciente	38
Tabla 41 CU Gráficas de la población de pacientes al cargo.....	39
Tabla 42 CU Gráficas de la población de todos los pacientes	39
Tabla 43 CU Listar variaciones	48
Tabla 44 CU Gestionar variaciones	49
Tabla 45 Listar relevancias clínicas de las variaciones en las enfermedades del paciente	49
Tabla 46 Puntos de entrada y métodos permitidos	65

Capítulo 1

Motivación

Este Trabajo de Fin de Máster (TFM) forma parte de un **proyecto de innovación** del grupo **Genoma del Centro PROS**, el cual tiene como objetivo el desarrollo de una plataforma *software* para el análisis, gestión, optimización y mejora de la calidad asistencial de las enfermedades de la retina-mácula (principales causas de ceguera).

La mácula, ubicada en el centro de la retina, desempeña un papel crucial en la agudeza visual, esencial para tareas cotidianas como la lectura, la conducción y la identificación de personas, lo que nos permite percibir detalles y colores [5]. Cuando esta área se daña, la autonomía y la calidad de vida de los pacientes se ven afectadas, empeorando la dependencia existente y aumentando el riesgo de depresión, así como reduciendo la esperanza de vida. Las enfermedades de la mácula son la causa líder de las pérdidas de visión en el mundo desarrollado, ya que causan una pérdida gradual e irreversible de la visión que puede llevar a la ceguera [8].

Este capítulo se organiza en cuatro secciones fundamentales:

1. *Una exposición del problema en cuestión, destacando la importancia de los Sistemas de Información en el ámbito clínico.*
2. *Como consecuencia del problema planteado, se establecen unos objetivos generales y específicos de este TFM para contribuir al proyecto.*
3. *La metodología de investigación aplicada durante el desarrollo de este TFM.*
4. *La descripción de la estructura de la memoria para documentar el trabajo realizado.*

1.1. Problema de investigación

Las **enfermedades maculares**, es decir, las enfermedades que afectan a la mácula han pasado de ser enfermedades generales a crónicas en las últimas décadas. Este acontecimiento es consecuencia del aumento de la esperanza de vida de la población, ya que desarrollan principalmente por el desgaste de la retina [8]. El aumento de la prevalencia de este tipo de enfermedades implica el tratamiento de un nuevo y voluminoso grupo de pacientes. Ante este aumento, existe mucho desconocimiento acerca de su tratamiento, sobre todo, por parte de los doctores. Los expertos clínicos recurren a la experiencia personal o a la recomendación de compañeros de profesión cuando se han de tomar decisiones acerca del tratamiento de un paciente con alguna enfermedad macular. Entre algunas decisiones, por ejemplo, si cambiar el medicamento o no cuando la agudeza visual no mejora.

Si nos centramos en los **medicamentos** o **tratamientos** que se aplican al paciente, la falta de evidencia implica que en muchos casos no sean tratados con la mejor opción y posiblemente, resultando contraproducente en la evolución de la visión. Conjuntamente, el coste de los medicamentos utilizados hoy en día supone un gasto elevado y en aumento cada año. Para observar con perspectiva, el gasto farmacéutico en 2019 en principios activos de administración oftálmica para el tratamiento de la Degeneración Macular Asociada a la Edad (DMAE), ocupaba

la posición 14 (de 15) con 115M € en los subgrupos químicos con mayor gasto hospitalario en el Sistema Nacional de Salud (*Informe Anual del Sistema Nacional de Salud 2020-21*). Por tanto, la **eficiencia económica** debe estar en el foco.

Para conseguir reducir el consumo de medicamentos de forma significativa, la tecnología es de vital importancia. Hoy en día, no se puede concebir la gestión de un gran número de datos clínicos sin la ayuda de un SI. Aunque parezca sorprendente, esto es lo que ocurre en algunos hospitales de la *Comunitat Valenciana* (CV) en el ámbito de la oftalmología. En cambio, hacen uso de documentos en formato físico o *softwares* generales (p.e., *Orion Clinic* [43]) y engorrosos que incurren en problemas de eficiencia temporal, entre otros. Estas dificultades son aún más graves teniendo en cuenta el contexto de la atención primaria, donde el tiempo de las consultas es limitado.

Un **Sistema de Información** no solo puede estar capacitado para la gestión de los pacientes, también puede proveer de conocimiento que hasta ahora no habían sido capaces de generar a causa del ineficiente método de trabajar con los datos clínicos. Por ejemplo, permitiría saber cuál ha sido el gasto total del hospital o incluso recomendar el medicamento más eficiente para un perfil de paciente. También puede ayudar a mejorar a conocer el historial de un paciente que ha sido tratado por otro médico.

En resumen, encontramos un problema de eficiencia en la **gestión y análisis de datos de pacientes** con enfermedades maculares, tanto temporal como económica. Para hacer frente a estos problemas, este TFM pretende hacerlo extendiendo el prototipo **G-MAC**, una solución software que se desarrolló como versión inicial (TRL 3-4) con este fin [1]. Aparte del propio artefacto, se hará especial atención a la enfermedad de **Degeneración de la Mácula Asociada a la Edad** (DMAE) con un análisis genómico actualizado usando herramientas bioinformáticas para evaluar su impacto desde una perspectiva de Medicina de Precisión [48](Capítulo 3).

1.2. Objetivos

El **objetivo principal** de este trabajo de investigación consiste en la extensión del prototipo **G-MAC**, para mejorar la captura y análisis de datos de los pacientes que presentan **enfermedades de la mácula**. Actualmente se encuentra en nivel de madurez **TRL4**(en fase de validación en un entorno relevante) y se espera que un futuro alcance el nivel **TRL7** (pruebas con éxito en un entorno real).

La nueva versión, **AGC-MAC**, pretende mejorar significativamente a la anterior con una solución más completa, usable y con funcionalidades que aporten nuevo conocimiento. Un conocimiento que ayude a entender mejor las enfermedades maculares y que proporcione herramientas a los médicos para facilitar la toma de decisiones y personalizar los tratamientos. Además, un *software* que incluya características de relevancia en la investigación, como es la gestión de **datos genómicos**, y ayuden a mitigar este problema.

Asimismo, al integrar todas las variables **clínicas y genómicas** de los y las pacientes, se podrán hacer estudios epidemiológicos con la finalidad de mejorar el diagnóstico, la atención, optimización de las consultas, tiempos de espera, indicaciones de tratamiento, mejorar el coste-beneficio y realizar un seguimiento más personalizado, creando perfiles de pacientes.

Para cumplir con este objetivo principal, se han determinado una serie de objetivos específicos:

- 1) Realizar un estudio y análisis del dominio junto a expertos clínicos con el fin de extender el Modelo Conceptual de la Patología Macular (**MCPM**, versión 1) [2] orientado a la agregación de datos para el posterior análisis.
- 2) Realizar la unión entre el modelo **MCPM** y el **Modelo del Genoma Humano** [18,17] que sirva como base para plantear una funcionalidad de medicina personalizada en la nueva versión.
- 3) Realizar un estudio de las **variaciones genómicas** que afectan en el desarrollo o progresión de la **DMAE** con el uso del Oráculo Genómico de Delfos (desarrollado en el Centro PROS) [16] y la aplicación de la metodología **SILE** [27].
- 4) Implementar la nueva versión de la plataforma software AGC-MAC basada en el MCPM extendido (versión 2.0) con las tecnologías actualmente establecidas en la industria.
- 5) Analizar la solución desarrollada con el fin de realizar una comparativa con el prototipo inicial G-MAC.
- 6) Desplegar la aplicación en los servidores de la UPV y realizar **pruebas de usuario** en los hospitales con el objetivo de evaluar la posible utilidad del prototipo AGC-MAC en un entorno real.

1.3. Metodología de Investigación

En este TFM ha seguido la metodología de investigación “*Design Science*” [21] aplicando en su desarrollo un ciclo de ingeniería. Esta metodología se usa comúnmente en la investigación para el desarrollo de SI a través de la creación de artefactos en un contexto específico. El artefacto en sí no resuelve nada, es la interacción en el contexto lo que contribuye a solucionar un problema. Observamos que los dos elementos principales de *Design Science* son el “*artefacto*” y el “*contexto*”.

El artefacto de este trabajo, teniendo en cuenta los objetivos marcados, es: «*Sistema de Información para la gestión y tratamiento a pacientes con enfermedades maculares*».

Por lo que respecta al contexto, se compone de dos puntos:

- **El clínico-oftalmológico**
- **El clínico-genómico**

Este se debe a que, aunque el objetivo principal de este proyecto es el desarrollo de un producto *software* en el ámbito clínico de la oftalmología, también se profundiza en los aspectos genómicos de las enfermedades maculares con la propuesta de una funcionalidad que une el modelo conceptual clínico-genómico.

Design Science se divide en dos partes, el diseño y la investigación. Cada una corresponde con dos tipos diferentes de actividades: problemas de diseño (*Design problems*) y preguntas de investigación (*Research questions*), donde una puede estar motivada por la otra (Ilustración 1)

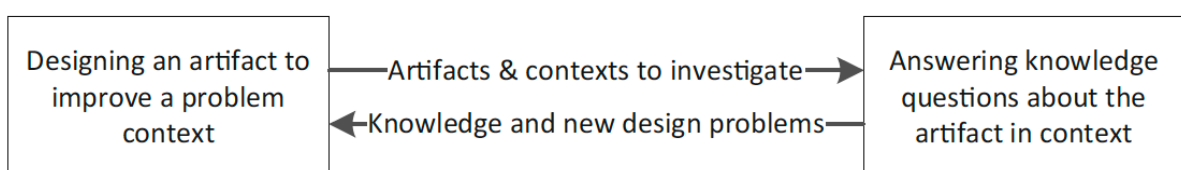


Ilustración 1 Iteración entre dos actividades de la metodología *Design Science* [21]

En primer lugar, tenemos la **pregunta de investigación** que surge de los objetivos de este TFM: «¿El prototipo inicial es apto para la gestión y análisis de pacientes con enfermedades maculares?» cuya respuesta, que se obtiene de analizar los resultados del trabajo previo, es «No, es un prototipo que carece de todas las funcionalidades que los expertos clínicos necesitan». A continuación, se genera el **problema de diseño**: diseñar una nueva versión de un sistema de información para lograr una mejor gestión y análisis de pacientes con enfermedades maculares.

La metodología extiende el problema del contexto con un marco en el cual se definen unos elementos concretos, que influyen en la interacción entre las actividades de diseño e investigación:

- **Contexto social:** contiene todas las personas, grupos u organización que tienen interés o influencia en el proyecto. Como los usuarios finales o los financiadores.
- **Contexto de conocimiento:** se refiere al conjunto de conocimientos, teorías, y conceptos que influyen en el desarrollo y diseño del artefacto. Proporciona un marco para entender los problemas y ser capaz de generar una solución de manera informada y efectiva.

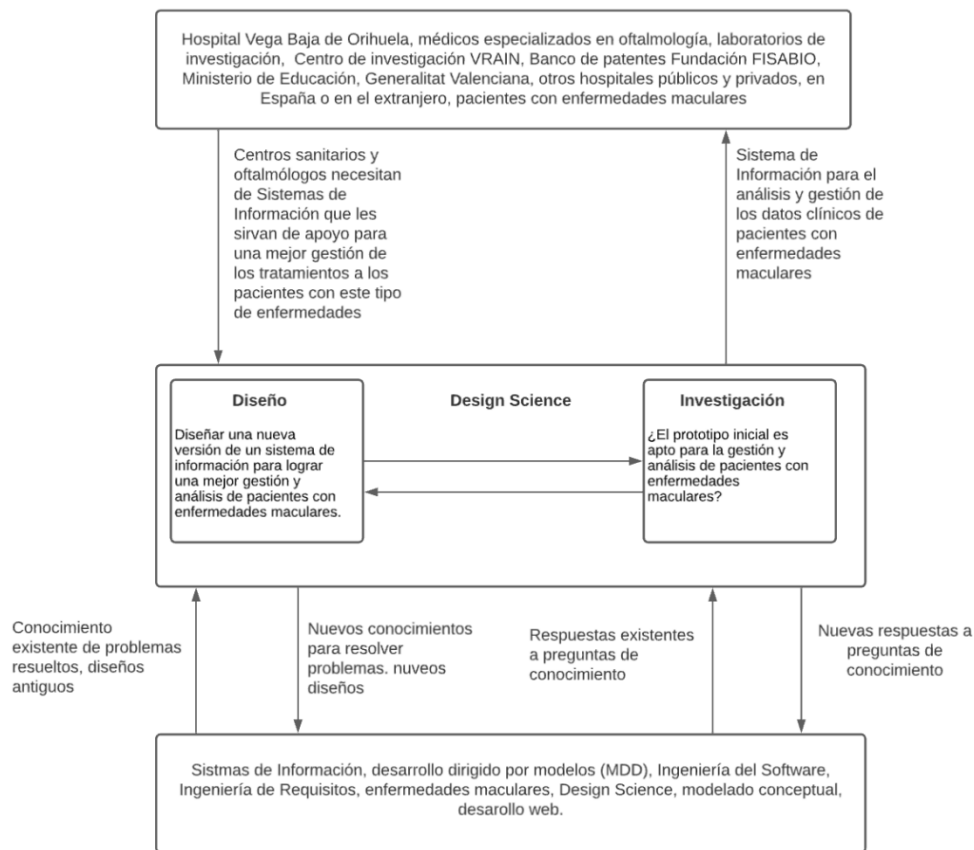


Ilustración 2 Framework para Design Science aplicado al proyecto de investigación

El contexto social de este trabajo se compone de un conjunto de personas, grupos u organizaciones las cuáles se han agrupado en las siguientes categorías:

- **Personas y organizaciones de la salud:** Para el desarrollo de este proyecto, hemos contado con el equipo de expertos conformado por los siguientes centros sanitarios y personal asistencial:

- Hospital Vega Baja de Orihuela
 - Dr. Rubén Cabrera Beyrouti
- **Cientes:** Hospitales de la red pública de la Comunidad Valenciana también podrían llegar a verse beneficiados.
- **Organizaciones:** VRAIN, Centro PROS, Universitat Politècnica de Valencia, Fundación FISABIO.
- **Mercado potencial:** Hasta hospitales ubicados en otros países, de dentro y fuera de la Unión Europea.

Según el tipo de proyecto, se trabaja sobre un ciclo de ingeniería o de investigación. En este caso, se trata un **ciclo de ingeniería**, donde la actividad de diseño se descompone en tres actividades: diseño del tratamiento, validación del tratamiento e implementación del tratamiento. Por tanto, resultando en cuatro actividades que se muestran en la Ilustración 3.

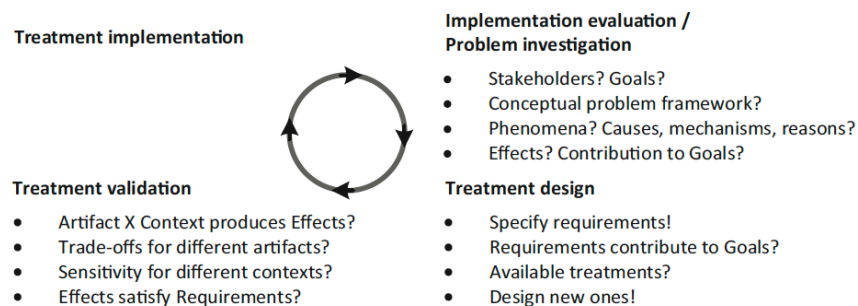


Ilustración 3 Ciclo de ingeniería de la metodología Design Science

En primer lugar, se realiza una evaluación de la implementación y una investigación del problema. Partimos de un prototipo inicial G-MAC en nivel **TRL4** desarrollado en el año 2020 del que se extrajeron una serie de mejoras tras una validación por expertos de diferentes hospitales. Tras este tiempo y con el proyecto reanudado, se realiza una reunión con el Dr. Rubén Cabrera, del Hospital de Vega Baja de Orihuela, para extraer conocimiento e investigar más acerca de los problemas expuestos. Se pudo conocer más acerca del problema de gestión y análisis de los pacientes afectados de la mácula, del elevado impacto económico de los medicamentos en este grupo de enfermedades y del dominio de la patología macular. Además, se realiza un recorrido por la aplicación para que expliquen de primera mano los problemas de usabilidad que surgieron en su momento. El problema principal es la incapacidad de agregar datos para realizar un análisis sobre un paciente o la población de pacientes. Esto se debe a que no disponen de las herramientas suficientes para ello. Por ejemplo, en algunos hospitales de la CV se utiliza el *software* de propósito general Orion Clinic [43], que no está actualizado a las necesidades de los médicos. Además de la gestión actual de pacientes ineficiente.

En segundo lugar, tras conocer a fondo el problema, se extiende el Modelo de la Patología Macular (MCPM, versión 2) para representar su dominio y se especifican los requisitos que ayuden a cumplir los objetivos de la forma más eficaz posible. Para ello, i) se diseñan *mockups*, si son necesarios, que cumplan los requisitos funcionales y no funcionales especificados antes de continuar a la siguiente fase, ii) si los *mockups* son validados positivamente, en este caso por el Dr. Rubén Cabrera, se pasa a la implementación del tratamiento, y iii) se realiza una evaluación de la implementación.

1.4. Estructura del proyecto

Después de definir el problema de investigación, los objetivos y la metodología aplicada, se presenta, a continuación, la estructura de la memoria este Trabajo de Fin de Máster:

- **Capítulo 2:** se presenta un estudio sobre el estado del arte asociado a este trabajo de investigación, de la patología macular y los sistemas de información. En primer lugar, se introduce la enfermedad de la degeneración de la mácula asociada a la edad, junto con sus factores de riesgo y tratamientos actuales y emergentes. En segundo lugar, se presentan una serie de estudios relacionados con el tratamiento de enfermedades con el desarrollo de SI.
- **Capítulo 3:** en este capítulo se realiza una obtención de las variaciones genómicas de la DMAE con herramientas semiautomáticas especializadas en la explotación de la evidencia actual y se analiza su posible uso en la gestión de datos genómicos.
- **Capítulo 4:** se presentan los resultados obtenidos de la fase de Especificación de Requisitos. En primer lugar, se detalla la extensión del Modelo de la Patología Macular que sirve como base para adaptarse las necesidades del problema. En segundo lugar, se exponen los requisitos funcionales y no funcionales del AGC-MAC. Además, se presentan los diagramas de casos de uso y los mockups diseñados para su validación antes de la implementación. Finalmente, se plantea una funcionalidad-clínico genómica para la aplicabilidad clínica de los datos genómicos.
- **Capítulo 5:** este capítulo consiste en la extensión del Modelo de la Patología Macular necesario para conseguir una implementación consistente y capaz de explotar los datos adecuadamente. En primer lugar, se expone la primera versión del modelo elaborado en el trabajo previo. En segundo lugar, se exponen los detalles de la nueva versión. Por último, se plantea una extensión con la unión del Modelo Conceptual del Genoma para la gestión y análisis de datos genómicos.
- **Capítulo 6:** se presentan los aspectos técnicos de la solución de desarrollada, tanto de la arquitectura como cada uno de sus componentes, en los que ha utilizado tecnología actual y establecidas en la industria como *React* y *NodeJS*.
- **Capítulo 7:** en este capítulo se muestran los detalles de las configuraciones necesarias para el despliegue del prototipo AGC-MAC en los servidores de PROS-UPV.
- **Capítulo 8:** en este capítulo se presenta, en primer lugar, los resultados de una prueba de usabilidad realizada cuyo objetivo era encontrar posibles problemas no detectados en las fases de desarrollo anteriores. En segundo lugar, se muestra la evaluación de la nueva versión de la solución *software* por parte de los *skateholders* implicados como son los expertos del Centro PROS y los especialistas en oftalmología.
- **Capítulo 9:** se exponen los resultados finales con la representación visual del artefacto, así como la experiencia de desarrollo tras la finalización del TFM.
- **Capítulo 10:** se extraen las conclusiones del trabajo realizado analizando las contribuciones principales a la solución del problema y planteando futuras líneas de desarrollo.

Capítulo 2

Estado del arte

Este Trabajo de Fin de Máster se centra en el tratamiento de enfermedades de una forma más personalizada, pero en concreto de las que afectan a la mácula. Por este motivo, en este capítulo se presenta los aspectos relevantes en este contexto de enfermedades como la *degeneración macular asociada a la edad* (DMAE).

Además, se exponen otras investigaciones que guardan relación con este trabajo respecto al diseño y construcción de sistemas de información que integren datos clínicos y genómicos. También se describen brevemente otros *softwares* disponibles que solucionan los mismos problemas de investigación planteados.

2.1. Marco conceptual

A continuación, se va a exponer el estado en el que se encuentra la enfermedad de la DMAE con los últimos avances en investigación. En primer lugar, se va a proceder a la introducción de la enfermedad brevemente, con nomenclaturas y clasificaciones actualizadas respecto al conocimiento investigado en el trabajo previo. En segundo lugar, se detalla los últimos datos de prevalencia e incidencia de la DMAE en el mundo, donde en algunas regiones la situación es notablemente más grave. Seguidamente, se exponen los factores de riesgo que influyen en el desarrollo o la progresión de la enfermedad, incluyendo factores genéticos. Finalmente, se presentan los métodos y recomendaciones para la prevención, así como también los tratamientos actuales y emergentes. En cambio, otros aspectos como los síntomas y el diagnóstico ya fueron tratados en el trabajo previo [1] y no se ha considerado relevante volverlos a explicar en profundidad en este TFM.

2.1.1. Degeneración macular asociada a la edad (DMAE)

La DMAE constituye la principal razón detrás de la pérdida grave de visión en personas mayores de 55 años en países desarrollados, representando entre el 6% y el 9% de los casos de ceguera a nivel mundial. Se estima que para el año 2040, aproximadamente 288 millones de personas a nivel global se verán afectadas por la DMAE [8]. Es la responsable de la mayoría de los casos de pérdida de visión severa. En las últimas décadas ha habido avances importantes que han mejorado la calidad de los pacientes afectados. Por ejemplo, los tratamientos anti-VEGF que al menos consiguen mantener la visión en muchos pacientes. Aun así, sigue siendo una enfermedad que sigue sin una cura o tratamiento totalmente efectivo y, por tanto, tiene un largo recorrido en la investigación.

La DMAE se define particularmente por la presencia de unos puntos amarillos llamados «*drusen*» o drusas en la zona de la mácula [4]. La existencia de estas afectaciones no tiene porqué suponer la pérdida de la visión, especialmente si el tamaño de estos es pequeño (< 63 micras). En cambio, si son más grandes suponen un alto riesgo para la evolución de la enfermedad.

Existen tres clasificaciones del estado de esta enfermedad: precoz, intermedia, avanzada neovascular o avanzada no neovascular. En las etapas precoces o intermedias, los síntomas son mínimos. En cambio, las etapas avanzadas son más severas y con síntomas mucho más claros. En la Ilustración 4 se muestra la nueva clasificación de la DMAE.

Clasificación		Definición
Sin patología	Sin cambios aparentes propios del envejecimiento	Ausencia de drusas y ausencia de alteraciones pigmentarias
	Cambios normales propios del envejecimiento	Drupas (pequeñas drusas, $\leq 63 \mu\text{m}$) y ausencia de alteraciones pigmentarias relacionadas con DMAE
DMAE	DMAE precoz	Drusas medianas, $>63 \mu\text{m}$ y $\leq 125 \mu\text{m}$, y ausencia de alteraciones pigmentarias relacionadas con DMAE
	DMAE intermedia	Drusas grandes, $>125 \mu\text{m}$, y/ o cualquier alteración pigmentaria relacionada con DMAE
	DMAE avanzada	DMAE neovascular o cualquier atrofia geográfica

Ilustración 4 Nueva clasificación clínica de la DMAE (Imagen de la Sociedad Española de Retina y Vítreo) [13]

Los síntomas más comunes en la fase avanzada de la enfermedad son visión central borrosa, metamorfopsia (visión distorsionada) y micropsia o macropsia (alteración del tamaño y profundidad de un objeto). El depósito local de estas drusas es un cambio significativo asociado a la edad.

Dentro de la fase avanzada, dependiendo de la lesión se puede tratar de una DMAE neovascular o no neovascular. También se le suele llamar húmeda (neovascular) o seca (no neovascular). La DMAE no neovascular presenta una atrofia geográfica en la región foveal (centro de la mácula) que puede acabar pérdida de visión progresiva permanente. Por otra parte, la DMAE neovascular (húmeda), la más común (alrededor del 90% de las DMAE avanzadas [12]), se desarrolla por una neovascularización coroidea y acaba en hemorragia y fuga de líquido en las capas internas de la retina [7]. Cabe destacar existe una posibilidad de que la enfermedad progrese de una DMAE no neovascular a una neovascular (entre un 10% y 15% de los casos). En la Ilustración 5 se observa de forma clara los diferentes estados y tipos de la DMAE:

- a) La DMAE afecta al centro de la mácula. El recuadro destaca esta zona en un ojo normal sin alteraciones patológicas.
- b) DMAE temprana con drusen de tamaño medio (depósitos amarillentos, puntas de flecha).
- c) DMAE intermedia con drusen grandes (puntas de flecha) e hiperpigmentación (flechas).
- d) DAME no neovascular con atrofia geográfica (puntas de flecha).
- e) DMAE neovascular con hemorragia (asteriscos).

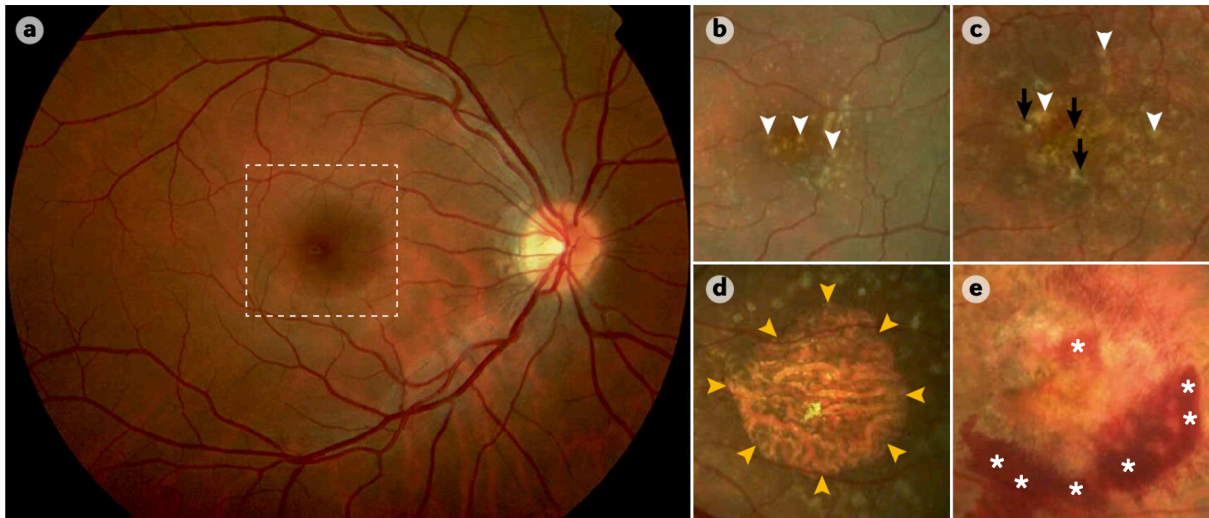


Ilustración 5 Manifestaciones de la DMAE observadas en la fotografía en color del fondo de ojo [4]

2.1.2. Prevalencia

Un metaanálisis llevado a cabo en 2014 en la población global en personas entre 45 y 85 años concluye que un 8.65% padece DMAE, en cualquiera de sus formas [8]. En concreto, la prevalencia para la enfermedad en estado inicial es del 8,01%, mientras que un 0,37% para la avanzada. Además, la prevalencia en la población europea descendente era superior, de un 12,3%. En cambio, los descendientes de la región de Asia era de un 7,4%.

Otro estudio realizado en 2020 pero en este caso, en mayores de 60 años residentes en Europa, encontró una prevalencia de un 25,3% de la DMAE en estado inicial o intermedia [20]. El número de afectados aumenta en todas las regiones junto con la edad, pero en Europa es notablemente superior al resto. Esto se debe a que la edad no es el único factor involucrado, aunque sí el principal.

2.1.3 Factores de riesgo no modificables

La edad es un factor que no puede ser controlado por razones evidentes. Es el factor que más afecta en diferencia al desarrollo de la DMAE. La edad en la que un paciente puede ser diagnosticado de esta enfermedad en estado inicial puede variar enormemente, pero normalmente a partir de los 55 años. Estudios han demostrado que la prevalencia es exponencial al incremento de la edad. Aproximadamente, la incidencia se cuadruplica por cada década. A modo de comparación, en las cataratas, por ejemplo, también encontramos muchos más pacientes en edades avanzadas.

2.1.4. Factores de riesgo del entorno

Existen una serie de factores del entorno que se han demostrado que pueden influir negativamente de diversas maneras el desarrollo y progresión de la DMAE [4]. En contra de los no modificables, estos si pueden ser controlados y, además, son factores que, si se controlan, impactan positivamente en nuestra calidad de vida, no solo en la propia enfermedad de la mácula. A continuación, se listan los factores de riesgo de entorno y cómo afectan:

- *Fumar*: fumar tabaco es un factor de riesgo independiente, incrementando el riesgo de progresión de la DMAE por dos o por cuatro. Fumar genera una serie de malfuncionamientos

que están relacionados directamente con la patogénesis de la enfermedad como por ejemplo en la presión de la sangre o en el estrés oxidativo.

- *Dieta:* seguir una dieta mediterránea se ha demostrado que está asociado con una reducción del riesgo de padecer AMD. Esta relación también se ha encontrado con la evolución de la DMAE en estado avanzada. También otros suplementos como omega3 y ácidos grasos intervienen positivamente. En la dirección opuesta también existen estudios, una alimentación alta en el ácido linoleico, las grasas monoinsaturadas, el ácido oleico o las grasas saturadas se asocian a un mayor riesgo de progresión de la DMAE.
- *Actividad física:* la actividad física continuada es otro factor que está involucrado positivamente en el desarrollo de la enfermedad, reduciendo el avance de esta. Este factor actúa como una barrera para frenar la caída de la pérdida de visión. En concreto, los estudios indican que se necesitan unas tres horas de actividad física moderada-alta para que cause impacto.

2.2.4 Factores de riesgo genéticos

Existe una fuerte evidencia de la influencia de la genética en la DMAE [4]. Esta evidencia parte de observar los familiares de sangre de los pacientes afectados por esta enfermedad. Existen diversas variaciones fuertemente asociadas a la DMAE, aunque actualmente los resultados no soportan cambiar los tratamientos en función de la genética de un paciente. Estos factores de riesgo pueden afectar negativamente, por ejemplo, aumentando el riesgo de progresión a estado avanzado. Respecto a la influencia de la genética en la DMAE, se realiza un análisis en más profundidad en el **Capítulo 3**.

2.2.5. Prevención y recomendaciones

Tras conocer los factores de riesgos implicados en la DMAE, se pueden tomar algunas acciones para prevenir los efectos adversos de la enfermedad. Especialmente si existen antecedentes familiares [4]. Se recomiendan que se realicen cambios en el estilo de vida tanto en pacientes en general como en aquellos con DMAE inicial en uno o ambos ojos. Estos cambios pueden abarcar la modificación de la dieta para incorporar alimentos que contengan antioxidantes y ácidos grasos omega-3 y omega-6 que se encuentran en el pescado, además de la gestión del peso y la cesación del tabaquismo. También se pueden controlar otros factores de riesgo como la presión arterial, los niveles de lípidos y el estado físico.

2.2.6. Tratamientos actuales

Los pacientes con DMAE en estado precoz o intermedia no presentan una necesidad subsidiaria de terapias por parte de un especialista en oftalmología. La estrategia en estos casos se basa en el diagnóstico, tratamiento y recomendaciones y seguimiento. En estas fases se centra en corregir los factores de riesgo para reducir el riesgo de progresión a una DMAE avanzada. La Ilustración 6 e Ilustración 7 muestran de una forma gráfica la estrategia de la DMAE precoz y DMAE intermedia, respectivamente.

DMAE PRECOZ

Drusas medianas ($>63 \mu\text{m}$ y $\leq 125 \mu\text{m}$) y ausencia de alteraciones pigmentarias relacionadas con DMAE

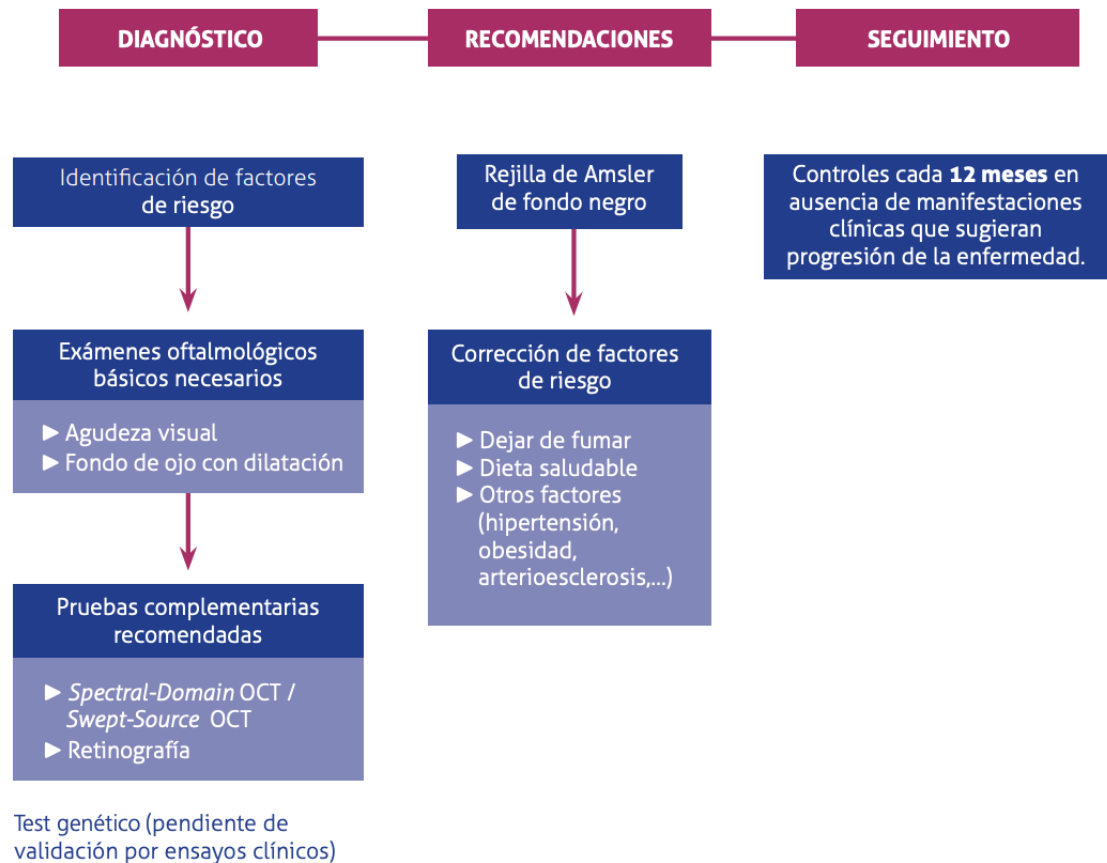


Ilustración 6 Estrategia de diagnóstico, tratamiento y seguimiento del paciente con DMAE precoz (Imagen de la Sociedad Española de Retina y Vítreo) [13]

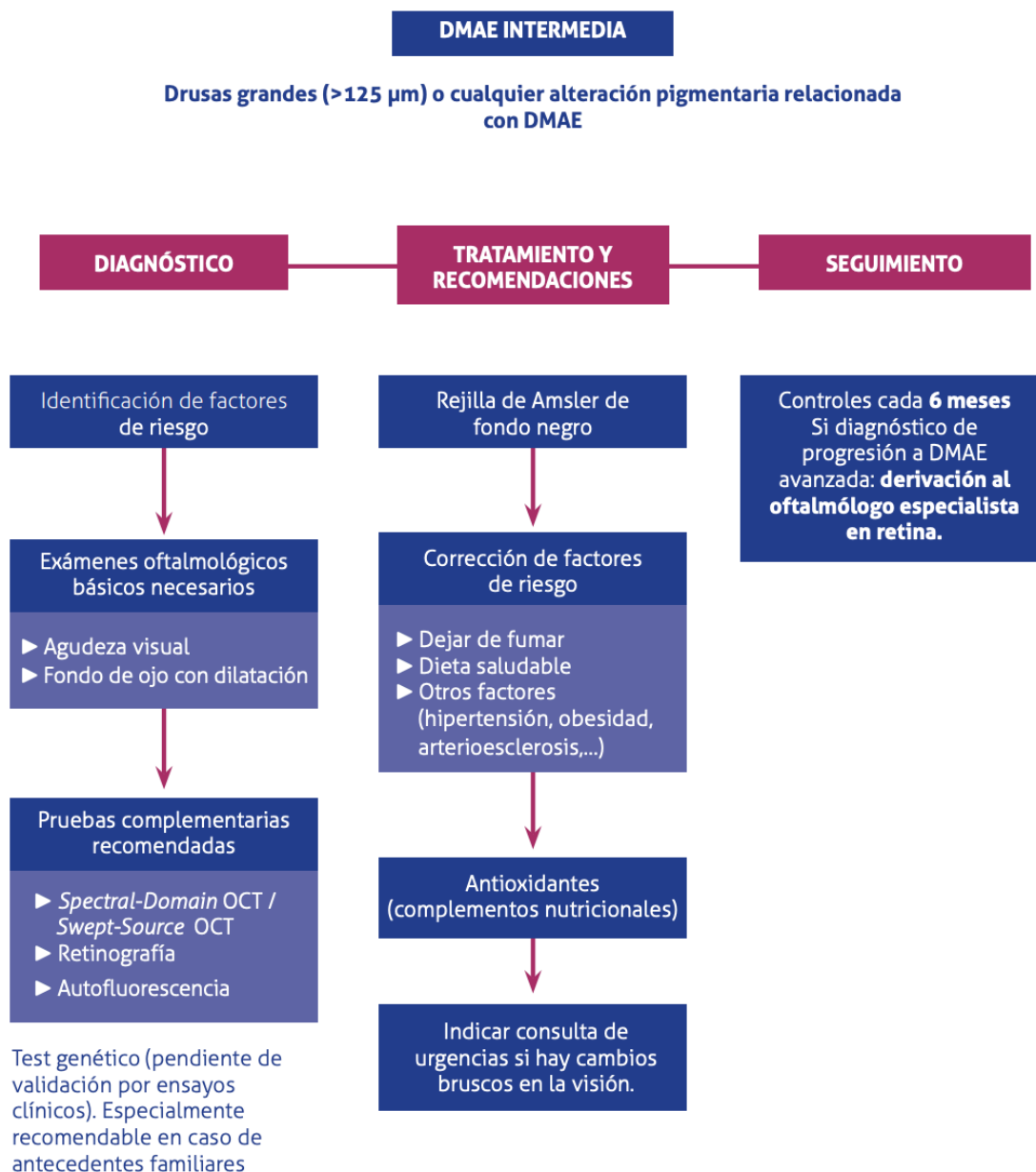


Ilustración 7 Estrategia de diagnóstico, tratamiento y seguimiento del paciente con DMAE intermedia (Imagen de la Sociedad Española de Retina y Vítreo)[13]

En cambio, en las DMAE avanzadas requieren de terapias específicas para tratar la enfermedad. Existen dos terapias más sofisticadas que han surgido en las últimas décadas y que han permitido frenar la pérdida de visión. [7]. Por desgracia, aún no existe ningún método al alcance para tratar DMAE no neovascular. Seguidamente, se detallan lista los tratamientos y se detallan brevemente en qué consisten.

- *Terapia láser:* la terapia fotodinámica (PTD) implica la administración intravenosa de un agente fotosensibilizante, llamado verteporfina, que se dispersa por todo el organismo y se concentra en la neovascularización coroidea (CNV). Después, mediante la aplicación de un láser infrarrojo de longitud de onda más extensa durante el tratamiento, se puede promover el cierre de las anomalías en la CNV. La TFD todavía se emplea en

la actualidad como un complemento en situaciones en las que los tratamientos anti-VEGF por sí solos no han tenido éxito.

- *Inyecciones intravítreas y agentes contra el factor de crecimiento endotelial vascular (anti-VEGF):* El avance de inyecciones intravítreas junto con la terapia anti-VEGF ha revolucionado el tratamiento de la DMA. Consiste en una inyección recurrente en la cavidad vítrea con fármacos que contengan alguno de los agentes deseados. Es una terapia que se puede realizar rápidamente y no supone problemas para los pacientes tras la inyección. Se ha demostrado efectiva en la preservación de la visión, aunque estos fármacos tienen un coste elevado.

2.2.7. Tratamientos emergentes

Entre todos últimos estudios y avances en la investigación destaca el uso del agente *pegcetacoplan* (ensayos DERBY y OAKS) que ha sido aprobado por la FDA (*Food and Drug Administration*) como el primer y único tratamiento para la DMAE seca [23]. Esta inyección reduce el crecimiento del área atrofiada significativamente. Se ha demostrado su efectividad a partir de los 24 meses de continuadas inyecciones y parece que su efecto es mayor si se sigue con ella. Ahora la clave está en empezar con la terapia en etapas más tempranas de la DMAE seca con el objetivo de minimizar la evolución de esta. Este acontecimiento significa poder evitar la ceguera completa de un gran número de pacientes afectados por la DMAE seca.

Aparte del uso de *pegcetacoplan*, también hay otros ensayos importantes en marcha para tratar la DMAE, también para la forma seca. Uno de ellos es la terapia génica [24], donde se intenta corregir el funcionamiento del sistema del complemento, implicado en la patogénesis de la DMAE, inyectando una copia de gene. También hay nuevos medicamentos anti-VEGF en proceso en proceso de aprobación.

2.2. Trabajos relacionados

En relación con el trabajo realizado en este TFM, se llevan a cabo numerosas investigaciones que tratan de dar solución a las enfermedades contemporáneas a través de los SI. En esta sección se van a presentar una serie de trabajos relacionados con este TFM.

En el trabajo previo del que parte este TFM [1, 3], se estudia y analiza el dominio de las enfermedades maculares, se facilita un modelo conceptual de la patología macular (MCPM) y se desarrolla un Sistema de Información para la explotación de los datos (G-MAC). El SI desarrollado ayuda a la gestión de datos clínicos de pacientes con enfermedades maculares, pero no facilita el análisis de ellos. El trabajo demuestra el uso de modelos conceptuales en el entorno clínico.

Hoy en día, se busca conseguir una medicina totalmente personalizada para tratar a los pacientes con los mejores métodos disponibles según su perfil. En este tipo de medicina, manejar el genoma humano de los pacientes es clave para tratar con precisión. Sin embargo, el entendimiento del dominio y la gestión de datos genómicos no es tarea fácil. Existen numerosos estudios en este camino, como la tesis doctoral de Reyes Román et. al. (2018) [18], que propone un Modelo Conceptual del Genoma Humano que ayuda a potenciar el conocimiento del genoma humano y que sirve como base para el desarrollo de un Sistema de Información Genómica. La extensión de este modelo la realiza García Simón et. al. (2022) [17], en su tesis doctoral, que lo actualiza para que pueda ser usado en aplicaciones del mundo real independientemente de la especie (Un MC del Genoma). También se han realizado exhaustivos estudios acerca de los componentes genéticos

de la DMAE [2] haciendo uso de *softwares* especializados en el tratamiento de datos genómicos complejos [16].

Respecto al estado de herramientas *software* especializadas para la gestión y análisis de pacientes con enfermedades maculares, solo existe uno en el mundo y es de difícil acceso. Esta aplicación se llama *Fight Retinal Blindness (FRT)* [30] y ha sido desarrollado por la Universidad de Sydney. Entre sus funcionalidades, ofrece gráficos de los pacientes, informes de resultados y formularios dinámicamente generados. FRT puede tratar pacientes con las enfermedades: DMAE húmeda, neovascularización coroidea, edema macular diabético y obstrucción de la vena retiniana.

Capítulo 3

Obtención de Variaciones Relevantes asociadas a DMAE: Uso Práctico del Oráculo

Para poder trabajar con datos genómicos, es importante que exista una evidencia fuerte que sustente su efecto en las enfermedades. Sin embargo, obtener y manejar este tipo de datos presenta dificultades de búsqueda, filtración e integración. Para ello, existen herramientas que hacen uso de los modelos conceptuales y de la ingeniería de *software* que consiguen resolver estos problemas. En este capítulo, se realiza un análisis de las variaciones relevantes implicadas en la DMAE mediante El Oráculo Genómico de Delfos para: i) obtener las últimas evidencias de variaciones implicadas en la DMAE y ii) analizar la herramienta Delfos respecto a su integración con los datos clínicos con el fin de brindar un tratamiento más personalizado.

3.1. Introducción

Como se ha explicado en la sección 2.1.1., la DMAE es la causa más común de ceguera en los países desarrollados, afectando a más de 195 millones de personas [8]. Esta enfermedad se desarrolla principalmente a partir de los 75 años y a causa del envejecimiento global de la población, se estima que el número de personas afectadas aumentará a 288 millones en 2040. Además, la prevalencia difiere significativamente entre regiones, siendo Europa la que ocupa el primer lugar con la prevalencia más alta.

El riesgo de padecer de la DMAE tiene un fuerte componente genético que puede condicionar el estilo de vida, tanto en la prevención, ya que existe una predisposición, como en el tratamiento de esta. Por ejemplo, el gen CFH está implicado en los procesos biológicos que causan el desarrollo de la enfermedad [25] y fumar influye en ellos negativamente [26]. Es en este punto donde las variaciones de los genes asociados toman un papel crucial.

En la genómica, una variación se refiere a una diferencia en la secuencia de ADN en comparación con una secuencia de referencia. Estas variaciones pueden tener diferentes efectos sobre la función génica y la expresión de genes, lo que a su vez puede estar relacionado con ciertos fenotipos. Sin embargo, en muchos casos, la relación entre las variaciones genéticas y los fenotipos es más compleja y menos clara.

Este capítulo tiene como objetivo principal identificar qué variaciones están relacionadas con el fenotipo de estudio, la DMAE. En concreto, aquellas que están soportadas con evidencia suficiente. Para ello, se hará uso de la Plataforma Delfos [16], que sigue la metodología SILE [27], y se evaluará la experiencia de esta desde la perspectiva de la ingeniería del *software*.

3.2. Materiales y métodos usados

La bioinformática es un campo interdisciplinario que se enfoca en el manejo y análisis de datos biológicos utilizando herramientas computacionales y estadísticas. La gran cantidad de datos generados ha llevado a la necesidad de integrar múltiples fuentes de información para obtener una comprensión más completa de los procesos biológicos.

Sin embargo, la heterogeneidad de los datos y la falta de una terminología estandarizada y consistente representan uno de los mayores desafíos en la integración de datos en la bioinformática. Esto se debe en gran parte a que diferentes grupos de investigación y comunidades científicas utilizan diferentes sistemas de nomenclatura y terminología para describir los mismos conceptos. El ejemplo de la nomenclatura de los fenotipos es un caso común en el que la terminología utilizada puede variar ampliamente.

Un modelo conceptual del genoma humano podría proporcionar una solución, al permitir una representación unificada y estandarizada de los datos genómicos. Tal modelo podría ser utilizado para definir términos comunes y establecer relaciones entre los datos, lo que permitiría una integración más fácil y efectiva de los datos de diferentes fuentes.

También es necesario tener una metodología clara y definida para llevar a cabo un proyecto de investigación en genómica o bioinformática. La metodología proporciona un marco estructurado para planificar y ejecutar los pasos necesarios para lograr los objetivos del proyecto, y ayuda a garantizar que los resultados sean precisos, reproducibles y confiables. Así como un sistema de información robusto y escalable que pueda manejar la gran cantidad de datos de manera efectiva.

Así pues, el método SILE, que se basa en un modelo conceptual, ha sido aplicado junto con la herramienta Delfos para alcanzar los objetivos del trabajo. A continuación, se describen con más detalle.

3.2.1. El método SILE

El método SILE [27] define un proceso sistemático a seguir para el tratamiento de datos genómicos. Este proceso consta de cuatro fases: *Search*, *Identification*, *Load* y *Exploitation*.

- 1) *Search*: Extracción de información de diferentes fuentes.
- 2) *Identification*: Identificar información relevante según el modelo conceptual que caracteriza el dominio.
- 3) *Load*: Cargar la información en un sistema de información.
- 4) *Exploitation*: Analizar y explotar la información para transformar la información en conocimiento.

3.2.2 Oráculo Genómico de Delfos

Seguir el método SILE de forma manual para encontrar las variaciones con evidencia suficiente es un trabajo costoso, especialmente si el número de fuentes es cuantioso. Una herramienta que semiautomatice este trabajo sería de gran ayuda para reducir el tiempo invertido como, por ejemplo, Delfos. Delfos [16] se compone de cuatro módulos (Hermes, Ulises, Delfos y Sibila) que implementan cada una de las fases del método SILE. Hermes se encarga de la fase de búsqueda, Ulises de la de identificación, Delfos de la de carga y Sibila de la de explotación.

- 1) *Hermes*: es un módulo en R que facilita la búsqueda de datos genómicos al automatizar la conexión, extracción, transformación e integración de datos de diferentes repositorios genómicos conocidos (ClinVar [44], Ensembl [45], GWAS y LOVD [46]).
- 2) *Ulises*: es un módulo que ayuda a identificar los datos genómicos relevantes para su uso clínico. Utiliza algoritmos para evaluar la calidad y relevancia de los datos y clasifica las variaciones genéticas según su evidencia.
- 3) *Delfos*: es un módulo que automatiza la carga las variaciones, previamente seleccionadas, en una base de datos persistente.
- 4) *Sibila*: es el módulo para analizar y explorar datos genómicos en el contexto clínico. Permite al usuario explorar las variaciones relevantes y acceder a análisis estadísticos a través de una interfaz web.

3.3. Resultados

Haciendo uso de Hermes, se han podido obtener un total de 1437 variaciones relacionadas con la DMAE. La Tabla 1 a continuación resume los resultados.

Repositorio Variaciones	
ClinVar	1327
Ensembl	116
Total diferentes	1437

Tabla 1 Variaciones obtenidas con el módulo Hermes (Mayo de 2023)

Estas variaciones proceden de las bases de datos ClinVar y Ensembl. Para realizar la búsqueda, el módulo Hermes requiere de unos términos como entrada. Se han utilizado las expresiones "age related macular degeneration"[Disease/Phenotype]" y "age-related macular degeneration" para ClinVar y Ensembl, respectivamente.

El siguiente paso a la recolección de las variaciones es pasarlas por Ulises para que sean clasificadas según su evidencia. Los resultados de esta herramienta se pueden resumir en las figuras Ilustración 8 (clasificadas por cromosoma), Ilustración 9 (clasificadas por tipo) e Ilustración 10 (clasificadas por relevancia clínica). Según esta herramienta, el 16,46% de las variaciones se consideran relevantes (Tabla 2), es decir, las que no han sido descartadas. Pero en esta fase se ha podido comprobar que Hermes ha extraído datos de fenotipos no relacionados con el de estudio, que serán filtrados en las próximas etapas.

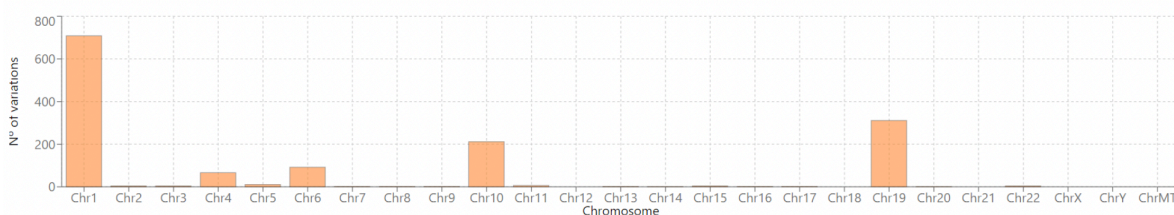


Ilustración 8 Variaciones clasificadas por cromosoma

Variations Grouped by Type

- SINGLE NUCLEOTIDE VARIANT (1389 variations)
- DELETION (21 variations)
- null (13 variations)
- DUPLICATION (6 variations)
- MICROSATELLITE (4 variations)
- Others (4 variations)

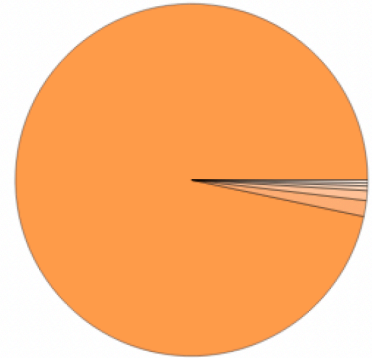


Ilustración 9 Variaciones clasificadas por tipo

Variations Grouped by Clinical Actionability

- Not disorder causing or protective effect (1980 variations)
- Uncertain role (2240 variations)
- Not provided (168 variations)
- Disorder causing or risk factor (862 variations)

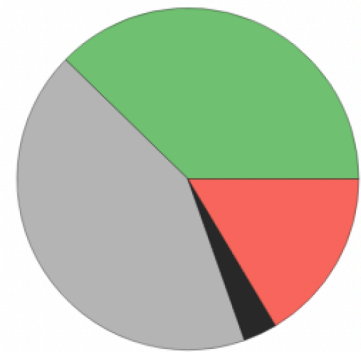


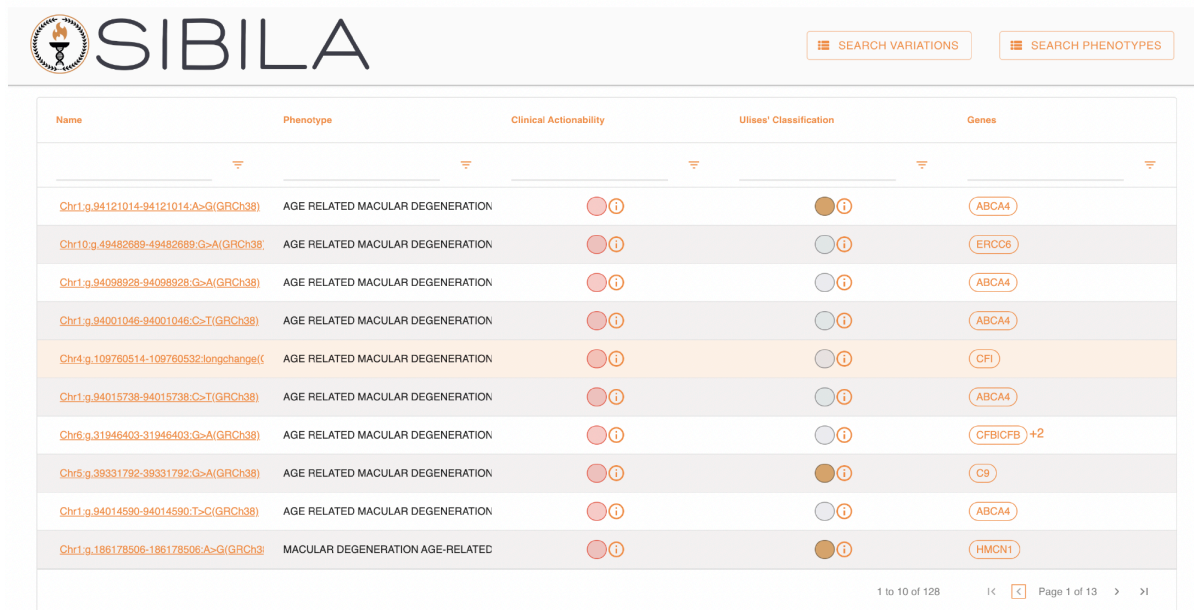
Ilustración 10 Variaciones clasificadas por relevancia clínica

Clasificación	Variaciones (repetidas por cada fenotipo)
<i>Rejected</i>	4386 (83,54%)
<i>Accepted with limited evidence</i>	257 (4,9%)
<i>Accepted with moderate evidence</i>	597 (11,37%)
<i>Accepted with strong evidence</i>	0 (0%)
<i>To follow up</i>	10 (0,19%)

Tabla 2 Variaciones clasificadas por evidencia por Ulises

En el módulo de Delfos se han cargado las variaciones clasificadas como *Accepted with limited evidence*, *Accepted with moderate evidence*, *Accepted with strong evidence* y *To follow up*. Después de este paso, las variaciones ya están listas para su explotación con el último módulo, Sibila.

Por último, con Sibila se visualizan los datos disponibles. En concreto, se ha hecho uso de la funcionalidad de búsqueda de variaciones. Gracias a esta funcionalidad, se puede obtener una tabla con todas las variaciones filtradas por los fenotipos de interés (Ilustración 11) y de este modo, eliminar el ruido para el análisis. Los fenotipos de interés son todos aquellos que incluyan «*macular degeneration*». En total se han obtenido 128 variaciones, que, frente a las 1437 iniciales, representan un 9,65%. Los genes afectados son: ABCA4, LOC1, APOE, ARMS2, C2, C3, C9, CFBICFB, CFH, CFHR3, CFHHR1, CFI, ERCC6, HMCN1, HTRA1 y MT-TL1.



The screenshot shows the Sibila web application interface. At the top left is the Sibila logo, and at the top right are two search buttons: 'SEARCH VARIATIONS' and 'SEARCH PHENOTYPES'. Below the header is a table with the following columns: Name, Phenotype, Clinical Actionability, Ulises' Classification, and Genes. The table displays 10 rows of genetic variations, all filtered by the phenotype 'AGE RELATED MACULAR DEGENERATION'. The variations are listed with their genomic coordinates and the specific mutation. The Clinical Actionability column shows red circles with an 'i' icon, and the Ulises' Classification column shows colored circles with an 'i' icon. The Genes column lists the affected genes: ABCA4, ERCC6, CFI, and C9. The table is paginated, showing '1 to 10 of 128' results on 'Page 1 of 13'.

Name	Phenotype	Clinical Actionability	Ulises' Classification	Genes
Chr1.g.94121014-94121014:A>G(GRCh38)	AGE RELATED MACULAR DEGENERATION			ABCA4
Chr10.g.49482689-49482689:G>A(GRCh38)	AGE RELATED MACULAR DEGENERATION			ERCC6
Chr1.g.94098928-94098928:G>A(GRCh38)	AGE RELATED MACULAR DEGENERATION			ABCA4
Chr1.g.94001046-94001046:C>T(GRCh38)	AGE RELATED MACULAR DEGENERATION			ABCA4
Chr4.g.109760514-109760532:longchange(f	AGE RELATED MACULAR DEGENERATION			CFI
Chr1.g.94015738-94015738:C>T(GRCh38)	AGE RELATED MACULAR DEGENERATION			ABCA4
Chr6.g.31946403-31946403:G>A(GRCh38)	AGE RELATED MACULAR DEGENERATION			CFBICFB +2
Chr5.g.39331792-39331792:G>A(GRCh38)	AGE RELATED MACULAR DEGENERATION			C9
Chr1.g.94014590-94014590:T>C(GRCh38)	AGE RELATED MACULAR DEGENERATION			ABCA4
Chr1.g.186178508-186178508:A>G(GRCh38)	MACULAR DEGENERATION AGE-RELATED			HMCN1

Ilustración 11 Variaciones del fenotipo visualizadas en Sibila

3.4. Discusión

Respecto al *software* Delfos, este ha sido de gran utilidad para aumentar la productividad en la búsqueda de variaciones de un fenotipo. Esto es posible gracias sus ventajas, como la integración de diferentes bases de datos en un mismo módulo (Hermes), que elimina la necesidad de realizar las consultas en sus correspondientes APIs. Otra ventaja es la clasificación y visualización de las variaciones de una manera rápida e interactiva, destacando las funcionalidades de Sibila.

Aunque es una excelente herramienta, durante la utilización de este ha habido algunos aspectos que han empeorado la experiencia de usuario. A continuación, se resumen los problemas principales, clasificados por módulo:

- *Hermes - Falta de documentación:* El manual de Hermes no ha sido suficiente para entender las funciones del módulo implementado en lenguaje R.
- *Hermes - Entorno de desarrollo:* Para hacer funcionar Hermes es necesaria la instalación de múltiples paquetes en el entorno RStudio. Además, la integración de las variaciones solo funciona con el sistema operativo Windows. Una posible solución sería desplegar Hermes en una aplicación web como el resto de los módulos de Delfos.

- *Ulises - Coherencia*: Ulises en primer lugar indica el número de variaciones cargadas. Este número no coincide con el que muestran algunas de sus gráficas, siendo mucho mayor, ya que cada variación se repite si aparece en diferentes fenotipos. Aunque no es incorrecto, es confuso para un nuevo usuario. También se refiere con diferentes términos al mismo concepto, por ejemplo, «Irrelevant» y «Rejected».
- *Ulises - Porcentajes clasificación*: Aunque Ulises indica el número absoluto de variaciones por cada clasificación, también es importante en términos de usabilidad conocer el porcentaje, ya que es más representativo. De hecho, los porcentajes calculados en este capítulo se han realizado manualmente.
- *Sibila - Filtrado por múltiples fenotipos*: La búsqueda de variaciones por fenotipos no ha sido satisfactoria debido al selector elegido por los desarrolladores, que no permite la selección de múltiples fenotipos en una misma iteración.
- *Sibila - Genes por grupo*: Sibila no permite conocer directamente los genes implicados en las variaciones del grupo. En cambio, se debe hacer una tarea manual ordenando las variaciones por genes.

3.5. Conclusiones

Los resultados de este trabajo de obtención de variaciones han podido generar conocimiento actualizado acerca de las variaciones asociadas a la DMAE. Se han obtenido 128 variaciones con evidencia científica y de estas, ninguna con evidencia fuerte. No haber encontrado variaciones con suficiente evidencia confirma el aún desconocimiento del rol del genoma en esta enfermedad y la necesidad de seguir investigando. Para lograr los objetivos, se ha utilizado la herramienta Delfos, que ha semiautomatizado cada una de las fases de la metodología SILE. Sin Delfos, la alternativa era realizar una revisión sistemática de la literatura, siendo mucho más costosa. En cualquier caso, se debe tener presente que solo se ha extraído información de las dos principales fuentes genómicas (ClinVar y Ensembl) y lo más probable es que no se disponga de todas las variaciones existentes en la literatura, a pesar de que las faltantes serían minoritarias.

Este conocimiento puede ser utilizado para conocer más sobre las enfermedades maculares y probar nuevos tratamientos de una manera personalizada a cada paciente. Es decir, la aplicabilidad clínica de la genómica. Para lograrlo es necesario un modelo conceptual clínico-genómicos que una los dos ámbitos para una gestión consistente y adecuada de los datos. Esta unión se realiza en el **Capítulo 5** para dar soporte a una funcionalidad que integra datos clínicos y genómicos.

Capítulo 4

Especificación de requisitos

La **especificación de requisitos** actúa como puente entre el equipo de desarrollo y el cliente para materializar las necesidades del *software* en cuestión. Esta fase es clave para captar exactamente lo que se pide y llevarlo a cabo. Si se consigue realizar una buena especificación, se pueden dedicar más recursos al desarrollo o a las pruebas, que típicamente son las fases más costosas temporalmente.

En este trabajo se han especificado los requisitos escuchando y preguntando al Dr. Rubén Cabrera por sus tareas, necesidades y problemas que se encuentra en su día a día durante el desempeño de su trabajo en el Hospital de Vega Baja de Orihuela, como también las proposiciones de funcionalidades que le serían de especial interés. Cabe decir que el Dr. Rubén Cabrera ha sido el único médico colaborador durante esta fase del desarrollo del TFM. Por ende, existe el riesgo de que la especificación de requisitos o el diseño pueda no representar fielmente al conjunto de usuarios finales.

Ya hay un trabajo previo de especificación de requisitos en el prototipo G-MAC. Por tanto, el trabajo realizado en esta fase no solo ha consistido en la creación de nuevos requisitos, sino también en la modificación y mejora de estos. De hecho, lo segundo ha sido la principal contribución por lo que respecta a requisitos.

En primer lugar, se listan los **requisitos funcionales y no funcionales** de aplicación finalmente implementada con los actores implicados. En segundo lugar, se exponen los **diagramas de casos de uso** (CU) que ayudan a entender las acciones permitidas por cada actor y la función de cada caso de uso. Seguidamente, se exponen los *mockups* diseñados. Por último, se dedica una sección separada a la especificación de funcionalidad clínico-genómica. Cabe señalar que todas estas secciones se explican en comparación al prototipo G-MAC para remarcar las mejoras introducidas.

4.1. Requisitos funcionales

Los requisitos funcionales de un *software* son las especificaciones detalladas de las acciones y funciones que el software debe realizar para satisfacer las necesidades del usuario y cumplir con sus objetivos. Estos requisitos definen cómo debe comportarse el *software* en respuesta a diversas entradas y condiciones. Por ejemplo, pueden incluir funciones como la capacidad de iniciar sesión, crear y editar perfiles de usuario, buscar y mostrar información específica, generar informes o cualquier otra acción que sea esencial para el propósito del *software*. Estos requisitos funcionales son esenciales para guiar el proceso de diseño y desarrollo del software, asegurando que el producto final cumpla con las expectativas y necesidades del usuario.

El trabajo de este TFM ha consistido en primer lugar, mejorar los requisitos ya implementados y, en segundo lugar, añadir de nuevos. A continuación, se presentan los requisitos funcionales implementados en la nueva versión, diferenciados por mejorados (naranja), nuevos (verde) en la Tabla 3. En el **Anexo I – Prototipo G-MAC** se pueden consultar capturas del prototipo inicial que ilustran el punto de partida de este TFM.

Se han añadido un total de cinco nuevos requisitos funcionales a la plataforma *software*. En la sección **Mockups** se mostrarán visualmente los diseños de mejoras y modificaciones aplicadas.

Entre estas mejoras cabe destacar el requisito funcional de la gestión de los médicos, que se ha mejorado para que un administrador pueda editar un médico, así como promocionar (designar otro administrador) o degradar los permisos de un usuario (quitar permisos de administrador). También puede cambiar la contraseña de un usuario en caso de que este la olvide.

Otro requisito aún más importante para este TFM es el **RF12**, que permite visualizar gráficas para conocer la población de los pacientes del médico. Esto es posible con la extensión del MCPM a la agregación de datos. Esta funcionalidad incluye un total de siete gráficas combinando las variables enfermedad, edad, sexo y antecedente médico. En concreto, el médico puede descubrir lo siguiente acerca de sus pacientes:

- Número total de pacientes
- Pacientes clasificados por enfermedad
- Pacientes clasificados por antecedente médico
- Pacientes clasificados por edad
- Pacientes clasificados por sexo
- Pacientes clasificados por enfermedad y antecedente médico
- Pacientes clasificados por enfermedad y edad
- Pacientes clasificados por enfermedad y sexo

Cabe señalar que solo se toman en cuenta los pacientes que tengan al menos una enfermedad en activo. Es decir, que tengan alguna enfermedad y no hayan sido dados de alta de ella. Esta funcionalidad es clave para mejorar el análisis de los datos clínicos, ya que aporta nuevo conocimiento a los médicos acerca de sus pacientes a través de la agregación de datos.

Respecto a los actores, en la versión inicial el actor Médico era el Admin (administrador) actual, que es el usuario que tiene acceso a todas las funcionalidades que ofrece el *software*. Puede gestionar los hospitales y los médicos del sistema, así como tener acceso a cualquier paciente registrado. En cambio, el médico tiene solamente acceso a funcionalidades asociadas a los pacientes a su cargo. Esto se verá con más detalle en la sección **Diagrama de casos de uso**.

Nº	Requisito funcional	Actor	
		Admin	Médico
RF1	El sistema permitirá la consulta y creación de nuevos hospitales y gestión de médicos que formen parte del sistema u organización.	✓	
RF2	El sistema permitirá realizar una búsqueda por pacientes a través de su SIP	✓	✓
RF3	El sistema permitirá la gestión de pacientes y sus datos personales (consulta, creación, modificación)	✓	✓
RF4	El sistema permitirá gestionar información que pueda influir en el diagnóstico del paciente, tales como: datos antropométricos, enfermedades y/o antecedentes médicos.	✓	✓
RF5	El sistema permitirá gestionar las revisiones realizadas a un paciente (consulta, creación, modificación y eliminación)	✓	✓
RF6	El sistema ofrecerá la funcionalidad de visualizar gráficas que ayuden a seguir la evolución del paciente	✓	✓
RF7	La aplicación permitirá a los usuarios el inicio y el cierre de sesión.	✓	✓
RF8	El sistema facilitará la habilitación e inhabilitación de médicos y pacientes.	✓	
RF9	El usuario podrá gestionar los datos personales de su perfil (e.g., nombre, correo electrónico, teléfono, etc.).	✓	✓
RF10	El sistema debe permitir la creación de datos históricos.	✓	✓
RF11	El sistema permitirá que varios médicos puedan tratar un mismo paciente. También en diferentes hospitales.	✓	✓
RF12	El sistema ofrecerá la funcionalidad de visualizar gráficas que ayuden a conocer la población de los pacientes del médico. Agrupados por edad, sexo, enfermedades y antecedentes.	✓	✓
RF13	El sistema ofrecerá la funcionalidad de visualizar la información más importante de un paciente de forma resumida.	✓	✓
RF14	El sistema permitirá la modificación de los datos personales de los médicos y la promoción o degradación de los roles (Admin o Médico).	✓	

Tabla 3 Requisitos funcionales

4.2. Requisitos no funcionales

Los requisitos no funcionales de un *software* se centran en aspectos como rendimiento, seguridad, escalabilidad y usabilidad, que influyen en la calidad y la experiencia del usuario, más allá de las funciones principales del sistema. Estos requisitos definen estándares de calidad y comportamiento para asegurar un rendimiento óptimo y satisfacción del usuario. Seguidamente, se exponen los requisitos no funcionales diferenciados por nuevos (verde) y previos (blanco) en la **¡Error! No se encuentra el origen de la referencia.**

Excepto el RNF1, los nuevos requisitos no funcionales (Tabla 4) surgen de analizar los resultados del análisis del prototipo G-MAC, que reportaron una serie de problemas de usabilidad:

- Para gestionar los datos del paciente como las revisiones y enfermedades, el flujo del usuario pasaba por múltiples ventanas. Esto resultaba en la pérdida de control del usuario y problemas con la memoria a corto plazo (la información necesaria no estaba en pantalla).

- Algunos campos o las elecciones de su diseño no aportaban ningún valor al médico. Por ejemplo, se mostraba la fecha de nacimiento del paciente en lugar de la edad, que es lo que realmente quiere conocer el médico.
- Alguna información no estaba bien estructurada o no era esencial. Por ejemplo, la lista de las revisiones en formato tabla dificultaba entender el historial de un paciente. Y en algunos casos, tampoco estaba claro que era lo que se mostraba exactamente.
- Algunos formularios no permitían finalizarlos sin completar campos que el médico puede no tener disponibles en ese momento o bien, no existen.

Nº	Requisito no funcional	Categoría
RNF1	Los nombres de los pacientes deben estar anonimizados.	Protección de datos
RNF2	El usuario debe navegar lo menos posible para que no se pierda durante su uso.	Usabilidad
RNF3	Los campos de los datos visualizados en la aplicación deben ser representativos y relevantes para el usuario.	Usabilidad
RNF4	La aplicación permitirá la consecución de las acciones, aunque no se dispongan de todos los datos necesarios, de modo de que no bloquee al usuario.	Usabilidad
RNF5	El sistema proporcionará feedback al usuario durante el uso de la aplicación, informando de errores o confirmando acciones realizadas.	Usabilidad
RNF6	La información debe estar estructurada correctamente y con elementos gráficos que ayuden a su comprensión.	Usabilidad
RNF7	El sistema proporcionará feedback al usuario durante el uso de la aplicación, informando de errores o confirmando acciones realizadas.	Usabilidad
RNF8	El código de la aplicación debe estar bien estructurado y explicado, utilizando patrones arquitectónicos, de modo que sea fácil corregir o añadir funcionalidades en un futuro.	Mantenimiento
RNF9	La aplicación no debe bloquearse en ningún momento mientras se esté utilizando.	Usabilidad
RNF9	La aplicación debe proveer <i>feedback</i> con información de errores suficiente para la recuperación de ellos.	Usabilidad
RNF10	La aplicación debe ser robusta frente a posibles datos inválidos por parte del cliente.	Consistencia

Tabla 4 Requisitos no funcionales del AGC-MAC

4.3. Diagramas de casos de uso

Los **diagramas de casos de uso UML [28]** son representaciones visuales que muestran cómo los usuarios interactúan con un sistema. Los usuarios se llaman actores, y las acciones que realizan se denominan casos de uso. Estos diagramas son útiles para comprender cómo el sistema satisface las necesidades de los usuarios de manera clara y concisa.

Para lograr una definición clara de los **Casos de Uso** (CU), se han elaborado una serie de tablas que incluyen los siguientes campos:

- 1) Nombre del Caso de Uso.
- 2) Identificador representativo.
- 3) Descripción del propósito.
- 4) Usuario o entidad que realiza la acción (actor).
- 5) Requisito previo necesario para llevar a cabo la acción (precondición)
- 6) Pasos a seguir para su realización (flujo principal) y el resultado resultante de su ejecución (postcondición).
- 7) Estado del sistema al concluir la ejecución.

En primer lugar, el diagrama representado en la Ilustración 12 incluye el inicio de sesión a la aplicación, gestión del perfil del usuario y la gestión de los hospitales y médicos. De estas funcionalidades, la gestión de hospitales y médicos solo es accesible por un administrador. El administrador realiza las labores de mantenimiento y registro de nuevos hospitales y médicos al sistema. Entre las posibilidades de la gestión de los médicos, también puede inhabilitar o habilitar a médicos.

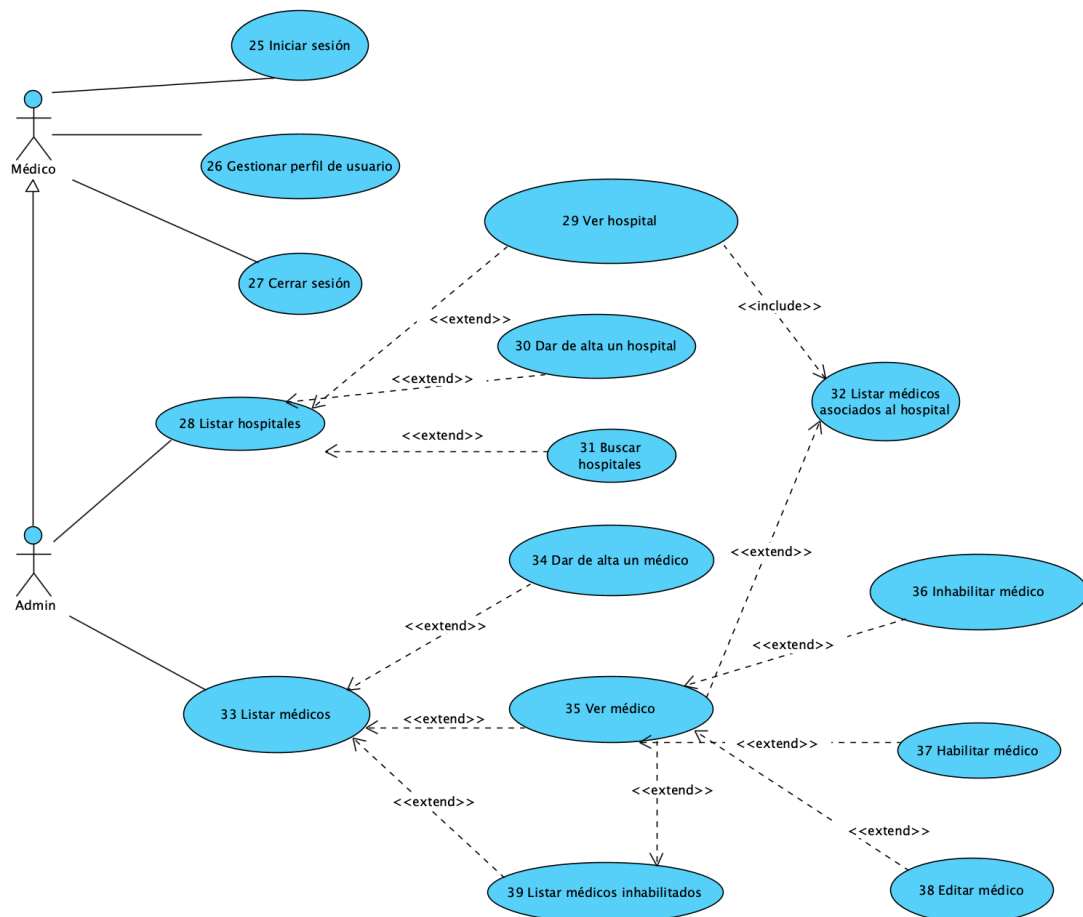


Ilustración 12 Diagrama de CU - Acceso a la aplicación y gestión de médicos y hospitales

CU	Listar hospitales	28
Descripción	Mostrar en un listado los hospitales que están dados de alta en el sistema.	
Actor	Admin	
Pre-condición	<ul style="list-style-type: none"> • Haber iniciado sesión. 	
Flujo básico	1) El usuario hace clic sobre «Hospitales» en el menú de la barra lateral.	
Post-condición	<ul style="list-style-type: none"> • Se muestra el listado de hospitales. 	

Tabla 5 CU Listar hospitales

CU	Buscar hospitales	31
Descripción	Buscar hospitales dados de alta en el sistema por nombre.	
Actor	Admin	
Pre-condición	<ul style="list-style-type: none"> • Haber iniciado sesión. 	
Flujo básico	1) El usuario hace clic sobre «Hospitales» en el menú de la barra lateral.	
Post-condición	<ul style="list-style-type: none"> • Se muestra el listado de hospitales. 	

Tabla 6 CU Buscar hospitales

CU	Ver hospital	29
Descripción	Consultar la información de un hospital.	
Actor	Admin	
Pre-condición	<ul style="list-style-type: none"> • Haber iniciado sesión. 	
Flujo básico	<ol style="list-style-type: none"> 1) El usuario hace clic sobre «Hospitales» en el menú de la barra lateral. 2) Hace clic en uno de los hospitales del listado. 	
Post-condición	<ul style="list-style-type: none"> • Se muestra la información completa del hospital. 	

Tabla 7 CU Ver hospital

CU	Listar médicos asociados al hospital	32
Descripción	Consultar los médicos asociados a un hospital.	
Actor	Admin	
Pre-condición	<ul style="list-style-type: none"> • Haber iniciado sesión. • Hay al menos un hospital registrado en el sistema. 	
Flujo básico	<ol style="list-style-type: none"> 1) El usuario hace clic sobre «Hospitales» en el menú de la barra lateral. 2) Hace clic en uno de los hospitales del listado. 	
Post-condición	<ul style="list-style-type: none"> • Se muestran el listado de médicos asociados al hospital junto con el resto de la información del hospital. 	

Tabla 8 CU Listar médicos asociados al hospital

CU	Dar de alta un hospital	30
Descripción	Dar de alta a un nuevo hospital en el sistema.	
Actor	Admin	
Pre-condición	<ul style="list-style-type: none"> • Haber iniciado sesión. 	
Flujo básico	<ol style="list-style-type: none"> 1) El usuario hace clic sobre «Hospitales» en el menú de la barra lateral. 2) Hace clic en el botón «Nuevo hospital». 3) Introduce todos los datos requeridos en el formulario que ha aparecido como una ventana modal. 4) Hacer clic en el botón «Crear», al final del formulario. 	
Post-condición	<ul style="list-style-type: none"> • Se muestra un mensaje informando del éxito o no de la operación realizada. • La lista de hospitales se actualiza con este nuevo paciente a la lista. 	

Tabla 9 CU Dar de alta un hospital

CU	Listar médicos	33
Descripción	Mostrar en un listado los médicos que están dados de alta en el sistema.	
Actor	Admin	
Pre-condición	<ul style="list-style-type: none"> • Haber iniciado sesión. 	
Flujo básico	<ol style="list-style-type: none"> 1) El usuario hace clic sobre «Médicos» en el menú de la barra lateral. 	
Post-condición	<ul style="list-style-type: none"> • Se muestra el listado de médicos. 	

Tabla 10 CU Listar médicos

CU	Listar médicos inhabilitados	39
Descripción	Mostrar en un listado los médicos que están dados de baja (inhabilitados) en el sistema.	
Actor	Admin	
Pre-condición	<ul style="list-style-type: none"> • Haber iniciado sesión. 	
Flujo básico	<ol style="list-style-type: none"> 1) El usuario hace clic sobre «Médicos» en el menú de la barra lateral. 2) Hace clic en el enlace «Pacientes inhabilitados» 	
Post-condición	<ul style="list-style-type: none"> • Se muestra el listado de médicos inhabilitados. 	

Tabla 11 CU Listar médicos inhabilitados

CU	Dar de alta un médico	34
Descripción	Dar de alta un nuevo médico en el sistema.	
Actor	Admin	
Pre-condición	<ul style="list-style-type: none"> • Haber iniciado sesión. 	
Flujo básico	<ol style="list-style-type: none"> 1) El usuario hace clic sobre «Médicos» en el menú de la barra lateral. 2) Hace clic en el botón «Nuevo médico» 3) Introduce todos los datos requeridos en el formulario que ha aparecido como una ventana modal. 4) Hacer clic en el botón «Crear», al final del formulario. 	
Post-condición	<ul style="list-style-type: none"> • Se muestra un mensaje informando del éxito o no de la operación realizada. • Se actualiza el listado de médicos. 	

Tabla 12 CU Dar de alta un médico

CU	Ver médico	35
Descripción	Consulta la información de un médico.	
Actor	Admin	
Pre-condición	<ul style="list-style-type: none"> • Haber iniciado sesión. • Hay al menos un médico registrado en el sistema y no está inhabilitado. 	
Flujo básico	<ol style="list-style-type: none"> 1) El usuario hace clic sobre «Médicos» en el menú de la barra lateral. 2) Hace clic en uno de los médicos del listado. <p>Flujo alternativo 1:</p> <ol style="list-style-type: none"> 1) El usuario hace clic sobre «Hospitales» en el menú de la barra lateral. 2) Busca o hace clic en uno de los hospitales del listado. 3) Hace clic en uno de los médicos del listado de médicos asociados al hospital. <p>Flujo alternativo 2:</p> <ol style="list-style-type: none"> 1) El usuario hace clic sobre «Médicos» en el menú de la barra lateral. 2) Hace clic en el enlace «Médicos inhabilitados». 3) Hace clic en uno de los médicos del listado de médicos inhabilitados. 	
Post-condición	<ul style="list-style-type: none"> • Se muestra la información completa de un médico. 	

Tabla 13 CU Ver médico

CU	Inhabilitar médico	36
Descripción	Inhabilita un médico del uso del sistema.	
Actor	Admin	
Pre-condición	<ul style="list-style-type: none"> • Estar en la ventana de detalle de un médico (CU Ver médico) • El médico consultado está habilitado 	
Flujo básico	<ol style="list-style-type: none"> 1) El usuario hace clic en el botón «Inhabilitar». 2) Pulsa el botón de confirmación para dar de baja al médico. 	
Post-condición	<ul style="list-style-type: none"> • Se muestra un mensaje informando del éxito o no de la operación. • Se actualiza la información del médico. 	

Tabla 14 CU Inhabilitar médico

CU	Habilitar médico	37
Descripción	Habilita un médico del uso del sistema.	
Actor	Admin	
Pre-condición	<ul style="list-style-type: none"> • Estar en la ventana de detalle de un médico (CU Ver médico) • El médico consultado está inhabilitado 	
Flujo básico	1) El usuario hace clic en el botón «Habilitar».	
Post-condición	<ul style="list-style-type: none"> • Se muestra un mensaje informando del éxito o no de la operación. • Se actualiza la información del médico. 	

Tabla 15 CU Habilitar médico


CU	Editar médico	38
Descripción	Edita la información de un médico.	
Actor	Admin	
Pre-condición	<ul style="list-style-type: none"> • Estar en la ventana de detalle de un médico (CU Ver médico) 	
Flujo básico	<ol style="list-style-type: none"> 1) Pulsa en el botón  (Editar) 2) Introducir los nuevos datos del médico en el formulario que aparece como una ventana modal. 3) Pulsar en el botón «Guardar» 	
Post-condición	<ul style="list-style-type: none"> • Se muestra un mensaje informando del éxito o no de la operación. • Se actualiza la información del médico. 	

Tabla 16 CU Editar médico

En segundo lugar, el diagrama representado en la Ilustración 13 incluye las funcionalidades principales del *software* que son las relativas a la gestión de los pacientes con enfermedades maculares. De los casos de uso representados, el administrador es el único actor que puede habilitar o inhabilitar a un paciente. El médico está limitado a la gestión de sus pacientes al cargo, mientras que el administrador tiene una lista completa de todos los pacientes del sistema.

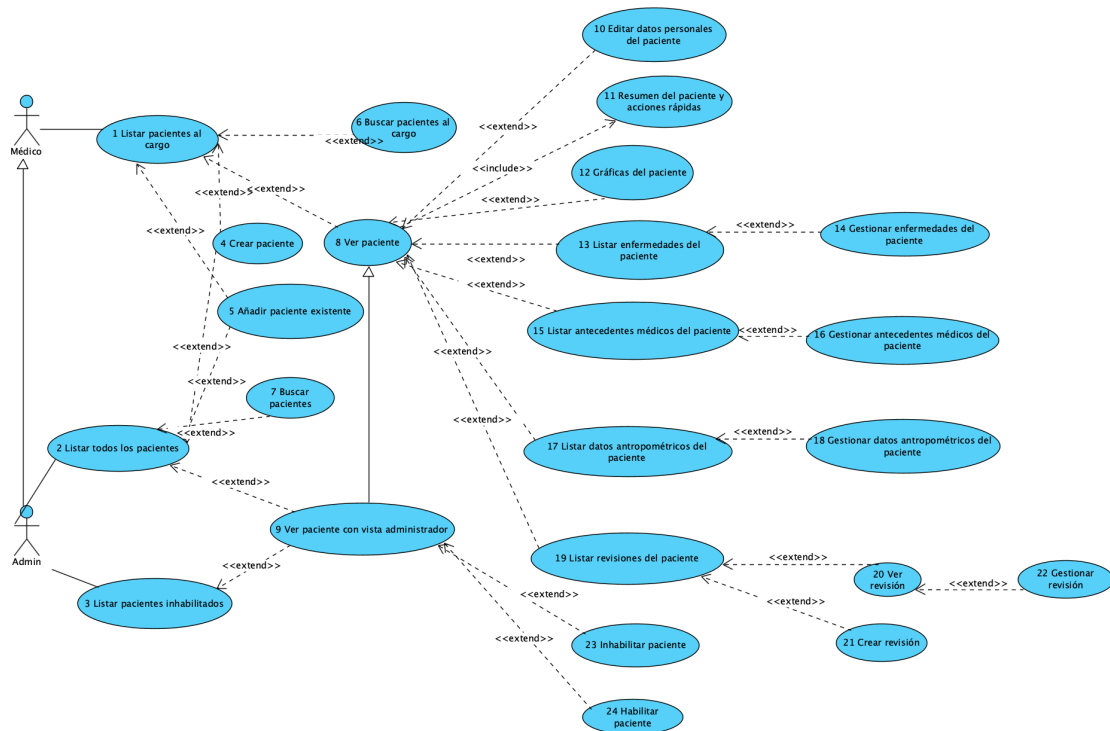


Ilustración 13 Diagrama de CU - Gestión de pacientes

CU	Listar pacientes al cargo	1
Descripción	Mostrar en un listado los pacientes que están asociados al médico y dados de alta. Por cada paciente muestra: iniciales, SIP y edad.	
Actor	Médico, Admin	
Pre-condición	<ul style="list-style-type: none"> Haber iniciado sesión. 	
Flujo básico	1) El usuario hace clic sobre «Pacientes» en el menú de la barra lateral.	
Post-condición	<ul style="list-style-type: none"> Se muestra el listado de pacientes al cargo. 	

Tabla 17 CU Listar pacientes al cargo

CU	Buscar pacientes al cargo	6
Descripción	Buscar un paciente o conjunto de pacientes habilitados al cargo del médico por SIP.	
Actor	Médico	
Pre-condición	<ul style="list-style-type: none"> Haber iniciado sesión. 	
Flujo básico	1) El usuario hace clic sobre «Pacientes» en el menú de la barra lateral. 2) En el campo de texto bajo la etiqueta Buscar pacientes introduce un conjunto de números para buscar pacientes por SIP.	
Post-condición	<ul style="list-style-type: none"> La lista de pacientes al cargo se actualiza con los pacientes cuyo SIP contenga los números introducidos en la búsqueda y estén datos de alta (habilitados) en el sistema. 	

Tabla 18 CU Buscar pacientes al cargo

CU	Crear paciente	4
Descripción	Dar de alta a un nuevo paciente en el sistema.	
Actor	Médico, Admin	
Pre-condición	<ul style="list-style-type: none"> • Haber iniciado sesión. 	
Flujo básico	<ol style="list-style-type: none"> 1) El usuario hace clic sobre «Pacientes» en el menú de la barra lateral. 2) Hace clic en el botón «Nuevo paciente». 3) Introduce todos los datos requeridos en el formulario que ha aparecido como una ventana modal. 4) Hacer clic en el botón «Crear», al final del formulario. 	
Post-condición	<ul style="list-style-type: none"> • Se muestra un mensaje informando del éxito o no de la operación realizada. • La lista de pacientes se actualiza con este nuevo paciente a la lista. 	

Tabla 19 CU Crear paciente

CU	Añadir paciente existente	5
Descripción	Añadir un paciente ya existente en el sistema a la lista de pacientes asociados al médico.	
Actor	Médico, Admin	
Pre-condición	<ul style="list-style-type: none"> • Haber iniciado sesión. 	
Flujo básico	<ol style="list-style-type: none"> 1) El usuario hace clic sobre «Pacientes» en el menú de la barra lateral. 2) Hace clic en el botón «Añadir paciente existente». 3) Introducir el SIP del paciente a añadir. 4) Hacer clic en el botón «Añadir», al final del formulario. 	
Post-condición	<ul style="list-style-type: none"> • Se muestra un mensaje informando del éxito o no de la operación realizada. • La lista de pacientes se actualiza con este nuevo paciente a la lista. 	

Tabla 20 CU Añadir paciente existente

CU	Ver paciente	8
Descripción	Consultar la información de un paciente.	
Actor	Médico	
Pre-condición	<ul style="list-style-type: none"> • Haber iniciado sesión. • Hay al menos un paciente registrado al cargo del médico y no está inhabilitado. 	
Flujo básico	<ol style="list-style-type: none"> 1) El usuario hace clic sobre «Pacientes» en el menú de la barra lateral. 2) Hace clic en uno de los pacientes del listado. 	
Post-condición	<ul style="list-style-type: none"> • Se muestra una página con los datos personales del paciente y pestañas para navegar entre toda la información. 	

Tabla 21 CU Ver paciente

CU	Resumen del paciente y acciones rápidas	11
Descripción	Mostrar un resumen del paciente (Datos antropométricos, antecedentes médicos y enfermedades) y realizar acciones rápidas: crear dato antropométrico, antecedente médico y enfermedad.	
Actor	Médico, Admin	
Pre-condición	<ul style="list-style-type: none"> • Haber iniciado sesión. • Hay al menos un paciente registrado al cargo del médico y no está inhabilitado. 	
Flujo básico	<ol style="list-style-type: none"> 1) El usuario hace clic sobre «Pacientes» en el menú de la barra lateral. 2) Hace clic en uno de los pacientes del listado. 3) Seguidamente, puede pulsar: <ol style="list-style-type: none"> a) El botón en la sección de «Datos antropométricos» para crear un nuevo dato antropométrico del paciente. b) El botón en la sección de «Enfermedades» para diagnosticar una nueva enfermedad al paciente. c) El botón en la sección de «Antecedentes médicos» médicos para crear un nuevo antecedente médico del paciente. 	
Post-condición	<ul style="list-style-type: none"> • Se muestra bajo la pestaña «Resumen», por defecto seleccionada, un resumen de los datos antropométricos (último dato), antecedentes médicos y enfermedades. • Se actualiza el resumen con la nuevos datos introducidos. 	

Tabla 22 CU Resumen del paciente y acciones rápidas

CU	Editar datos personales del paciente	10
Descripción	Modificar los datos personales de un paciente.	
Actor	Médico, Admin	
Pre-condición	<ul style="list-style-type: none"> • Haber iniciado sesión. • Hay al menos un paciente registrado al cargo del médico y no está inhabilitado. 	
Flujo básico	<ol style="list-style-type: none"> 1) El usuario hace clic sobre «Pacientes» en el menú de la barra lateral. 2) Hace clic en uno de los pacientes del listado. 3) Pulsa en el botón (Editar) 4) Introducir los nuevos datos personales en el formulario que aparece como una ventana modal. 5) Pulsar en el botón «Guardar» 	
Post-condición	<ul style="list-style-type: none"> • Se muestra un mensaje informando del éxito o no de la operación realizada. • Se actualiza la página del paciente con los nuevos datos. 	

Tabla 23 CU Editar datos personales del paciente

CU	Gráficas del paciente	12
Descripción	Mostrar la gráfica de la evolución de la agudeza visual de un paciente en el tiempo. También muestra los medicamentos que se han aplicado, en la misma gráfica.	
Actor	Médico, Admin	
Pre-condición	<ul style="list-style-type: none"> • Haber iniciado sesión. • Hay al menos un paciente registrado al cargo del médico y no está inhabilitado. 	
Flujo básico	<ol style="list-style-type: none"> 1) El usuario hace clic sobre «Pacientes» en el menú de la barra lateral. 2) Hace clic en uno de los pacientes del listado. 3) Pulsa en la pestaña «Gráficas» 	
Post-condición	<ul style="list-style-type: none"> • Se muestra la gráfica de la evolución del paciente 	

Tabla 24 CU Gráficas del paciente

CU	Listar enfermedades del paciente	13
Descripción	Mostrar un listado de las enfermedades que padece un paciente.	
Actor	Médico, Admin	
Pre-condición	<ul style="list-style-type: none"> • Haber iniciado sesión. • Hay al menos un paciente registrado al cargo del médico y no está inhabilitado. 	
Flujo básico	<ol style="list-style-type: none"> 1) El usuario hace clic sobre «Pacientes» en el menú de la barra lateral. 2) Hace clic en uno de los pacientes del listado. 3) Pulsa en la pestaña «Enfermedades» 	
Post-condición	<ul style="list-style-type: none"> • Se muestra el listado de enfermedades del paciente con toda la información acerca de ella. 	

Tabla 25 CU Listar enfermedades del paciente



CU	Gestionar enfermedades del paciente	14
Descripción	Gestionar la enfermedad de un paciente: crear, editar y eliminar.	
Actor	Médico, Admin	
Pre-condición	<ul style="list-style-type: none"> • Haber iniciado sesión. • Hay al menos un paciente registrado al cargo del médico y no está inhabilitado. 	
Flujo básico	<ol style="list-style-type: none"> 1) El usuario hace clic sobre «Pacientes» en el menú de la barra lateral. 2) Hace clic en uno de los pacientes del listado. 3) Pulsa en la pestaña «Enfermedades» 4) Seguidamente, puede pulsar: <ol style="list-style-type: none"> a) «Nueva enfermedad» para diagnosticar una enfermedad al paciente. b) El botón  de una enfermedad del listado para editar sus datos. c) El botón  de una enfermedad del listado para eliminarla. Finalmente, pulsa el botón de confirmación para eliminarla. 	
Post-condición	<ul style="list-style-type: none"> • Se muestra un mensaje informando del éxito o no de la operación realizada. • Se actualiza el listado de enfermedades. 	

Tabla 26 CU Gestionar enfermedades del paciente

CU	Listar antecedentes médicos del paciente	15
Descripción	Mostrar un listado de los antecedentes médicos de un paciente.	
Actor	Médico, Admin	
Pre-condición	<ul style="list-style-type: none"> • Haber iniciado sesión. • Hay al menos un paciente registrado al cargo del médico y no está inhabilitado. 	
Flujo básico	<ol style="list-style-type: none"> 1) El usuario hace clic sobre «Pacientes» en el menú de la barra lateral. 2) Hace clic en uno de los pacientes del listado. 3) Pulsa en la pestaña «Antecedentes médicos» 	
Post-condición	<ul style="list-style-type: none"> • Se muestra el listado de antecedentes médico del paciente con toda la información acerca de ellos. 	

Tabla 27 CU Listar antecedentes médicos del paciente



CU	Gestionar antecedentes médicos del paciente	16
Descripción	Gestionar un antecedente médico del paciente: crear, editar y eliminar.	
Actor	Médico, Admin	
Pre-condición	<ul style="list-style-type: none"> • Haber iniciado sesión. • Hay al menos un paciente registrado al cargo del médico y no está inhabilitado. 	
Flujo básico	<ol style="list-style-type: none"> 1) El usuario hace clic sobre «Pacientes» en el menú de la barra lateral. 2) Hace clic en uno de los pacientes del listado. 3) Pulsa en la pestaña «Antecedentes médicos» 4) Seguidamente, puede pulsar: <ol style="list-style-type: none"> a) «Nuevo antecedente» para añadir un nuevo antecedente médico al paciente. b) El botón  de un antecedente del listado para editar sus datos. c) El botón  de un antecedente del listado para eliminarlo. Finalmente, pulsa el botón de confirmación para eliminarlo. 	
Post-condición	<ul style="list-style-type: none"> • Se muestra un mensaje informando del éxito o no de la operación realizada. • Se actualiza el listado de antecedentes médicos. 	

Tabla 28 CU Gestionar antecedentes médicos del paciente

CU	Listar datos antropométricos del paciente	17
Descripción	Mostrar un listado de los datos antropométricos de un paciente.	
Actor	Médico, Admin	
Pre-condición	<ul style="list-style-type: none"> • Haber iniciado sesión. • Hay al menos un paciente registrado al cargo del médico y no está inhabilitado. 	
Flujo básico	<ol style="list-style-type: none"> 1) El usuario hace clic sobre «Pacientes» en el menú de la barra lateral. 2) Hace clic en uno de los pacientes del listado. 3) Pulsa en la pestaña «Datos antropométricos» 	
Post-condición	<ul style="list-style-type: none"> • Se muestra el listado del historial de los datos antropométricos del paciente con toda la información acerca de ellos. 	

Tabla 29 CU Listar datos antropométricos del paciente



CU	Gestionar datos antropométricos del paciente	18
Descripción	Gestionar un dato antropométrico del paciente: crear, editar y eliminar.	
Actor	Médico, Admin	
Pre-condición	<ul style="list-style-type: none"> • Haber iniciado sesión. • Hay al menos un paciente registrado al cargo del médico y no está inhabilitado. 	
Flujo básico	<ol style="list-style-type: none"> 1) El usuario hace clic sobre «Pacientes» en el menú de la barra lateral. 2) Hace clic en uno de los pacientes del listado. 3) Pulsa en la pestaña «Datos antropométricos» 4) Seguidamente, puede pulsar: <ol style="list-style-type: none"> a) «Nuevo dato» para registrar un nuevo dato antropométrico al paciente. b) El botón  de un dato antropométrico del listado para editar sus datos. c) El botón  de un dato antropométrico del listado para eliminarlo. Finalmente, pulsa el botón de confirmación para eliminarlo. 	
Post-condición	<ul style="list-style-type: none"> • Se muestra un mensaje informando del éxito o no de la operación realizada. • Se actualiza el listado de datos antropométricos. 	

Tabla 30 CU Gestionar datos antropométricos del paciente

CU	Listar revisiones del paciente	19
Descripción	Mostrar un listado de las revisiones de un paciente.	
Actor	Médico, Admin	
Pre-condición	<ul style="list-style-type: none"> • Haber iniciado sesión. • Hay al menos un paciente registrado al cargo del médico y no está inhabilitado. 	
Flujo básico	<ol style="list-style-type: none"> 1) El usuario hace clic sobre «Pacientes» en el menú de la barra lateral. 2) Hace clic en uno de los pacientes del listado. 3) Pulsa en la pestaña «Revisiones» 	
Post-condición	<ul style="list-style-type: none"> • Se muestra el historial de revisiones del paciente separadas por ojo izquierdo y derecho, en forma de línea temporal. 	

Tabla 31 CU Listar revisiones del paciente

CU	Ver revisión	20
Descripción	Consultar la información de una revisión de un paciente.	
Actor	Médico, Admin	
Pre-condición	<ul style="list-style-type: none"> • Haber iniciado sesión. • Hay al menos un paciente registrado al cargo del médico con al menos una revisión registrada y no está inhabilitado. 	
Flujo básico	<ol style="list-style-type: none"> 1) El usuario hace clic sobre «Pacientes» en el menú de la barra lateral. 2) Hace clic en uno de los pacientes del listado. 3) Pulsa en la pestaña «Revisiones» 4) Hace clic en el enlace «Ver revisión» de una revisión del historial. 	
Post-condición	<ul style="list-style-type: none"> • Se muestra la información completa de la revisión. 	

Tabla 32 CU Ver revisión

CU	Crear revisión	21
Descripción	Crear una nueva revisión a un paciente.	
Actor	Médico, Admin	
Pre-condición	<ul style="list-style-type: none"> • Haber iniciado sesión. • Hay al menos un paciente registrado al cargo del médico y no está inhabilitado. 	
Flujo básico	<ol style="list-style-type: none"> 1) El usuario hace clic sobre «Pacientes» en el menú de la barra lateral. 2) Hace clic en uno de los pacientes del listado. 3) Pulsa en la pestaña «Revisiones» 4) Hace clic sobre el botón «Nueva revisión» de alguno de los dos ojos. 5) Introduce todos los datos requeridos en el formulario que ha aparecido como una ventana modal. 6) Hacer clic en el botón «Crear», al final del formulario. 	
Post-condición	<ul style="list-style-type: none"> • Se muestra un mensaje informando del éxito o no de la operación realizada. • Se actualiza el listado de revisiones. 	

Tabla 33 CU Crear revisión



CU	Gestionar revisión	22
Descripción	Gestionar una revisión de un paciente: consultar, crear y eliminar.	
Actor	Médico, Admin	
Pre-condición	<ul style="list-style-type: none"> • Haber iniciado sesión. • Hay al menos un paciente registrado al cargo del médico con al menos una revisión registrada y no está inhabilitado. 	
Flujo básico	<ol style="list-style-type: none"> 1) El usuario hace clic sobre «Pacientes» en el menú de la barra lateral. 2) Hace clic en uno de los pacientes del listado. 3) Pulsa en la pestaña «Revisiones» 4) Hace clic en el enlace «Ver revisión» de una revisión del historial. 5) Seguidamente, puede pulsar: <ol style="list-style-type: none"> a) El botón  para editar sus datos. b) El botón  para eliminar la revisión. Finalmente, pulsa el botón de confirmación para eliminarla. 	
Post-condición	<ul style="list-style-type: none"> • Se muestra un mensaje informando del éxito o no de la operación. • Se actualiza la información de la revisión. 	

Tabla 34 CU Gestionar revisión

CU	Listar todos los pacientes	2
Descripción	Mostrar en un listado los pacientes dados de alta en el hospital. Por cada paciente muestra: iniciales, SIP y edad.	
Actor	Admin	
Pre-condición	<ul style="list-style-type: none"> Haber iniciado sesión. 	
Flujo básico	1) El usuario hace clic sobre «Pacientes» en el menú de la barra lateral.	
Post-condición	<ul style="list-style-type: none"> Se muestra el listado de pacientes. 	

Tabla 35 CU Listar todos los pacientes

CU	Listar pacientes inhabilitados	3
Descripción	Mostrar en un listado los pacientes dados de baja (inhabilitados) en el hospital. Por cada paciente muestra: iniciales, SIP y edad.	
Actor	Admin	
Pre-condición	<ul style="list-style-type: none"> Haber iniciado sesión. 	
Flujo básico	1) El usuario hace clic sobre «Pacientes» en el menú de la barra lateral. 2) Pulsa sobre el enlace «Pacientes inhabilitados»	
Post-condición	<ul style="list-style-type: none"> Se muestra el listado de pacientes. 	

Tabla 36 CU Listar pacientes inhabilitados

CU	Buscar pacientes	7
Descripción	Buscar un paciente o conjunto de pacientes habilitados por SIP.	
Actor	Admin	
Pre-condición	<ul style="list-style-type: none"> Haber iniciado sesión. 	
Flujo básico	1) El usuario hace clic sobre «Pacientes» en el menú de la barra lateral. 2) En el campo de texto bajo la etiqueta Buscar pacientes introduce un conjunto de números para buscar pacientes por SIP.	
Post-condición	<ul style="list-style-type: none"> La lista de pacientes se actualiza con los pacientes cuyo SIP contenga los números introducidos en la búsqueda y estén datos de alta (habilitados) en el sistema. 	

Tabla 37 CU Buscar pacientes

CU	Ver paciente con vista administrador	9
Descripción	Consultar información del paciente con vista de administrador.	
Actor	Admin	
Pre-condición	<ul style="list-style-type: none"> Haber iniciado sesión. Hay al menos un paciente registrado y no está inhabilitado. 	
Flujo básico	1) El usuario hace clic sobre «Pacientes» en el menú de la barra lateral. 2) Hace clic en uno de los pacientes del listado.	
Post-condición	<ul style="list-style-type: none"> Se muestra la información del paciente con vista de administrador. 	

Tabla 38 CU Ver paciente con vista administrador

CU	Inhabilitar paciente	23
Descripción	Inhabilita un paciente.	
Actor	Admin	
Pre-condición	<ul style="list-style-type: none"> Haber iniciado sesión. Hay al menos un paciente registrado y no está inhabilitado. 	
Flujo básico	<ol style="list-style-type: none"> El usuario hace clic sobre «Pacientes» en el menú de la barra lateral. Hace clic en uno de los pacientes del listado. Hace clic en el botón «Inhabilitar paciente». Hace clic en el botón de confirmación para dar de baja del sistema al paciente. 	
Post-condición	<ul style="list-style-type: none"> Se muestra un mensaje informando del éxito o no de la operación. Se actualiza la información del paciente. 	

Tabla 39 CU Inhabilitar paciente

CU	Habilitar paciente	24
Descripción	Habilita un paciente.	
Actor	Admin	
Pre-condición	<ul style="list-style-type: none"> Haber iniciado sesión. Hay al menos un paciente registrado y está inhabilitado. 	
Flujo básico	<ol style="list-style-type: none"> El usuario hace clic sobre «Pacientes» en el menú de la barra lateral. Hace clic en uno de los pacientes del listado. Hace clic en el botón «Habilitar paciente». Hace clic en el botón de confirmación para dar de alta (habilitar) en el sistema al paciente. 	
Post-condición	<ul style="list-style-type: none"> Se muestra un mensaje informando del éxito o no de la operación. Se actualiza la información del paciente. 	

Tabla 40 CU Habilitar paciente

Por último, el diagrama correspondiente a las gráficas de la población de pacientes (Ilustración 14). El acceso es el mismo, pero para el administrador la población de pacientes representada es la de todos los pacientes en el sistema.

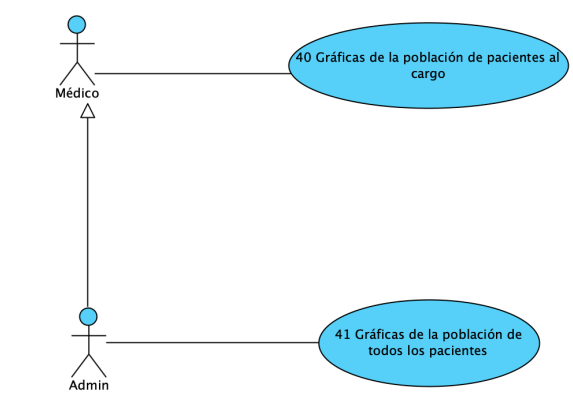


Ilustración 14 Diagrama de CU - Gráficas de población de pacientes

CU	Gráficas de la población de pacientes al cargo	40
Descripción	Muestra gráficas de la población de pacientes al cargo	
Actor	Médico	
Pre-condición	<ul style="list-style-type: none"> • Haber iniciado sesión. 	
Flujo básico	1) El usuario hace clic sobre «Inicio» en el menú de la barra lateral.	
Post-condición	<ul style="list-style-type: none"> • Se muestra el número total de la población (pacientes al cargo) y 7 gráficas diferentes combinando las variables: enfermedad, sexo, edad y antecedentes médicos. 	

Tabla 41 CU Gráficas de la población de pacientes al cargo

CU	Gráficas de la población de todos los pacientes	41
Descripción	Muestra gráficas de la población de pacientes al cargo	
Actor	Admin	
Pre-condición	<ul style="list-style-type: none"> • Haber iniciado sesión. 	
Flujo básico	1) El usuario hace clic sobre «Inicio» en el menú de la barra lateral.	
Post-condición	<ul style="list-style-type: none"> • Se muestra el número total de la población (todos los pacientes) y 7 gráficas diferentes combinando las variables: enfermedad, sexo, edad y antecedentes médicos. 	

Tabla 42 CU Gráficas de la población de todos los pacientes

4.4. Mockups

Los *mockups* son representaciones visuales que muestran cómo se verá y funcionará un *software* antes del desarrollo. Son herramientas esenciales para planificar y diseñar aplicaciones de manera efectiva, permitiendo corregir problemas antes de la implementación ahorrando recursos.

Se han diseñado unos *mockups* de alta fidelidad para mostrar una representación más detallada y cercana al producto final. En esta nueva versión los elementos gráficos como imágenes, iconos o gráficos son realmente importantes para solucionar los problemas de usabilidad anteriores y, en consecuencia, mejorar la satisfacción del usuario. Para diseñarlos, se ha hecho uso de la herramienta Figma [38], junto con las librerías de Bootstrap para Figma. Bootstrap es la librería de estilo que se ha seleccionado para la mayoría de los componentes que conforman la interfaz del prototipo AGC-MAC.

Los requisitos especificados son en mayoría relativos a la gestión y análisis de los pacientes y, por tanto, los *mockups* diseñados también. La validación se ha realizado en colaboración con el Dr. Rubén Cabrera del Hospital de la Vega Baja de Orihuela, cómo se ha comentado en la sección **Metodología**. A continuación, se enseñan los diseños que han sido necesarios elaborar para mejorar el prototipo inicial y el motivo de ellos:

- *Ilustración 15 Mockup Resumen del paciente y acciones rápidas:* Ventana que muestra los datos personales del paciente y un resumen de las enfermedades y antecedentes médicos y el último dato antropométrico. También admite la creación de cualquiera de estos tres objetos directamente desde esta ventana. Este diseño de pestañas permite la navegación rápida entre cualquiera de sus datos, frente al redireccionamiento de páginas del prototipo inicial. Entre sus desventajas, se realizan más llamadas al servidor para tener disponibles todos los datos al momento.
- *Ilustración 16 Mockup Revisiones del paciente:* Pestaña con línea temporal con el historial de las revisiones del paciente, una por cada por ojo. Cada revisión del historial indica el medicamento que se aplicó, la medición de agudeza visual que se le realizó y si

hubo escáner de retina. Para ver con más detalle la revisión, se puede abrir en otra página pulsando en «Ver revisión». El diseño de las líneas temporales, separadas por ojo, y la no saturación con información no esencial permite ver y crear revisiones históricas (o nuevas) con agilidad.

- *Ilustración 17 Mockup Nueva revisión colapsado*: Formulario con los elementos colapsados para la creación de una revisión de un ojo del paciente. En el mismo formulario, el usuario puede añadir la medición, el medicamento y el escáner de retina. Este diseño supone una mejora de usabilidad frente al prototipo inicial en el que cada uno de los datos tenía una página aparte.
- *Ilustración 19 Mockup Nueva revisión desplegado*: Formulario con los elementos desplegados para la creación de una revisión de un ojo del paciente.
- *Ilustración 25 Mockup Detalle de revisión del paciente*: Ventana que muestra todos los detalles de una revisión de forma estructurada, sin necesidad de navegar por la aplicación como ocurría en la versión anterior.
- *Ilustración 18 Mockup Ventana modal para la creación de objetos*: Ventana modal que aparece para completar cualquier formulario de la aplicación. En la versión anterior la apertura y cierre de los formularios era confusa resultando en problemas de usabilidad.
- *Ilustración 20 Mockup Gráficas del paciente*: Pestaña que visualiza las gráficas acerca del paciente. Cabe recordar que el prototipo inicial incluía la gráfica del coste total del paciente en el tiempo junto con la de la evolución, pero se ha eliminado provisionalmente en esta versión por problemas con la integración de la posibilidad de crear datos históricos. Respecto a la gráfica de evolución de la agudeza visual del paciente, se han sobrepuesto los medicamentos aplicados en el tiempo para observar cómo ha respondido el paciente.
- *Ilustración 21 Mockup enfermedades del paciente*: Pestaña que muestra en formato de tabla un listado de las enfermedades que padece el paciente con toda la información.
- *Ilustración 22 Mockup Antecedentes médicos del paciente*: Pestaña que muestra en formato tabla un listado de los antecedentes médicos del paciente con toda la información.
- *Ilustración 23 Mockup Datos antropométricos del paciente*: Pestaña que muestra los datos antropométricos en formato tabla un listado de los datos antropométricos del paciente con toda la información.

Cabe señalar que los *mockups* diseñados no muestran la vista completa de la aplicación, solamente la parte central donde está el contenido, a falta del encabezado i la barra lateral.

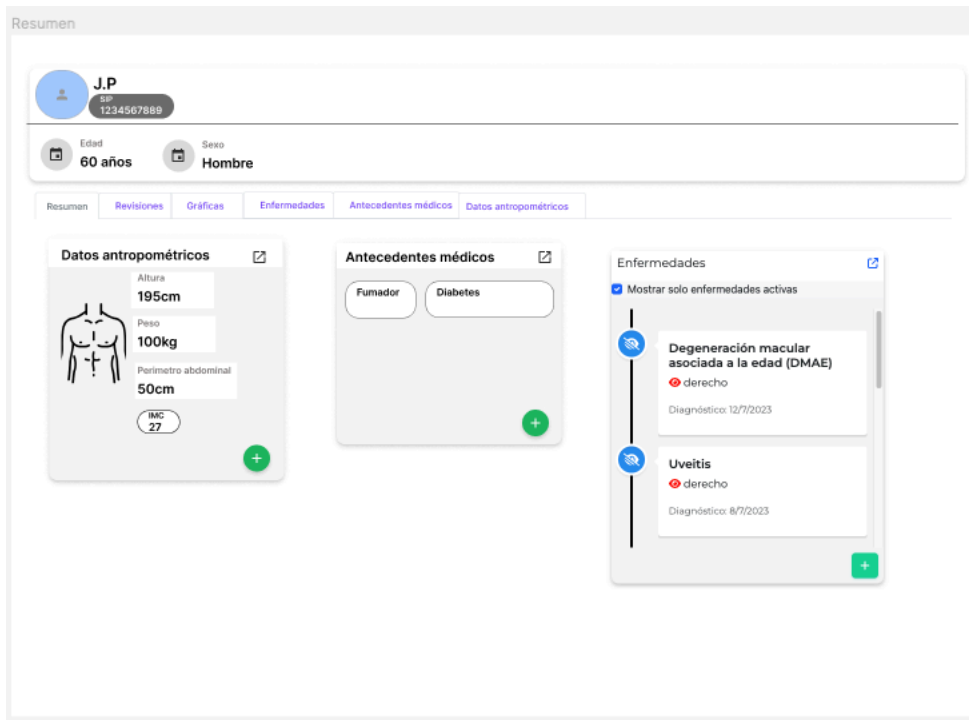


Ilustración 15 Mockup Resumen del paciente y acciones rápidas

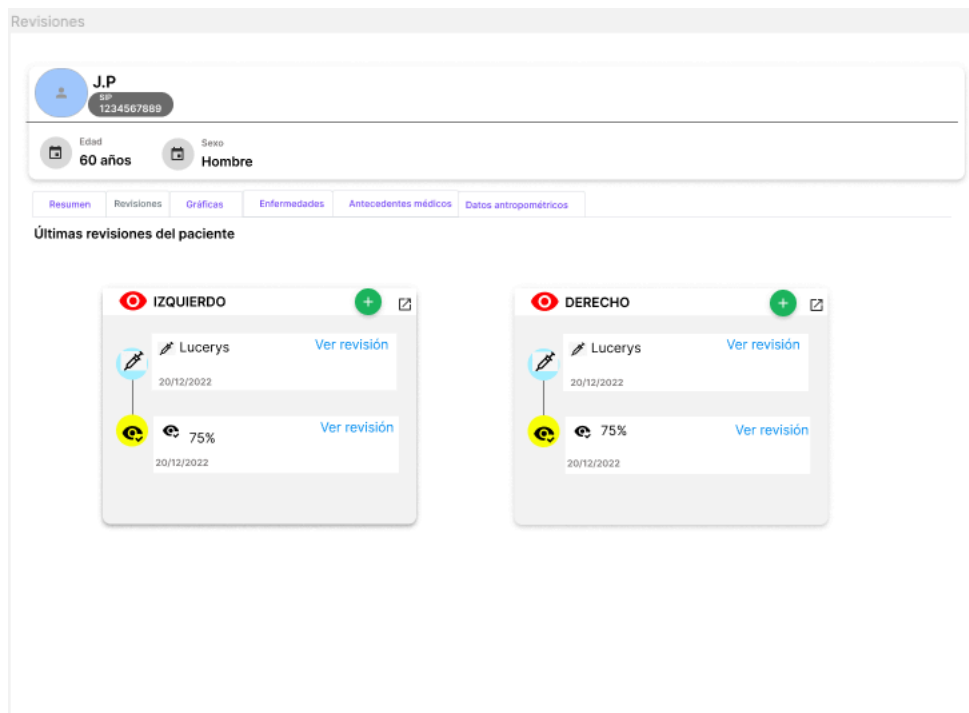


Ilustración 16 Mockup Revisiones del paciente

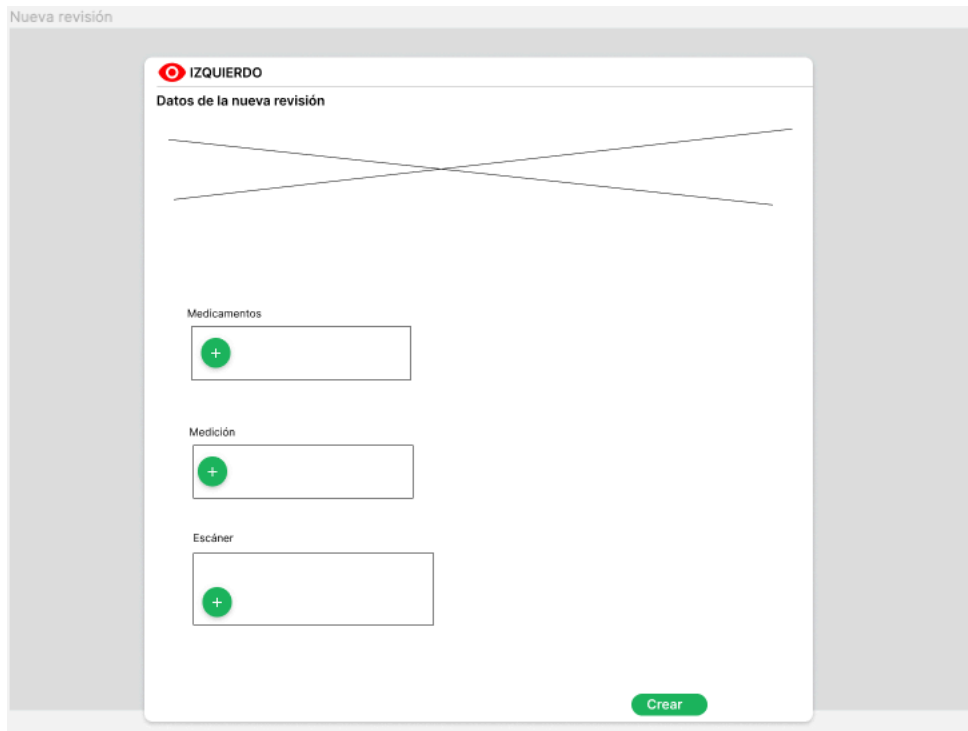


Ilustración 17 Mockup Nueva revisión colapsado

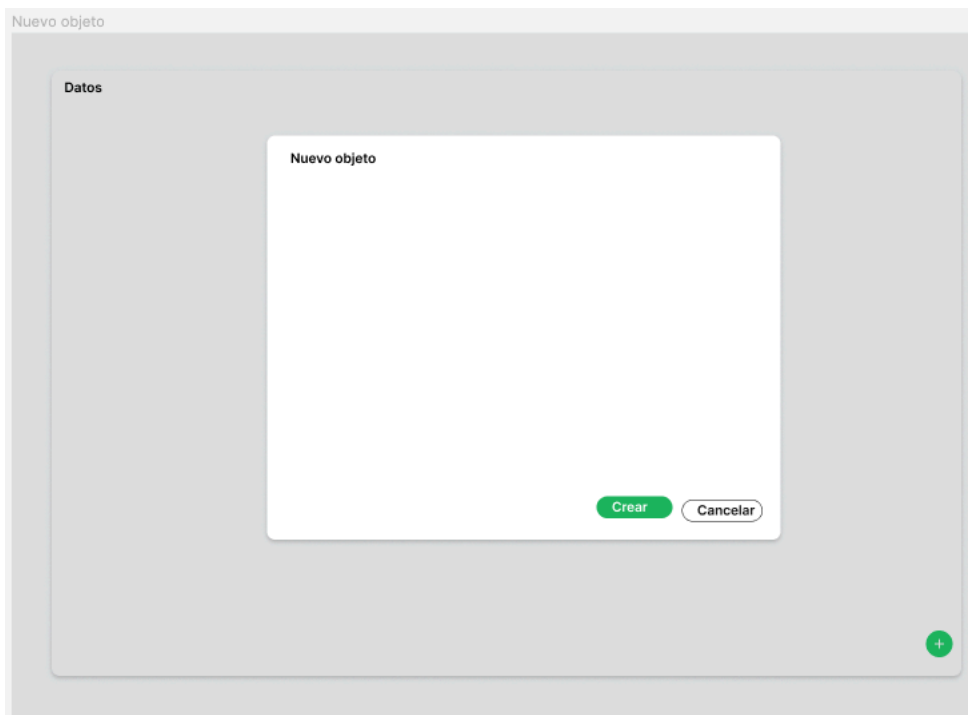


Ilustración 18 Mockup Ventana modal para la creación de objetos

Nueva revisión

IZQUIERDO

Datos de la nueva revisión

X

Medicamentos

✖ Select ▼

+

Medición

✖

0.05

No Sí

No Sí

No Sí

Escáner

No Sí

No Sí

No Sí

No Sí

No Sí

No Sí

No Sí

Crear
Cancelar

Ilustración 19 Mockup Nueva revisión desplegado



Ilustración 20 Mockup Gráficas del paciente

Enfermedades

J.P
SPR
1234567889

Edad: **60 años** Sexo: **Hombre**

Resumen Revisiones Gráficas Enfermedades Antecedentes médicos Datos antropométricos

[+ Nueva enfermedad](#)

Nº	Enfermedad	Ojo	Descripción	Fecha de diagnóstico	Alta
1	Obstrucción de vena central de la retina	DERECHO	aaa	14/7/2023	No

Ilustración 21 Mockup enfermedades del paciente

Antecedentes medicos

J.P
SPR
1234567889

Edad: **60 años** Sexo: **Hombre**

Resumen Revisiones Gráficas Enfermedades Antecedentes médicos Datos antropométricos

[+ Nuevo antecedente](#)

Nº	Nombre	Descripción
1	Dislipemia	ooo

Ilustración 22 Mockup Antecedentes médicos del paciente

Antecedentes medicos

J.P
SIP
123456789

Edad **60 años** Sexo **Hombre**

Resumen Revisiones Gráficas Enfermedades **Antecedentes médicos** Datos antropométricos

[+ Nuevo dato](#)

Fecha de creación	Peso(Kg)	Altura(cm)	Perimetro abdominal	IMC
12/7/2023	2	2		99



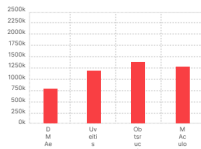



Ilustración 23 Mockup Datos antropométricos del paciente

PACIENTES
163

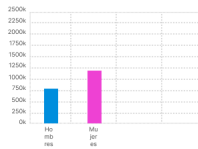
Pacientes por enfermedad

Secondary text



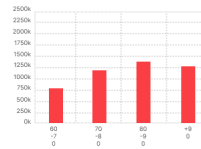
Pacientes por sexo

Secondary text



Pacientes por edad

Secondary text

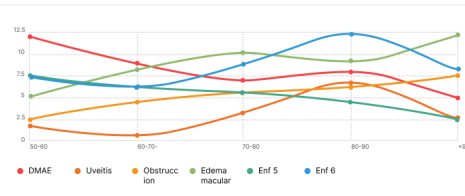


Pacientes por antecedente

Secondary text

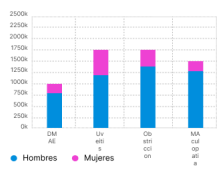


Pacientes por enfermedad y edad



Pacientes por enfermedad y sexo

Secondary text



Pacientes por enfermedad y antecedente

Secondary text

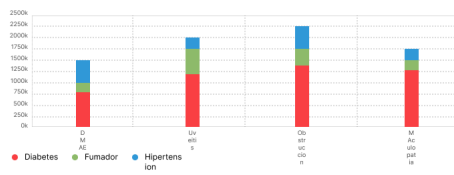


Ilustración 24 Mockup Gráficas de la población de pacientes del médico

Revision

Datos de la revisión

Médico: Juan Garcia Hospital: Hospital Vega Baja Fecha: 24/05/2023

Coste total: 500€

Motivo: Ejemplo

Observaciones: Ejemplo

Datos del paciente

SIP: 1234567889 Iniciales: A. A Sexo: Femenino Edad: 56 años Ojo afectado: IZQUIERDO

Medición

Agudeza visual: 56% Percepción de luz: Sí Movimiento de manos: Sí Cuenta dedos: Sí

Escáner de retina

Desprendimiento neurosensorial: No DRIL: Sí

Medicamento

Nombre: Lucentis Coste: 500€

Ilustración 25 Mockup Detalle de revisión del paciente

4.5. Funcionalidad clínico-genómica: gestión de variaciones y relevancia clínica

Uno de los objetivos de este Trabajo de Fin de Máster, y presente en el título, es la gestión y análisis de datos genómicos y clínicos. Sin embargo, los requisitos funcionales especificados, e implementados, en la sección 4.2. solamente se encargan de la gestión de los clínicos. En esta sección se presenta la especificación de una funcionalidad para la gestión de datos genómicos de los pacientes que pretende personalizar el tratamiento de los pacientes con enfermedades de la retina-macula. Por tanto, aplicando la genómica en el ámbito clínico. El desarrollo de esta funcionalidad merece una atención especial del resto de requisitos y es por este motivo, que se le dedica una sección separada.

Primero, es necesario describir con detalle en qué consiste esta funcionalidad clínico-genómica. Con este nuevo requisito funcional, el sistema ofrece la posibilidad de gestionar (añadir y eliminar) las variaciones genómicas del paciente e informar al médico de la relevancia clínica de estas en las enfermedades diagnosticadas al paciente. La relevancia clínica se clasifica en los niveles: causante del trastorno o factor de riesgo, rol incierto, no causa trastornos o tiene efectos

protectores, afecta a la respuesta de medicamentos o trastornos, y no proporcionado. Con esta relevancia clínica, el médico puede tomar decisiones respecto a los medicamentos y tratamientos, si lo considera necesario, para adaptarse al perfil del paciente.

4.5.1. Diagrama de casos de uso

El segundo paso es especificar los casos de uso y modelarlos en el diagrama. Se han añadido un total de tres casos de uso con ambos actores implicados (médico y administrador). En la Ilustración 26 se pueden observar los cambios del diagrama en color verde. A continuación, se describen con detalle cada uno de ellos, siguiendo la misma estructura que en la sección **Diagramas de casos de uso**

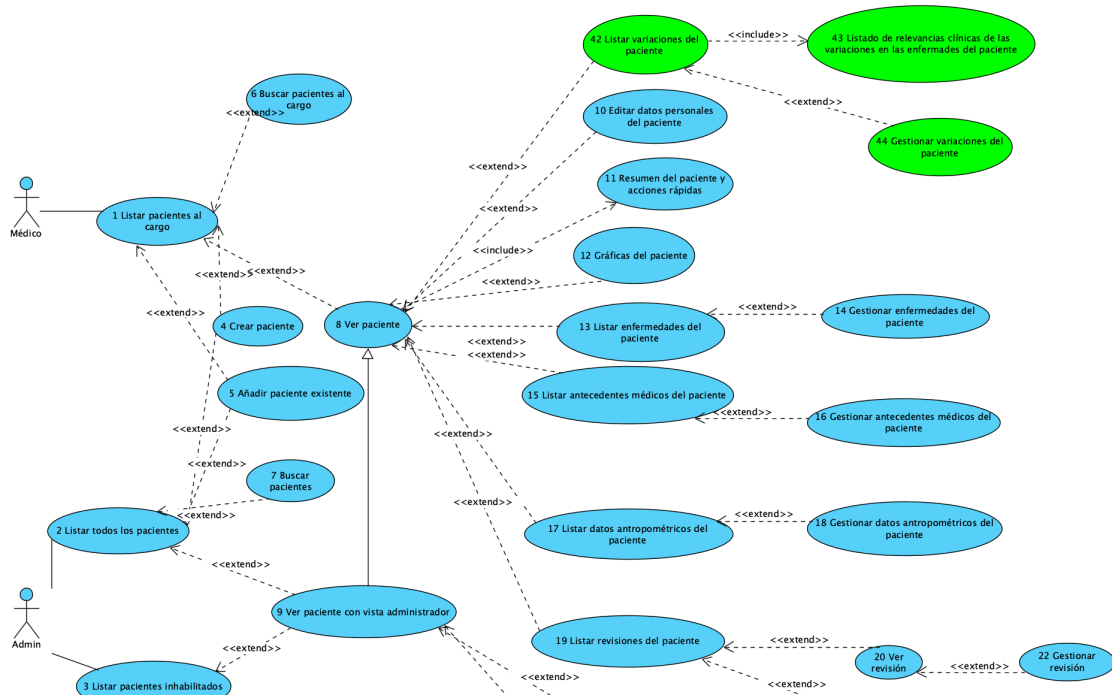


Ilustración 26 Diagrama de CU con gestión de variaciones genómicas y relevancia clínica (Parcial)

CU	Listar variaciones del paciente	42
Descripción	Mostrar un listado de las variaciones que el paciente tiene en su genoma.	
Actor	Médico, Admin	
Pre-condición	<ul style="list-style-type: none"> Haber iniciado sesión. Hay al menos un paciente registrado al cargo del médico y no está inhabilitado. 	
Flujo básico	<ol style="list-style-type: none"> El usuario hace clic sobre «Pacientes» en el menú de la barra lateral. Hace clic en uno de los pacientes del listado. Pulsa en la pestaña «Genómica» 	
Post-condición	<ul style="list-style-type: none"> Se muestra el listado de variaciones del paciente. 	

Tabla 43 CU Listar variaciones


CU	Gestionar variaciones del paciente	44
Descripción	Gestionar las variaciones del paciente: crear y eliminar.	
Actor	Médico, Admin	
Pre-condición	<ul style="list-style-type: none"> • Haber iniciado sesión. • Hay al menos un paciente registrado al cargo del médico y no está inhabilitado. 	
Flujo básico	<ol style="list-style-type: none"> 1) El usuario hace clic sobre «Pacientes» en el menú de la barra lateral. 2) Hace clic en uno de los pacientes del listado. 3) Pulsa en la pestaña «Genómica» 4) Seguidamente, puede pulsar: <ol style="list-style-type: none"> a) «Nueva variación» para registrar una nueva variación. b) El botón  de una variación de la lista para eliminarla. Finalmente, pulsa el botón de confirmación para eliminarla. 	
Post-condición	<ul style="list-style-type: none"> • Se muestra un mensaje informando del éxito o no de la operación. • Se actualiza el listado de variaciones. 	

Tabla 44 CU Gestionar variaciones

CU	Listar relevancias clínicas de las variaciones en las enfermedades del paciente	43
Descripción	Mostrar un listado de las relevancias clínicas de las variaciones en las enfermedades del paciente.	
Actor	Médico, Admin	
Pre-condición	<ul style="list-style-type: none"> • Haber iniciado sesión. • Hay al menos un paciente registrado al cargo del médico y no está inhabilitado. 	
Flujo básico	<ol style="list-style-type: none"> 1) El usuario hace clic sobre «Pacientes» en el menú de la barra lateral. 2) Hace clic en uno de los pacientes del listado. 3) Pulsa en la pestaña «Genómica» 	
Post-condición	<ul style="list-style-type: none"> • Se muestra el listado de las relevancias clínicas de las variaciones en las enfermedades del paciente. 	

Tabla 45 Listar relevancias clínicas de las variaciones en las enfermedades del paciente

4.5.3 Mockup

Finalmente, igual que el resto de los requisitos funcionales, se ha diseñado un *mockup* (Ilustración 27) para la prototipar la interfaz de usuario de esta funcionalidad. En cambio, este *mockup* no ha sido validado, ya que los requisitos de tipo genómico no son actualmente prioritarios para los *skateholders*. En consecuencia, la funcionalidad **no ha sido implementada** en el prototipo AGC-MAC, aunque se espera que en un futuro el *software* sea capaz de gestionar datos genómicos.

4.5.2. Consultar la evidencia

Para que viabilidad de estas funcionalidades sea posible, es necesario que el *software* extraiga la evidencia disponible para poder ofrecer, por ejemplo, la relevancia clínica de las variaciones. A diferencia del resto de objetos como las revisiones, el usuario final no crea una base de datos de fenotipos, variaciones y relevancias clínicas. Por tanto, el sistema de extraer la información de algún repositorio confiable.

Como se ha analizado en el **Capítulo 3**, el Oráculo Genómico Delfos permite consultar la evidencia disponible acerca de las variaciones asociadas a fenotipos, incluyendo su relevancia clínica. Se plantea la comunicación del AGC-MAC con el Oráculo Genómico de Delfos como repositorio para obtener los datos genómicos necesarios, que se representarían el modelo unión

presentado, para el diseño de la funcionalidad. Además, está desarrollada y desplegada en los servidores PROS/UPV, igual que el AGC-MAC (**Capítulo 6. Despliegue**). Este último detalle es una ventaja en términos de interoperabilidad, seguridad y rendimiento frente a otras posibles alternativas.

The mockup shows a patient profile for J.P. (SIP: 1234567889) with a navigation menu including Resumen, Revisiones, Gráficas, Enfermedades, Antecedentes médicos, Datos antropométricos, and Genómica. A '+ Nueva variante' button is present. The 'Lista de variantes' section contains two entries: 'Chr1:g.94098874-94098874:A>T(GRCh38)' and 'Chr1:g.94111561-94111561:G>A(GRCh38)', each with a red trash icon. The 'Relevancia clínica de las variantes con las enfermedades del paciente' section lists two findings: 'Chr1:g.94098874-94098874:A>T(GRCh38) afecta los medicamentos o el tratamiento de la DMAE' and 'Chr1:g.94111561-94111561:G>A(GRCh38) causa o incrementa el riesgo de la DMAE'.

Ilustración 27 Mockup Gestión de variaciones genómicas e información de la relevancia clínica

Capítulo 5

Modelo Conceptual de la Patología Macular (Extensión)

El **modelado conceptual** en el desarrollo de *software* desempeña un papel fundamental al permitir a los equipos de desarrollo comprender y comunicar de manera efectiva la esencia de un sistema antes de entrar en detalles técnicos. Esta fase inicial de diseño proporciona una representación abstracta y simplificada de los conceptos clave y las interacciones dentro del *software*, lo que ayuda a establecer una base sólida para el proyecto. Además, el modelado conceptual facilita la detección temprana de posibles conflictos o ambigüedades en los requisitos, lo que ahorra tiempo y recursos en etapas posteriores del desarrollo.

Este capítulo presenta

- El primer Modelo de la Patología Macular (MCPM)
- La extensión del MCPM para la gestión y análisis de datos clínicos que se genera a partir de los requisitos especificados.
- La extensión del MCPM con la vista genómica para plantear el futuro desarrollo de la funcionalidad clínico-genómica planteada.

5.1. Primer modelo

El primer modelo conceptual de la patología macular (Ilustración 28 MCPM) desarrollado por Alejandro Marco ha supuesto una gran base de la que sobre trabajar para la extensión. El modelo se elaboró con la colaboración de expertos en el tratamiento de pacientes con enfermedades maculares y se consiguió una buena representación del dominio. El modelo se divide en tres vistas: Estructural, Paciente y Evolución.

- *Vista Estructural*: almacena la información de los médicos y hospitales del sistema. Esta parte está más enfocada a la gestión y el control de los recursos humanos e infraestructura del sistema.
- *Vista Paciente*: grupo compuesto por las clases: «Paciente», «Enfermedad», «Dato_Antropometrico» y «Anteced_Medico», que forman parte de la información registrada del paciente durante el inicio del tratamiento.

- *Vista Evolución.* Compuesta por el resto de las clases del MCPM, permite realizar el seguimiento de la evolución del paciente gracias al almacenamiento de la información relativa a sus revisiones y pruebas realizadas.

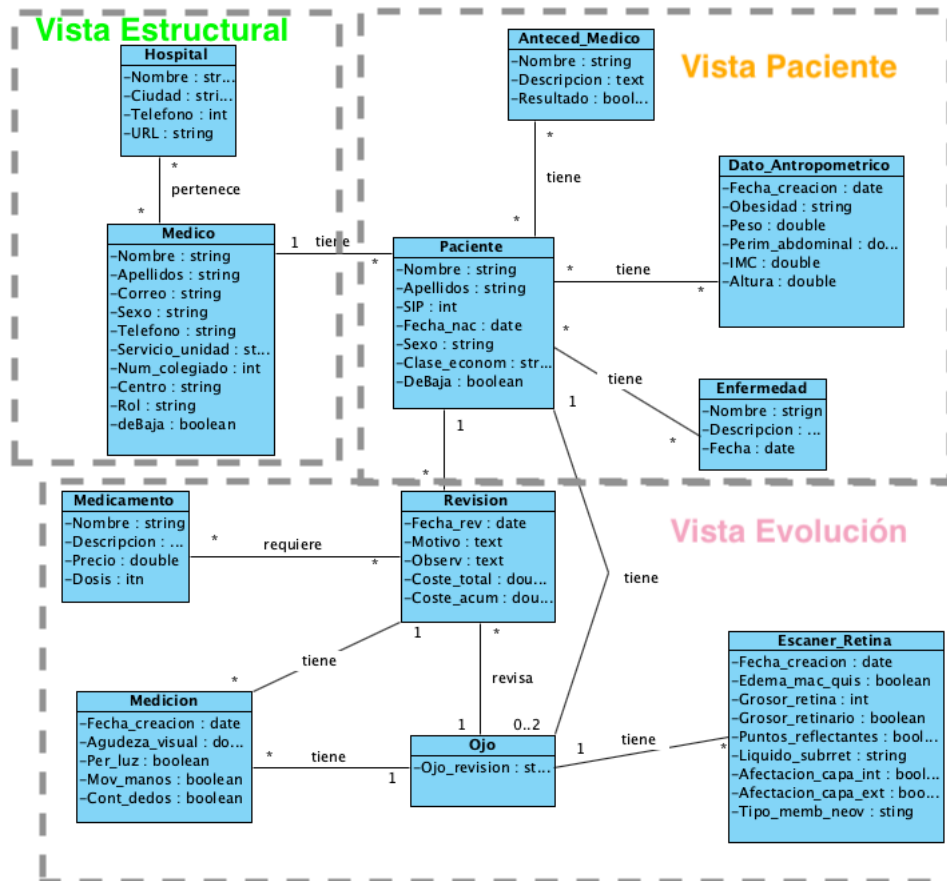


Ilustración 28 MCPM del prototipo G-MAC [1]

Antes de realizar ningún análisis en el dominio para la extensión y modificación del modelo, encontramos un problema a atacar para lograr nuestros objetivos. El modelo no está orientado a la explotación para poder realizar ciencia de datos y generar nuevo conocimiento del conjunto de los pacientes. En la clase «Medicamento», se almacena el nombre del medicamento que fue aplicado al paciente en una o más revisiones junto con los datos que le dan el contexto. Es decir, se crea un registro por cada medicamento aplicado que, además, podría diferir por nombre con otros registros siendo el mismo el medicamento. La información de los medicamentos existentes debe estar separada de sus aplicaciones en revisiones. No hay una manera directa de obtener la lista de medicamentos disponibles en el sistema, cuántas veces ha sido aplicado o cuál es su precio actual. En definitiva, este problema dificulta el análisis. Pues este mismo problema también ocurre con las clases «Enfermedad» y «Anteced_Médico».

5.2. Modelo extendido

Para extender el modelo de la manera más cercana a la realidad de las enfermedades de la patología macular se han llevado a cabo reuniones semanales con el Dr. Rubén Cabrera. Se ha revisado el primer modelo de forma exhaustiva para encontrar imprecisiones o actualizar los

conceptos con respecto al año en que se desarrolló. Esta revisión también ha consistido en analizar todas las actividades implicadas en la vida clínica de un paciente con enfermedades maculares, así como también las asociaciones entre los conceptos.

El modelo extendido (Ilustración 29) mantiene las mismas tres vistas que el primer modelo: pacientes, estructural y evolución. Las extensiones realizadas al modelo se pueden agrupar en dos partes: **refinamiento del dominio**, **asociaciones** y **orientación a la agregación de datos**. Seguidamente, se exponen cada uno de los cambios que se han producido y por qué motivo.

Respecto al **refinamiento del dominio**, se han anonimizado los nombres y fechas de nacimiento de los pacientes. En su lugar, se han cambiado por las iniciales y el año de nacimiento, respectivamente. Este cambio se debe a una condición comunicada por los *skateholders* para cumplir con normas de ética y protección de datos.

También se han cambiado los roles pasando de Médico y Colaborador a Médico y Admin. Estos dos roles son más precisos. El rol de médico es el rol que permite hacer cualquier gestión de los pacientes de un médico, mientras que un administrador (Admin) tiene acceso a otras operaciones más allá de las relativas a los pacientes.

La clase «Paciente» incluía el atributo de clase económica, el cual se ha decidido eliminar en la nueva versión. Aunque es una propiedad de gran interés para la investigación en búsqueda de asociaciones, es un atributo que es difícil de recoger i clasificar. Típicamente existen tres clases sociales: baja, media y alta. Para clasificar a un paciente, el doctor debe extraer información acerca de la condición económica del paciente y clasificarlo en una de las clases, algo que no es tarea fácil. Además, puede entorpecer la calidad asistencial ralentizando la consulta o incomodando al paciente.

En la clase «Dato_Antropometrico», se ha descartado el atributo obesidad. En el prototipo se podía añadir si el paciente era obeso o no. Esta información es redundante teniendo en cuenta el índice de masa corporal (IMC) que establece varios niveles de obesidad. Por tanto, se ha eliminado la obesidad del modelo. Otro cambio en esta clase es como se almacena el IMC, que en la extensión es auto calculado con el peso y la altura frente a la introducción manual en el prototipo.

Por lo que hace a las enfermedades de los pacientes, se ha incluido la propiedad de alta. Es decir, para registrar el alta de un paciente en una enfermedad. Normalmente cada ojo tiene una evolución independiente y se le puede dar de alta para un solo un ojo. También se ha renombrado el atributo fecha a fecha de diagnóstico, siendo mucho más claro el último. Aparte de las enfermedades, se ha removido el atributo resultado de la clase «Anteced_Medico», que era confuso saber a qué se refería.

En las revisiones de los pacientes, se incluía el coste acumulado como derivado de la suma del coste total de las revisiones anteriores. En la extensión se ha decidido eliminar provisionalmente el coste acumulado tras generar problemas con la inclusión de revisiones históricas, algo que no permitía el primer modelo. Respecto a los elementos de una revisión, medicamentos, mediciones y escáneres de retina, se ha eliminado la dosis del medicamento, ya que no estaba siendo utilizada por el software, y se han renombrado y actualizado los campos del escáner de retina.

La siguiente parte de la extensión se refiere a las **asociaciones** entre conceptos. Se han creado y eliminado relaciones, así como editado las cardinalidades. El primer cambio es la relación entre los pacientes y los hospitales. Inicialmente, un médico podía trabajar en diferentes hospitales,

pero esto no se ajustaba a la realidad del software. El *software* se distribuye en un hospital. Por tanto, un médico solo puede trabajar en uno a la vez.

Otro cambio significativo para el *software* es la posibilidad de que varios médicos puedan tener al cargo al mismo paciente, algo que suele ocurrir en este tipo de enfermedades y tratamientos. Con el primer modelo implementado, si dos médicos compartían un paciente, debían de registrar cada uno al paciente. Entonces existían dos registros (datos duplicados) que no permitían saber el uno del otro. El cambio supone dos ventajas frente al anterior: un médico no necesita volver a registrar a un paciente ya registrado para añadirlo a su lista, y facilita la comunicación entre médicos pudiendo acceder a su historial completo.

En la extensión del modelo se han creado las asociaciones entre «Revisión» y «Hospital», y «Revisión» y «Médico». Las dos relaciones deben estar representadas en MCPM, puesto que las revisiones son realizadas por un médico en un hospital. En el primer modelo se asumía que el médico que revisaba al paciente era su médico. Pero en la práctica ocurre que un paciente puede ser revisado por distintos médicos por diferentes razones. Incluso puede cambiar de hospital y es necesario conocer los todos los datos de sus revisiones pasadas. Con este cambio, los médicos de diferentes hospitales que compartan paciente también pueden conocer el hospital y el médico en el que se realizó la revisión.

El primer modelo no asigna las enfermedades a los ojos, siendo esto muy importante. Una enfermedad afecta al ojo derecho, al ojo izquierdo o ambos. Es posible que el paciente solo tenga una enfermedad macular en uno de los ojos o incluso, una diferente en cada ojo. Esta asociación ha sido añadida en la extensión.

Siguiendo la misma línea de separar los ojos, se ha cambiado la cardinalidad de la asociación «Revisión – Ojo» para que una revisión solo revise un ojo. Aunque en la misma consulta se pueden revisar los dos, sigue siendo una revisión por ojo. De esta forma, se separan también los medicamentos, mediciones y escáneres. En consecuencia, en una revisión de un ojo solo puede realizar una medición, un escáner y aplicar un medicamento. Tras este cambio, la relación entre un ojo y las mediciones y escáneres de retina son redundantes. Por tanto, se han quitado del modelo.

Por último, en esta parte, de las relaciones del paciente con las enfermedades, datos antropométricos y antecedentes médicos solo se ve afectada la cardinalidad de la asociación «Dato_Antropometrico - Paciente», que se ha cambiado a 1 a muchos frente a muchos a muchos del primer modelo. Un dato antropométrico solamente pertenece a un paciente.

La última parte de extensiones del primer modelo MCPM se trata de la **orientación a la agregación de datos**. En la introducción del modelo inicial, se exponía el problema que tenía para poder generar información agregada. Era posible, pero se dificultaba en gran medida. Uno de los objetivos del *software* es poder conocer más acerca de las enfermedades maculares como, por ejemplo, cuántos pacientes padecen DMAE y cuántos de estos son diabéticos. Para lograr esto con ciencia de datos, se ha de facilitar una herramienta con un modelo que lo permita de forma directa, rápida y sin ambigüedades en la información.

La estrategia que se ha seguido es añadir clases asociación que extraigan los datos no relevantes entre aquellas clases de interés para el análisis. Por ejemplo, en la asociación «Paciente – Enfermedad» se ha agregado la clase asociación con los atributos que tenía la clase original «Enfermedad». De este modo, en la nueva clase «Enfermedad» solo tenemos un conjunto fijo de las enfermedades que son tratadas en el sistema. Mientras que en la nueva clase asociación

«Paciente_Enfermedad», tenemos los datos relativos al paciente sobre la enfermedad. Del mismo modo, se ha extendido con los antecedentes médicos y medicamentos. No solo nos sirve para el análisis, también para tener una lista actualizada de las enfermedades, antecedentes médicos o medicamentos. Esto último ayuda a la mantenibilidad en el caso de que se cambie el precio de a medicamento o se renombre una enfermedad, por ejemplo. De hecho, el precio del medicamento se tenía que actualizar en el *frontend* de la aplicación, en el formulario concretamente.

Con esta extensión del modelo podemos gestionar los medicamentos, antecedentes médicos y enfermedades del sistema de una forma más sencilla. De hecho, se han añadido los siguientes medicamentos:

- *Iluvien*: se usa para abordar la pérdida de visión relacionada con el edema macular diabético cuando otros tratamientos disponibles no han tenido éxito. El edema macular diabético es una condición que afecta a algunas personas con diabetes y causa daño en la mácula. El ingrediente activo, el fármaco acetónido de fluocinolona, ayuda a reducir la inflamación y la hinchazón que se desarrollan gradualmente en la mácula. En consecuencia, Iluvien tiene el potencial de mejorar la visión deteriorada o prevenir su empeoramiento [29].
- *Beovu*: es medicamento que pertenece al grupo de medicamentos llamados agentes antineovascularización (anti-VEGF). Se utiliza para tratar la DMAE neovascular (húmeda). Beovu es otra opción más a considerar si se plantea probar con otro agente anti-VEGF [29].

Estos medicamentos se suman a los ya presentes: *Avastin*, *Eylea*, *Lucentis* y *Ozurdex*. También se han actualizado la lista de enfermedades de patología macular que puede padecer un paciente. Además de las existentes en el G-MAC (DMAE, edema macular diabético, obstrucción de vena central de la retina, obstrucción de la rama venosa retiniana, vasculopatía polipoidea y uveítis), en la nueva versión se incluyen las enfermedades:

- *Obstrucción central de la arteria*: ocurre cuando la arteria que lleva el oxígeno a las células neuronales de la retina en la parte posterior del ojo se bloquea. La falta de oxígeno puede provocar una pérdida grave de visión [14].
- *Edema macular pseudofáquico*: ocurre cuando se produce un acúmulo de fluido intrarretiniano en el área macular, que produce una pérdida de visión [13].
- *Telangiectasia macular tipo 1 y tipo 2*: La telangiectasia macular se desarrolla cuando hay problemas con los vasos sanguíneos pequeños alrededor de la fovea (centro de la mácula). Hay dos tipos de telangiectasia macular, y cada una afecta a los vasos sanguíneos de manera diferente [15].
- *Maculopatía*: esta enfermedad engloba a todas aquellas en las que la mácula de la retina sufre daños. La maculopatía se ha añadido en los casos que el paciente tenga una maculopatía, pero la enfermedad concreta no está en la lista o bien, aún no se le ha diagnosticado.

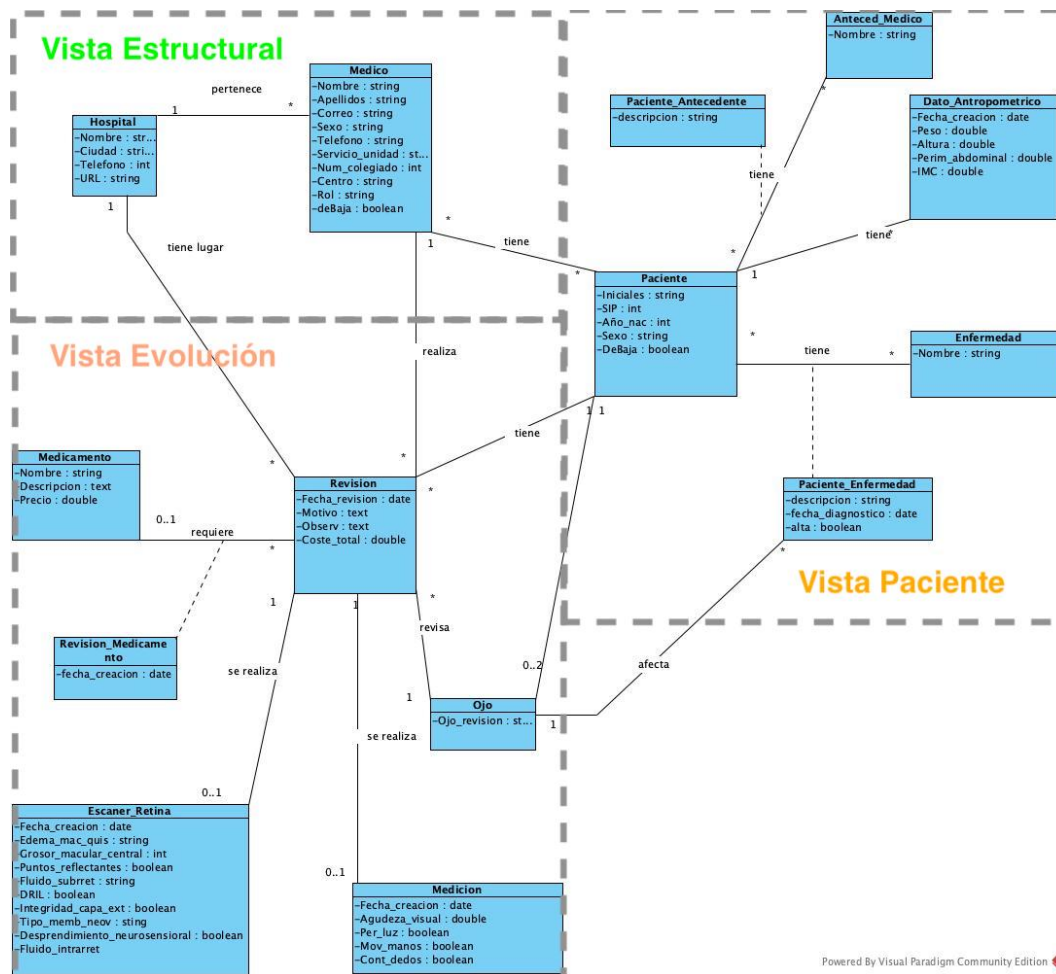


Ilustración 29 Modelo extendido MCPM

5.3. Modelo extendido con la Vista Genómica

Para construir el prototipo AGC-MAC disponemos del modelo expuesto en la sección 4.1. pero, como es evidente, precisa de una **Vista Genómica** para abordar la nueva funcionalidad clínico-genómica, aunque no implementada. Por tanto, se ha de modelar una extensión que tenga en cuenta los conceptos de relevancia clínica, variaciones, etc. Para ello, se hace uso del **Modelo Conceptual del Genoma Humano (CSG)** (Ilustración 30) para una mejor representación de la relevancia clínica, propuesto por García Simón [17].

El CSG para la relevancia clínica está formado por los siguientes conceptos:

- *Variation*: un cambio en la secuencia de ADN respecto a una de referencia.
- Phenotype: es la enfermedad o trastorno. Por ejemplo, la DMAE.
- *Certainty*: indica como de fuerte es la evidencia de la relación variación-fenotipo (*level_of_certainty*). También incluye el atributo *clinical_significance* (relevancia clínica) que indica el efecto que puede tener una variación sobre un fenotipo. Los valores que puede tomar la relevancia clínica son los siguientes:
 - *Pathogenic*: aumenta la susceptibilidad de la predisposición a un determinado trastorno mendeliano.

- *Benign*: reduce la susceptibilidad de la predisposición a un determinado trastorno mendeliano.
 - *Likely Benign*: pruebas sólidas a favor de reducir la susceptibilidad de la predisposición a un determinado trastorno mendeliano.
 - *Likely Pathogenic*: pruebas sólidas a favor de aumentar la susceptibilidad de la predisposición a un determinado trastorno mendeliano.
 - *Affects*: causa un fenotipo no relacionado con la enfermedad, como la intolerancia a la lactosa.
 - *Drug Response*: altera de algún modo la respuesta a un fármaco específico.
 - *Confers Sensitivity*: confiere sensibilidad a un fármaco específico.
 - *Association*: se ha identificado la asociación a un trastorno en un estudio GWAS.
 - *Uncertain Significance*: pruebas limitadas respecto a la patogenicidad.
 - *Protective*: disminuye el riesgo de padecer un trastorno.
 - *Conflicting data from submitters*: grupos dentro de un consorcio tienen interpretaciones contradictorias de una variación.
 - *Not Provided*: no se ha informado de significación clínica.
 - *Other*: cualquier otro valor posible.
- *Submitter*: es quien publica la certeza. Por ejemplo, un laboratorio clínico.
 - *Submission*: representa la información que publica el *submitter*.
 - *Clinical Actionability*: agrega todas las relevancias clínicas de la variación-fenotipo para obtener una evaluación más precisa de la relevancia clínica real. Los valores que puede tomar son los siguientes:
 - *Disorder causing or risk factor*: La variación causa un fenotipo o aumenta su probabilidad de aparición. Este valor considera los siguientes significados clínicos: patogénico, probablemente patogénico, afecta, factor de riesgo o asociación.
 - *Uncertain role*: El rol de la variación en el desarrollo de un fenotipo no está claro. Este valor considera las significaciones clínicas "inciertas" o cuando hay conflictos entre significaciones clínicas.
 - *Not disorder causing or protective effect*: La variación no causa el fenotipo o proporciona un efecto protector contra él. Este valor considera los siguientes significados clínicos: benigno, probablemente benigno, asociación no encontrada o protector.

- *Affects drugs or treatment response*: La variación afecta a la sensibilidad o respuesta de un fármaco o tratamiento. Este valor considera los siguientes significados clínicos: respuesta al fármaco o confiere sensibilidad.
- *Not provided*: Se desconoce el significado clínico de la variación. Este valor considera los siguientes significados clínicos: desconocido y no proporcionado.

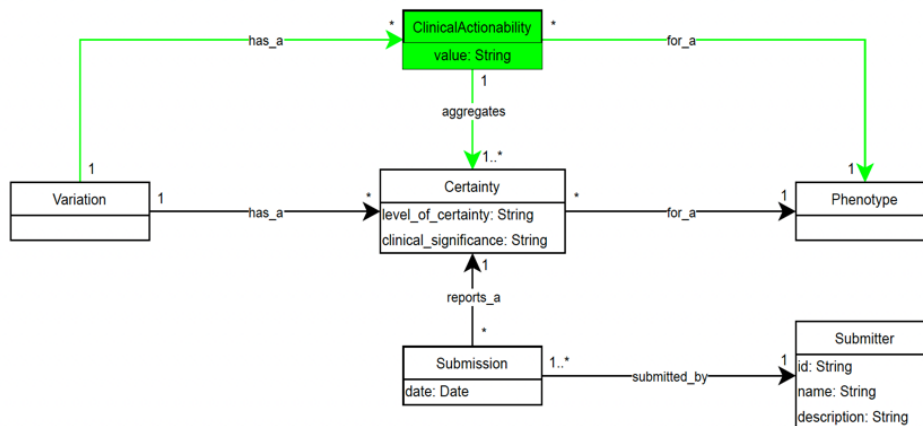
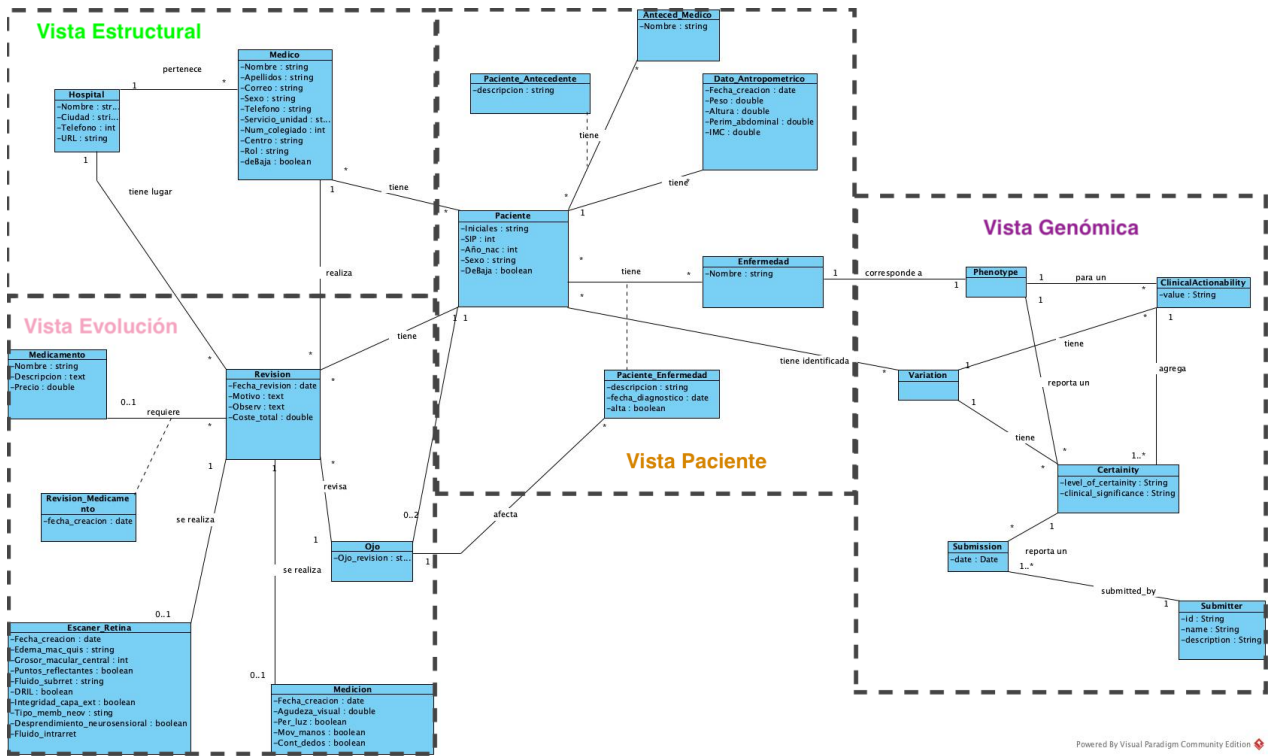


Ilustración 30 Modelo Conceptual del Genoma (CSG) para una mejor representación de la relevancia clínica (García Simón, 2022)

Una vez introducido el Modelo Conceptual del Genoma, se puede relacionar el dominio específico del requisito funcional con los conceptos del CSG. Una «*Variation*» identificada al paciente tiene una «*ClinicalActionability*» asociada a uno de los «*Phenotype*» que padece. Tras este *mapping*, lo siguiente es la extensión del modelo MCPM añadiendo la **Vista Genómica**, que se compone por los conceptos del CSG, a las otras tres vistas (Paciente, Estructural y Evolución), mediante las asociaciones «Enfermedad – Phetnoype» y «*Variation* – Paciente» (Ilustración 31). La primera asocia la enfermedad clínica con el fenotipo correspondiente, asociación de 1 a 1. Mientras que la segunda representa la identificación de variaciones en el genoma del paciente, de muchos a muchos, puesto que un paciente puede tener identificadas múltiples variaciones y una variación puede existir en muchos pacientes.

Gracias al modelo extendido con la Vista Genómica, se puede establecer una base de la que dirigir el desarrollo de funcionalidades que integren los ámbitos clínicos y genómicos. Por ejemplo, algunas más sofisticadas como gráficas que muestren la respuesta a un medicamento en pacientes, clasificados por variación genómica, y observar diferencias. En definitiva, un amplio abanico de posibilidades. Entre ellas, la funcionalidad planteada para gestionar variaciones de un paciente y conocer la relevancia clínica de estas en las enfermedades diagnosticadas



Powered By Visual Paradigm Community Edition

Ilustración 31 Unión del MCPM y del CSG para la relevancia clínica

Capítulo 6

Diseño e implementación de la solución

Para desarrollar un *software* profesional que satisfaga las necesidades de los médicos en la gestión de los pacientes con patologías maculares, es necesario el uso de las últimas tecnologías disponibles, las cuales que se hayan demostrado estables y confiables para el desarrollo de proyectos maduros y complejos. La solución implementada se trata de una **aplicación web** y consta de dos componentes, aparte de la base de datos, que se comunican entre sí, el *frontend* y el *backend*. En este capítulo, en la sección 6.1. se expone la arquitectura en la que se acoplan estos componentes, así como el detalle del diseño de cada uno de ellos en las secciones 6.2. y 6.3. Por último, en el **Anexo III – Fragmentos de código** se pueden observar fragmentos de código de la implementación.

6.1. Diseño de la arquitectura

La importancia de la arquitectura en el *software* radica en su capacidad para proporcionar una base sólida que permita la escalabilidad, mantenibilidad y eficiencia del desarrollo, así como la gestión de la complejidad, lo que resulta en un software más confiable y de mayor calidad.

El prototipo G-MAC seguía una arquitectura dividida claramente por dos componentes independientes que interactuaban entre sí: *frontend* y *backend* (Ilustración 32). El *backend* es el encargado de publicar los recursos del sistema y las operaciones permitidas para interactuar con ellos mediante una API REST [40], mientras que, el *frontend*, corresponde con la interfaz del usuario para comunicarse con el servidor (*backend*) de una forma más usable para un usuario final. El *frontend*, desarrollado con Angular [35], envía peticiones HTTP [47] con el cuerpo en formato JSON a la API REST del *backend* para solicitar o actualizar los recursos necesarios.

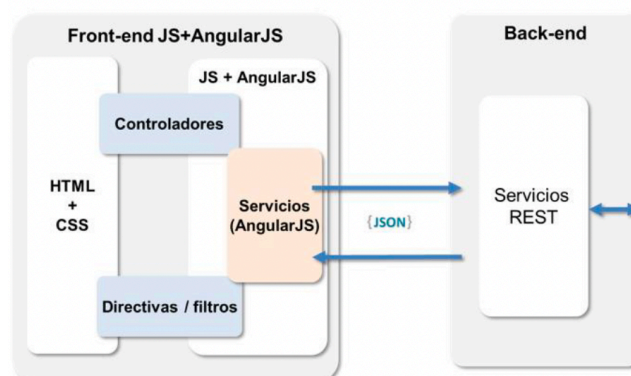


Ilustración 32 Arquitectura del prototipo G-MAC (Imagen del TFM de Alejandro Marco [1])

Gracias a esta arquitectura con una comunicación definida entre componentes, se pueden desacoplar y acoplar las piezas según se requieran de manera muy sencilla. En la nueva versión se ha decidido mantener la arquitectura (Ilustración 33), aunque con el cambio de tecnología del

frontend. Una alternativa de arquitectura podría haber sido de microservicios, pero aumenta la complejidad significativamente para el tamaño actual, relativamente pequeño, de la aplicación.

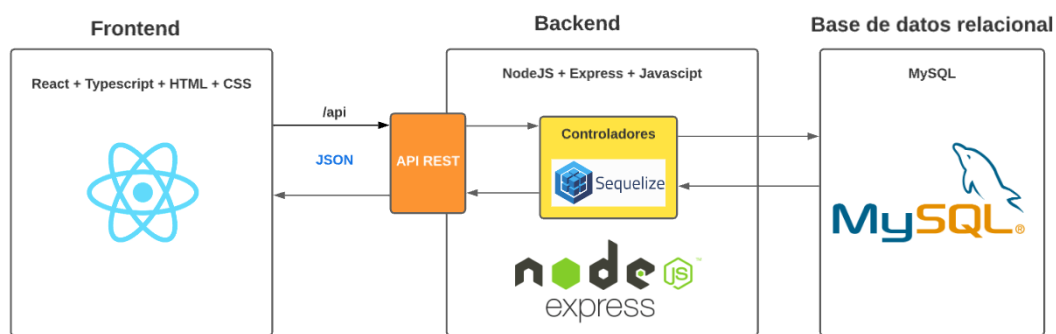


Ilustración 33 Arquitectura del prototipo AGC-MAC

6.2. Diseño de la parte del cliente (frontend)

6.2.1. Tecnología

El frontend del prototipo G-MAC se desarrolló con el *framework* Angular en el 2020. Desde entonces, la plataforma no se ha utilizado ni mantenido. En consecuencia, muchas de las librerías se han quedado obsoletas, además de tener vulnerabilidades en seguridad. En consecuencia, se plantearon dos vías desarrollo: actualizar el proyecto de Angular a la última versión o migrar a otra tecnología.

La elección de la mejor opción se vio influenciada, en parte, por las circunstancias del autor de este TFM. El autor tenía un conocimiento bajo de Angular y el aprendizaje de una nueva tecnología, junto con la reparación de errores de integración de librerías, podía ralentizar el proyecto. En cambio, sí que tenía un dominio medio con React [31].

React es una librería de código abierto de Facebook que se ha popularizado en la última década. Esta librería permite la creación de interfaces a partir de pequeños bloques reutilizables que se pueden utilizar en cualquier parte de la web. La programación de estos componentes se hace con una sintaxis llamada JSX, que combina los lenguajes HTML y Typescript para «jugar» con la interfaz dinámicamente, además de facilitar el mantenimiento al programador. Otra de sus ventajas es el manejo de estados de los componentes, estas piezas pueden mantener un estado del que puede depender la interfaz y conseguir una interacción más sofisticada que con Angular.

Por otro lado, uno de los requisitos que se presentan para el prototipo AGC-MAC es conseguir una interfaz mucho más usable y satisfactoria para los usuarios, el cual encaja con los puntos fuertes de React. Por tanto, este motivo y la experiencia previa han sido las principales razones para decantar la balanza hacia empezar el desarrollo del nuevo software con una migración a React. Cabe destacar que React también se utiliza en los *softwares* del Centro PROS.

6.2.2. Estructura

A diferencia de Angular, React no es un *framework*, sino una librería. Un *framework*, como Angular, incluye un entorno con los paquetes necesarios para desarrollar una aplicación. La mayor ventaja de los *framework* es que el programador no necesita recurrir, o lo necesita poco, a paquetes de terceros para completar las funcionalidades del *software*. Por ejemplo, el paquete que se encarga de realizar las llamadas al *backend*. Además de las librerías integradas, proporciona una estructura definida y rígida sobre la que montar la aplicación. Usar los paquetes y estructura del

framework asegura que, si se mantiene actualizado, la integración será compatible, segura y funcional.

En cambio, para construir una aplicación medianamente compleja con React es necesario instalar paquetes externos, o de lo contrario, no se puede desarrollar. Esto es claramente una desventaja, pero al mismo tiempo proporciona una flexibilidad total del funcionamiento de la aplicación. Obviamente, esto no sería viable sin el soporte masivo que tienen los paquetes para React.

Del mismo modo que los paquetes, React tampoco ofrece una guía en la que se deben estructurar los archivos dentro del proyecto. Este aspecto lo deja a decisión de los desarrolladores. Afortunadamente, existen diseños de estructuras que la comunidad de React recomienda seguir [41]. Siguiendo estas recomendaciones y adaptándolas al caso de AGC-MAC se ha diseñado una estructura que se muestra en la Ilustración 34. A continuación, se describe brevemente:

- *Models*: Define los modelos de datos del dominio de la aplicación. Para ello, se programa con el lenguaje tipado Typescript. Los modelos ayudan a mantener la consistencia dentro de la lógica de la aplicación. Por ejemplo, cuando se define el tipo del objeto devuelto en una llamada a la API. Por supuesto, los modelos definidos en el frontend deben coincidir con los del *backend*.
- *Pages*: En esta carpeta se guardan los archivos TSX que corresponden a los componentes raíz de las páginas de la aplicación web. Cuando se accede, por ejemplo, a la página /pacientes React carga el componente «PacientesPage». Cabe señalar que este archivo no incluye todos los elementos visibles en la página, solo aquellos que se deseen cambiar. Esto es gracias a la componentización de la interfaz y la gestión de React.
- *Components*: Incluye los componentes de la interfaz de usuario que puedan ser reutilizables como, por ejemplo, el botón de eliminación.
- *Features*: Corresponde a las interfaces que realicen una funcionalidad. Dentro de esta, está dividida en subcarpetas por modelos de datos (médicos, antecedentes, enfermedades...). Por ejemplo, una interfaz (TSX) existente es el formulario de creación de una revisión.
- *Layouts*: Contiene interfaces que son compartidas por diferentes páginas o componentes. Por ejemplo, el menú lateral está presente en todo momento.
- *Hooks*: Los *hooks* son un tipo funciones específicas de *React* que ayudan a extraer la lógica del componente relativa a la gestión de sus estados. Por ejemplo, la lista de pacientes es un estado del componente que se encarga de visualizarlos. Pues el hook, externaliza toda la lógica (inicialización, llamada a la API, actualización...).
- *APIs*: Esta carpeta contiene las interfaces para interactuar con las diferentes *API'S* con la que se comunique el *frontend*. Cada interfaz de cada API contiene cuatro métodos: *get*, *post*, *put* y *del*. Aunque la nueva versión solo solicita recursos al propio *backend*, esta estructura facilita la extensión con otras posibles APIs..
- *Services*: Corresponde a lógica de los componentes que se encarga de obtener, crear, modificar o eliminar los recursos. Está dividido en subcarpetas por los diferentes recursos existentes (modelos) y hacen uso de las *API'S*. Por ejemplo, una de las funciones de los servicios del hospital es: «cargarMedicosByHospital».

- *Utils*: Contiene diferentes funciones auxiliares reutilizables en la lógica del *frontend*.
- *Assets*: Corresponde a las imágenes, hojas de estilo, iconos o cualquier otro tipo de archivo relativo a la estética de la aplicación.

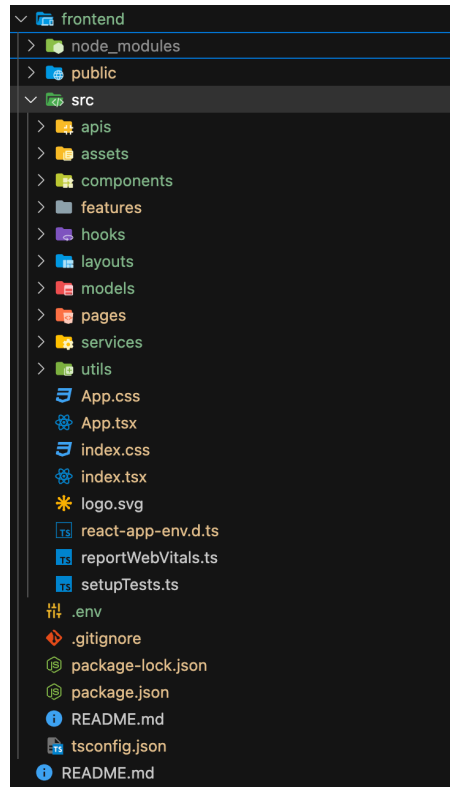


Ilustración 34 Estructura del proyecto de frontend

6.3. Diseño de la parte del servidor (backend)

6.3.1. Tecnología

La tecnología empleada para desarrollar el servidor de la aplicación no se ha cambiado respecto al prototipo G-MAC, ya que su soporte se ha mantenido durante los años, pero sí se ha actualizado a las últimas versiones disponibles. En este caso, actualizar sí que era viable porque se trata de un proyecto mucho menos complejo y unas tecnologías de aprendizaje rápido. Además, son tecnologías perfectamente válidas para cumplir con los requisitos.

Seguidamente, se listan las tres tecnologías principales que conforman el *backend*:

- *Express – NodeJS*: es un *framework* que proporciona un conjunto de características para la construcción de infraestructuras web básicas. Permite la definición de APIs con puntos de entrada, controladores y un *middleware*. Es uno de los *frameworks* más usados en la industria [33,34].
- *Sequelize*: es un ORM (*Object Relational Mapping*) para ayudar al programador en las tareas de consulta a las bases de datos relacionales. Mediante una definición de modelo de datos y configuraciones, permite obtener y modificar los recursos de una forma

declarativa a nivel de código. Por ejemplo, como «Medico.findAll()» para obtener todos los médicos. Por tanto, se trata de MDD (*Model Driven Development*). Se encarga de realizar la traducción a comandos SQL que ejecuta en la base de datos. Sin la herramienta, se necesitaría realizar manualmente las *queries*, que pueden anidar múltiples *JOIN*, complicando el desarrollo y la depuración del proyecto. También permite definir *seeders*, datos con los que poblar la base de datos, y *migrations*, cambios en el esquema de tablas [32].

- *MySQL*: base de datos relacional que permite almacenar y acceder a los datos a través de comandos SQL [42].

La estructura del proyecto tampoco ha cambiado, pero sí el contenido. Tras extender el modelo conceptual, se deben definir nuevos puntos de entrada a la *API* y configurar el ORM Sequelize para que trabaje con el nuevo modelo. Estos puntos y otros cambios relevantes se exponen en las siguientes secciones del capítulo.

6.3.2. API REST

IBM define API como un conjunto de reglas que definen cómo las aplicaciones o los dispositivos pueden conectarse y comunicarse entre sí [40]. Una API REST es una API que cumple los principios de diseño del estilo de arquitectura **REST** o *transferencia de estado representacional*. Por este motivo, las API REST a veces se conocen como *API RESTful*.

Entre los principios de REST [40], los dos más importantes son los siguientes:

- *Identificación de los recursos con URIs*: Una URI identifica a un recurso y se deben evitar los verbos para realizar operaciones sobre ellos. En cambio, se utilizan los métodos HTTP: *GET*, *PUT*, *POST*... Tampoco se deben usar nombres en singular, por consistencia, para acceder a un recurso. Por ejemplo, para obtener el paciente con id=2 del médico con id=1, el método y la URL serían: «GET /médicos/1/pacientes/2»
- *Sin estado*: Las API REST se caracterizan por ser API sin estado, lo que implica que cada solicitud debe contener todos los datos requeridos para su procesamiento, prescindiendo de la necesidad de mantener sesiones en el servidor. En otras palabras, las aplicaciones del servidor no pueden retener información vinculada a las solicitudes de los clientes.

Las URIs de una API REST se construyen con las relaciones entre los recursos. Pues en este trabajo se ha extendido el modelo conceptual en el cual se han creado y eliminado diferentes asociaciones. En consecuencia, la API REST del prototipo inicial se ha de rediseñar para adaptarse a los nuevos cambios.

Para manejar las peticiones de los clientes, se asigna un controlador a cada uno de los patrones de URIs admitidas por la API, que se encarga de procesar la petición y devolver una respuesta al cliente. Estos controladores se agrupan por los conceptos del modelo. A continuación, se documenta en una tabla los puntos de entrada a la API y las operaciones permitidas del nuevo diseño (Tabla 46), documentación importante que faltaba en el trabajo del G-MAC.

URL	GET	POST	PUT	DELETE
Controladores de Médico				
/api/medicos	✓	✓	✗	✗
/api/medicos/baja	✓	✗	✗	✗
/api/médicos/{id}	✓	✗	✓	✗
/api/medicos/{id}/pacientes/búsqueda/{termino}	✓	✗	✗	✗
/api/medicos/{id}/pacientes	✓	✓	✗	✗
/api/medicos/{id}/pacientes/graficas	✓	✗	✗	✗
Controladores de Paciente				
/api/pacientes	✓	✓	✓	✗
/api/pacientes/baja	✓	✗	✗	✗
/api/pacientes/{id}	✓	✗	✓	✗
/api/pacientes/búsqueda/{termino}	✓	✗	✗	✗
/api/pacientes/{id}/antecedentes	✓	✓	✗	✗
/api/pacientes/{id}/antecedentes/{id}	✗	✗	✓	✓
/api/pacientes/{id}/datos-antropometricos	✓	✓	✗	✗
/api/pacientes/{id}/datos-antropometricos/{id}	✗	✗	✓	✓
/api/pacientes/{id}/enfermedades	✓	✓	✗	✗
/api/pacientes/{id}/enfermedades/{id}	✗	✗	✓	✓
/api/pacientes/{id}/revisiones	✓	✓	✗	✗
/api/pacientes/{id}/revisiones/{id}	✓	✗	✓	✓
/api/pacientes/{id}/ojos	✓	✗	✗	✗
/api/pacientes/{id}/graficas/evolucion	✓	✗	✗	✗
Controladores de Enfermedad				
/api/enfermedades	✓	✗	✗	✗
Controladores de Antecedentes Médicos				
/api/antecedentes	✓	✗	✗	✗
Controladores de Hospital				
/api/hospitales	✓	✓	✗	✗
/api/hospitales/{id}	✓	✗	✗	✗
/api/hospitales/búsqueda/{termino}	✓	✗	✗	✗
/api/hospitales/{id}/medicos	✓	✗	✗	✗
Controladores de Medicamento				
/api/medicamentos	✓	✗	✗	✗
Controladores de Login				
/api/login	✓	✗	✗	✗

Tabla 46 Puntos de entrada y métodos permitidos

Las rutas de la API REST, así como la respuesta al cliente, se han construido para ajustarse a las necesidades exactas del *frontend*. Cabe destacar que algunas rutas podrían parecer redundantes como, por ejemplo, «/api/pacientes/{id}/enfermedades» existiendo «/api/pacientes/{id}/ojos/», ya que las enfermedades son de un ojo y los ojos padecen enfermedades. Pero este diseño devuelve la información directamente estructurada al *frontend* según el contexto en el que se requieran. Siguiendo el mismo ejemplo, se pueden obtener todas las enfermedades del paciente («/api/pacientes/{id}/enfermedades») o mostrar las revisiones separadas por ojo («/api/pacientes/{id}/ojos») (devuelve las revisiones además de las enfermedades).

Como un punto pendiente a mejorar, se puede observar que algunas rutas no siguen los principios REST previamente expuestos. Por ejemplo, la ruta «/api/hospitales/búsqueda/{termino}» que utiliza un verbo. Esto se debe a que estos defectos estaban en el prototipo inicial y se han priorizado otros requisitos de la aplicación.

6.3.3. SequelizeORM

Aunque la herramienta es muy potente para la comunicación con la base de datos mediante modelos, la definición de estos es por código Javascript y se debe realizar una tarea más manual en algunos casos específicos. Aun así, sigue siendo más ágil que sin la herramienta. Pero lo ideal sería que Sequelize permitiera poder generar el esqueleto de código necesario a partir de un modelado gráfico. Definitivamente, sería una herramienta totalmente *low-code*.

Los casos específicos en los que se han encontrado dificultades han sido aquellos en los que incluyen clases asociación. Pues Sequelize genera automáticamente las tablas puente entre las asociaciones de muchos a muchos, pero no lo hace si estas incluyen algunos atributos como ocurre en las clases asociación añadidas en la extensión. En consecuencia, se ha tenido que diseñar el **esquema relacional de la base de datos** manualmente (Ilustración 35). Cabe recordar que el primer modelo no tenía clases asociación.

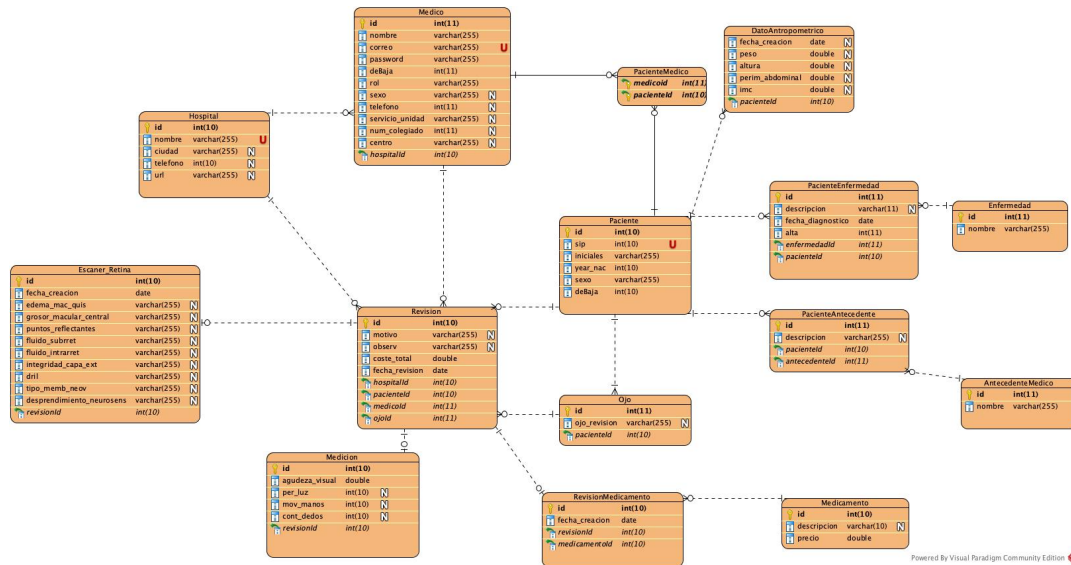


Ilustración 35 AGC-MAC Esquema de la base de datos relacional

Uno de los requisitos de la nueva versión es construir una aplicación robusta frente a la entrada incorrecta de datos. Para ello, las **validaciones** no solo se han realizado en el *frontend*, sino también en el *backend*. Pues entre otras características, Sequelize ofrece las validaciones a nivel de modelo. Estas validaciones se comprueban automáticamente antes de modificar el estado un recurso. Simplemente, se definen en los modelos de Sequelize. Por ejemplo, el *backend* del AGC-MAC comprueba si el atributo *email* (texto) tiene el formato de un *email* o que el número SIP ha de ser único. Si una validación falla, se devuelve un código HTTP de error con un mensaje con más información.

Sequelize también permite configurar las bases de datos con la que interactuar, indicando la instancia a la que se conecta, puede ser incluso de otra tecnología. En el nuevo prototipo AGC-MAC se han configurado tres bases de datos: de **producción**, de **desarrollo** y de **testing**. Con estas tres bases de datos independientes se puede desarrollar en las diferentes fases del *software* de una manera mucho más práctica. De lo contrario, los datos se mezclarían y puede entorpecer el desarrollo.

6.3.4. Testing

Realizar pruebas sobre un *software* es una parte clave para lograr un producto maduro. En este trabajo se han añadido **tests de integración** al *backend*, que no tenía el G-MAC, para no solo encontrar errores, sino también dirigir el desarrollo (*Test Driven Devolpment*). Se han utilizado los paquetes *supertest*, para las llamadas la API, y *chai*, para evaluar aserciones. También se ha poblado la base de datos de *test* con los *seeders* de Sequelize.

No hay un nivel de exhaustividad de los *tests* mejor o peor, depende de la estrategia y necesidades del equipo de desarrollo. Algunos equipos pueden optar por realizar pocos *tests* escritos y encontrar errores con el uso habitual del software, o con pruebas de usuario, ya que los *tests* no garantizan la ausencia de errores. De hecho, los errores marginales no se suelen encontrar con los casos de *testing*.

En este proyecto se han escrito dos casos de *test* de integración por cada uno de los puntos de entrada de la API REST. El primer caso es del tipo positivo, es decir, la petición a la API con datos de entrada válidos espera una respuesta positiva (códigos de estado HTTP 200, 201..). La función del segundo caso prueba una solicitud con la autorización a la API con un token inválido (vacío), esperando como respuesta un código HTTP 401. En los dos casos también se comprueba que la estructura de la respuesta es la esperada. Por ejemplo, al obtener un paciente concreto, que sea en formato JSON, tipo objeto y que contenga las propiedades: iniciales, SIP, sexo, etc. En la Ilustración 36 se puede observar el resultado de ejecutar los tests. Cabe recordar que no se debe confiar en que la aplicación funciona correctamente tras los *tests*, sino que no ha encontrado ningún error.

```
✓ should not be authorized to get a Revision of a Paciente
✓ should update a Revision of a Paciente
✓ should not be authorized to update a Revision of a Paciente
✓ shoul delete a Revision of a Paciente
✓ should not be authorized to delete a Revision of a Paciente
✓ should get all Ojos of a Paciente
✓ should not be authorized to get all Ojos of a Paciente
✓ should get Graficas Evolución of a Paciente

81 passing (225ms)
```

Ilustración 36 Resultado de los tests de integración del backend

6.3.5. Seguridad

Finalmente, un apartado importante pero menor en el desarrollo del AGC-MAC, es el de la **seguridad**. La única medida de seguridad activa que tenía el prototipo inicial era el uso del token para autorizar las llamadas a la API. Este token se añade al final de la URI como un parámetro que el *backend* lee. Los tokens de autorización nunca deben ir en la propia URI por temas de seguridad, en cambio, debe añadirse al encabezado específico de HTTP, *Authorization*. Aun así, esta vulnerabilidad no se ha corregido, pero queda como una tarea futura. Otras mejoras en seguridad pueden ser *hashear* las contraseñas y aplicar los principios de *OAuth*.

En cambio, una vulnerabilidad encontrada que sí se ha corregido es la exposición de la contraseña del usuario. Inicialmente, cuando se solicitaba el recurso de un médico al *backend*, este devolvía toda la información del objeto, incluido la contraseña. Pudiendo cualquier usuario conocer la contraseña del resto de usuarios. Este problema se ha corregido añadiendo un *scope*, de Sequelize, que se excluye la contraseña por defecto en cualquier obtención del modelo de la base de datos.

Capítulo 7

Despliegue

Una vez implementados los requisitos de la aplicación, es importante probarla en un entorno de producción con el **despliegue** en un servidor. Hasta esta fase, el proyecto se ha ido desarrollando y probando en local. Pero es muy posible que una aplicación funcione localmente y, en cambio, ocasione problemas cuando se monte en un servidor para su utilización. Es conveniente realizar despliegues tras una serie de implementaciones importantes. En este capítulo se comentan los aspectos técnicos del despliegue y los problemas encontrados.

7.1. Despliegue del prototipo AGC-MAC

La aplicación AGC-MAC se ha desplegado en los servidores de PROS/UPV (<http://genomics-hub.pros.dsic.upv.es/>) y en el momento de redacción de este TFM es accesible a través del puerto 4000. El prototipo G-MAC también estaba desplegado, pero desde el primer momento de este trabajo que no funcionaba. Tampoco se documentó como se desplegó. Por tanto, en este capítulo no se puede hacer una comparación del despliegue.

Sin embargo, se van a exponer cómo se ha llevado a cabo en el AGC-MAC y qué problemas se han ido corrigiendo a partir de entonces. En primer lugar, la forma en la que corre aplicación entera (frontend, backend y base de datos) en la fase de desarrollo es totalmente diferente a como lo hace en producción. Durante el desarrollo, el *frontend* corre en un su propio servicio local con un puerto específico y de igual manera, el *backend*, que publica la API. Es decir, se accede al servicio del *frontend* para obtener la web y entonces, se ataca al servidor. En cambio, en producción, es el *backend* (aplicación *express*) el que devuelve los archivos *web* (HTML, CSS, *scripts*...) y, además, publica la API. Para ello, se ha de generar la versión compilada del *frontend* y añadirla a la estructura del *backend*. También se han de gestionar las rutas para que no haya conflictos entre devolver los archivos estáticos (*web*) y solicitudes a la *API*. Es por este motivo que los puntos de entrada a la API empiezan por «/api». Cabe señalar que esta no es la única forma de desplegar una aplicación de este tipo, pero sí era la más sencilla teniendo en cuenta las circunstancias del servidor de PROS/UPV provisto.

En segundo lugar, la aplicación corre en una máquina virtual de los servidores de PROS/UPV. Estos servidores están configurados para hacer *port-forwarding*, que es una forma de redirigir las conexiones. La configuración de la máquina para el prototipo AGC-MAC es redirigir las conexiones entrantes desde el exterior al puerto 4000 al servicio local corriendo en el puerto 3000. Para entender esto de una forma más clara, en la Ilustración 37 se observa cómo queda el acceso a la aplicación una vez desplegada.

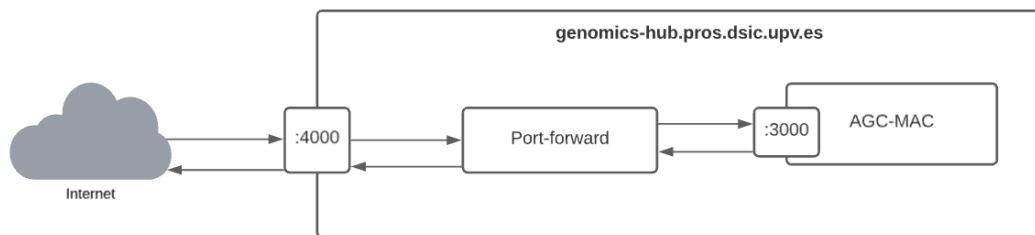


Ilustración 37 Despliegue del prototipo AGC-MAC en los servidores PROS/UPV

En tercer lugar, para mantener la aplicación disponible en todo momento se ha hecho uso del paquete PM2. PM2 es un gestor avanzado de procesos en producción. Este paquete permite arrancar y mantener una aplicación 24/7, lo que es idóneo para microservicios (no es el caso).

Por último, tras realizar el primer despliegue y un recorrido por la aplicación, se han encontrado problemas y puntos a mejorar que han sido revelados por el propio despliegue:

- *Tiempo de carga:* se ha visto incrementado el tiempo de carga de los datos en comparación a la versión local, por obvias razones. Evidentemente, los tiempos de la aplicación en la máquina local (fase de desarrollo) son mucho menores que esperar la respuesta de un servidor externo. Pero esto no se había contemplado y la interacción con la aplicación desplegada no era satisfactoria, debido a la falta de *feedback* mientras se cargaban los datos. Para solucionar el problema de usabilidad, se ha añadido una animación de carga.
- *Introducción de fechas:* en las validaciones de modelo, explicadas en la **sección 6.3.3. SequelizeORM**, se añadió una validación para que las fechas introducidas no sean posteriores a la fecha actual. Para ello, el *backend* calculaba dinámicamente el día para comprobar la satisfacción o no de la validación. Pues esto no funcionó, tras unos días con el prototipo en los servidores de la UPV, no se podían introducir fechas posteriores al día que se desplegó. Parece ser que el entorno de la aplicación no actualiza esa información. Simplemente se eliminó la validación del modelo y se valida solamente en el *frontend*.

Con los dos problemas corregidos, la aplicación corre aparentemente bien en los servidores de PROS/UPV. Con el prototipo AGC-MAC desplegado, se pueden realizar pruebas de usuario más valiosas para el desarrollo, ya que se acerca más a un entorno real (aplicación *online*).

Capítulo 8

Pruebas y validación de la solución

Las **pruebas de usabilidad** se centran en la experiencia del usuario, asegurando que la interfaz sea intuitiva y usable. La actividad de **evaluación** del ciclo de ingeniería se enfoca en comprobar que el resultado cumple con las expectativas de los *skateholders*. En este capítulo se presenta una prueba de usabilidad llevada a cabo sobre el AGC-MAC y la evaluación que podemos realizar de este artefacto implementado.

8.1. Prueba de usabilidad

8.1.1. Planteamiento

Las pruebas de usabilidad (*usability testing*) consisten proporcionar un artefacto a un conjunto de usuarios para que lo prueben mediante la supervisión de un facilitador con el objetivo de encontrar problemas de usabilidad que no se han podido identificar en las fases anteriores del desarrollo [37]. En este TFM, se ha llevado a cabo una **prueba de usabilidad presencial** del prototipo AGC-MAC desplegado (artefacto) y el supervisor ha sido el autor de este TFM (facilitador).

Además del artefacto y el facilitador, existen otros dos elementos importantes que se han de definir correctamente para poder obtener unos resultados de mayor valor:

- *Participantes*: se debe seleccionar los participantes adecuados y documentar la información que se dispone de ellos que pueda influir en los resultados. Por ejemplo, el nivel general de dominio de herramientas *web*.
- *Tareas o instrucciones del usuario*: se han de definir unas tareas o acciones con suficiente detalle que el usuario debe seguir de modo que sepa el siguiente paso con claridad y no se desvíe de la prueba. Se debe decir el qué, pero no el cómo.

Respecto a los **participantes** de la prueba, se tratan de los usuarios finales del prototipo. Es decir, médicos especialistas en oftalmología que gestionen pacientes con enfermedades de la retina mácula. Aunque se necesitaría un número relevante de participantes, en este trabajo solamente se ha podido conseguir uno, debido a la no disponibilidad del resto de expertos con los que se ha contactado. Esta participante es una oftalmóloga que usa a diario un *software* provisto por el hospital y, por tanto, entiende el dominio clínico.

En cuanto a las **instrucciones** de la prueba, se han separado en cinco casos reales:

- *Tratamiento de un paciente nuevo al sistema*
- *Tratamiento de un paciente con tratamientos previos pero nuevo para el médico.*
- *Edición de un paciente con datos incorrectos*
- *10 pacientes al cargo y visualización de la población de pacientes*
- *Cambio de contraseña*

Como se puede observar, la prueba no incluye los casos de uso de administrador, puesto que la participante no era usuaria final de este rol. En cambio, incluye todos los casos de uso de un médico no administrador especificados en la sección **Diagramas casos de uso**.

Antes de empezar la prueba, se le ha explicado en qué consiste el *software*, las funcionalidades que implementa y los objetivos de la prueba. Finalmente, se le ha introducido brevemente la aplicación (*login* y listado de pacientes). Las instrucciones que se han definido en cada situación para recorrer todos los casos reales se pueden consultar en el **Anexo II – Instrucciones de la prueba de usabilidad**.

Finalmente, tras concluir la prueba, se le ha pedido rellenar un cuestionario SUS (*System Usability Scale*) [10]. Este cuestionario permite medir la usabilidad de una forma rápida con 10 preguntas que tienen en cuenta aspectos como la facilidad de uso, aprendizaje de la herramienta, y la consistencia. En la Ilustración 38 se muestran todas las preguntas que contiene esta técnica. Para poder hacer medias o comparativas, se puede calcular un resultado numérico del cuestionario mediante una fórmula específica.

	Strongly disagree				Strongly agree
1. I think that I would like to use this system frequently					
	1	2	3	4	5
2. I found the system unnecessarily complex					
	1	2	3	4	5
3. I thought the system was easy to use					
	1	2	3	4	5
4. I think that I would need the support of a technical person to be able to use this system					
	1	2	3	4	5
5. I found the various functions in this system were well integrated					
	1	2	3	4	5
6. I thought there was too much inconsistency in this system					
	1	2	3	4	5
7. I would imagine that most people would learn to use this system very quickly					
	1	2	3	4	5
8. I found the system very cumbersome to use					
	1	2	3	4	5
9. I felt very confident using the system					
	1	2	3	4	5
10. I needed to learn a lot of things before I could get going with this system					
	1	2	3	4	5

Ilustración 38 Cuestionario Sistem Usability Scale (SUS) (J. Brook, 1995)

8.1.2. Resultados de la prueba de usabilidad

En primer lugar, se ha obtenido la máxima puntuación en el cuestionario SUS (100). Aunque podría variar el resultado debido a la subjetividad de las respuestas, definitivamente el *software*

probado ha conseguido un resultado muy positivo en términos de usabilidad. Pues durante la prueba ha habido muy pocos problemas que hayan entorpecido la experiencia del usuario, pero los que se han encontrado son los siguientes:

- *Campo de alta (sí o no) en una nueva enfermedad es confuso*: la participante no sabía a qué se refería este campo en el formulario, ya que cuando se diagnostica una enfermedad, significa que la padece y no está de alta. Cabe decir que este campo existía para la gestión de datos históricos.
- *Actualización de los datos*: la participante esperaba que el resumen de enfermedades del paciente se actualizara tras la enfermedad que se había editado. En este caso, se le indicó que refrescara la página.
- *Selección involuntaria de opciones*: los campos de formulario que consisten en una selección desplegable de opciones, si se desplegaban, se seleccionaba automáticamente una opción. Esto ocurrió en el escáner de la retina, en el que la participante quería consultar las opciones del tipo de membrana, pero no seleccionar ninguna.
- *Estructura de la información del escáner de retina*: dentro de los diferentes parámetros de un escáner de retina, hay algunos más importantes que otros. El software los ponía todos al mismo nivel y resulta en un pequeño impedimento en la lectura del escáner.
- *Otros*: otros problemas menores propios del aprendizaje de la aplicación como no encontrar ciertos elementos, aunque se podrían reducir con soluciones de diseño.

En segundo lugar, durante la prueba, la oftalmóloga participante ha considerado necesario la inclusión de ciertos campos para que la gestión de pacientes con el AGC-MAC sea más completa. Estas sugerencias son respecto a la lista de medicamentos y al nivel de detalle de los campos de los formularios, además de dar su opinión acerca del prototipo, que se tendrá en cuenta para el capítulo de **Conclusiones**.

Finalmente, aunque son resultados positivos, es importante recalcar que esta prueba por sí sola no representa una evaluación real de la usabilidad del prototipo, ya que solo consta de un participante. Además, se han encontrado pocos problemas de usabilidad, algo no habitual en las primeras pruebas de *software*. Para mejorar este aspecto, se podría replicar la prueba planteada con un número de participantes relevante.

8.2. Evaluación

Tras el desarrollo del AGC-MAC, la siguiente actividad del ciclo de ingeniería propuesto por Wieringa et. al. (2014) es la evaluación de la implementación, que consiste en comprobar cómo de exitosa ha sido la implementación, entrando en nueva iteración del ciclo que puede generar nuevos problemas de investigación.

Durante el transcurso de este TFM se han llevado a cabo una serie de reuniones con los expertos del Centro PROS con el objetivo obtener opiniones de calidad acerca del estado del trabajo. Gracias a las críticas constructivas se pueden tomar decisiones más adecuadas que ayuden a conseguir un trabajo de ingeniería de *software* de alto nivel. El Centro PROS forma parte del contexto social del marco del proyecto y, por tanto, el prototipo ha de cumplir también con sus expectativas. Aunque no se ha podido realizar una evaluación del resultado final por parte de este grupo de stakeholders, se podría predecir con sus validaciones y opiniones previas.

Durante la primera revisión, se expuso el proyecto del prototipo AGC-MAC y el cambio de tecnología a React. Se dio el visto bueno del cambio y se incidió en el modelado conceptual, que

debía tener un rol importante, lo cual consideramos que se ha cumplido. En la segunda reunión, se presentaron los avances del artefacto junto con el análisis genómico de la DMAE. En la última reunión, se vio en más detalle el uso del prototipo en un estado avanzado, el cual validaron satisfactoriamente.

Otros *skateholders* implicados son los médicos especializados en oftalmología, que al final son los que van a hacer uso de la plataforma. Para conseguir un *software* que cumpla con sus expectativas, se han realizado validaciones continuas por parte del Dr. Rubén Cabrera, con reuniones frecuentes. El Dr. Rubén Cabrera opina respecto al AGC-MAC que se ha conseguido *software* mucho más usable para el día a día en la gestión de pacientes, en comparación al G-MAC, y además provee información útil acerca la evolución de un paciente o la población de pacientes de un médico. Aparte de la opinión del Dr. Rubén Cabrera, también se ha contado con la colaboración de una oftalmóloga, también participante en la prueba de usabilidad, la cual comparte la misma opinión respecto a la usabilidad del *software*. Ambos están de acuerdo en que el AGC-MAC lo podrían utilizar actualmente como herramienta apoyo, sobre todo para analizar la evolución de un paciente y tomar decisiones más rápidas, aunque para empezar se tendría que cargar toda la base de datos de pacientes al cargo. En resumen, a falta de la opinión de más *skateholders*, se puede calificar la evaluación de la nueva versión de positiva, ya que es un prototipo que satisface las necesidades para la gestión y análisis de pacientes con enfermedades maculares. Cabe señalar que, para este grupo de *skateholders*, la selección de las tecnologías para la implementación del AGC-MAC es indiferente.

Capítulo 9

Resultados

Este capítulo se exponen los resultados del desarrollo del prototipo AGC-MAC con la representación visual en detalle. Aparte del artefacto, también se explica cómo ha impactado este trabajo en la experiencia del desarrollo desde la perspectiva de un ingeniero de *software*.

9.1. Representación visual de la solución

Para poder mostrar cual es el resultado de la extensión del G-MAC de una forma visual, se presentan diferentes capturas de pantalla que conforman las interfaces de la aplicación web desplegada.

- *Inicio de sesión*: la Ilustración 39 corresponde con la página de acceso a la aplicación o *login*. Esta interfaz corresponde con el caso de uso **CU25**. Se han añadido los logos de las entidades implicadas (UPV, VRAIN, PROS, Departamento de Salud de Orihuela y Fundación Fisabio). También se ha cambiado el nombre del logo a AGC-MAC.
- *Inicio*: en la página de inicio (Ilustración 40) se incluye el **CU40** y **C41**. Esto corresponde a la nueva funcionalidad de gráficas de la población de pacientes.
- *Pacientes*: desde esta ventana se pueden acceder a todos los casos de uso representados en el diagrama de CU de gestión de pacientes. En la Ilustración 41 se muestran **CU1**, **CU2**, **CU6**, **CU7**, **CU4**, **CU5** correspondientes al listado de pacientes, búsqueda de pacientes, creación de paciente y adición de paciente existente.
 - **Resumen (Ilustración 44)**: se muestran los datos personales y el resumen del paciente y los atajos para crear enfermedades, antecedentes y datos antropométricos (**CU8**, **CU9**, **CU23**, **CU24**, **CU10**, **CU11**).
 - **Revisiones (Ilustración 45)**: corresponde al listado de revisiones del paciente separadas por ojo con una línea temporal (**CU19** y **CU21**). Se puede ver el detalle de una revisión en concreto en una nueva ventana (Ilustración 51), que corresponde al **CU20**. Des del detalle, también se puede editar y eliminar la revisión (**CU22**).
 - **Gráficas (Ilustración 46)**: esta pestaña incluye el **CU12**, que corresponde a la visualización de la gráfica de la evolución del paciente.
 - **Enfermedades (Ilustración 47)**: se muestra el listado de enfermedades (**CU13**). Directamente desde esta vista, se puede gestionar cualquier enfermedad del paciente (crear, editar y eliminar), correspondiente al **CU14**.
 - **Antecedentes Médicos (Ilustración 49)**: se muestra el listado de antecedentes médicos (**CU15**). De igual modo que las enfermedades, se pueden gestionar los antecedentes médicos (crear, editar y eliminar) directamente sin navegar a otra ventana (**CU6**).
 - **Datos antropométricos (Ilustración 50)**: se muestra el listado de datos antropométricos (**CU17**) y posibilidad de realizar acciones de gestión (crear, editar y eliminar) (**CU18**).

- *Médicos*: la ventana de gestión de médicos (Ilustración 54) solo es visible para el administrador y desde la cual se pueden acceder a todos los casos de uso representados en el diagrama de CU para la gestión de médicos. En la primera página se observa los **CU33** y **CU34** correspondientes a la lista de médicos y creación de médico. Se puede navegar al detalle de un médico (Ilustración 54) y a la página de médicos inhabilitados (Ilustración 53).
- *Hospitales*: ventana tampoco visible para el rol médico (Ilustración 55), desde la cual se pueden acceder al **CU28**, **CU31** y **CU34**, correspondientes a buscar, crear y listar hospitales. Se puede navegar al detalle del hospital (Ilustración 56) para ver la información del hospital y listar los médicos asociados.

Cabe destacar que algunas interfaces son muy similares que el prototipo G-MAC, ya que se ha decidido mantener los mismos elementos básicos y la distribución de ellos y, por tanto, tampoco se han diseñado *mockups* para estos casos. Pero no significa que no hayan sido actualizadas, existen algunos cambios menores que no han precisado de *mockup*:

- *Paginación de resultados*: las listas de objetos como, por ejemplo, de médicos se han paginado para no saturar la pantalla con demasiada información.
- *Diálogos de confirmación*: en las eliminaciones de objetos aparece un diálogo para confirmar la acción (Ilustración 48).
- *Coherencia de términos*: los mismos conceptos en la aplicación deben usar los mismos términos. En este caso, el concepto de inhabilitación, que anteriormente utilizaba también el término de baja. Ahora solo aparece como inhabilitar o habilitar.



Ilustración 39 AGC-MAC Login o inicio de sesión



Ilustración 40 AGC-MAC Ventana Gráficas de la población de pacientes

Pacientes

Búsqueda de pacientes

SIP del paciente

Pacientes inhabilitados

+ Nuevo paciente + Añadir paciente existente

Listado de pacientes al cargo

Iniciales	SIP	Edad
A. A.	1	56 años

1-1 of 1 < >

Ilustración 41 AGC-MAC Ventana Pacientes

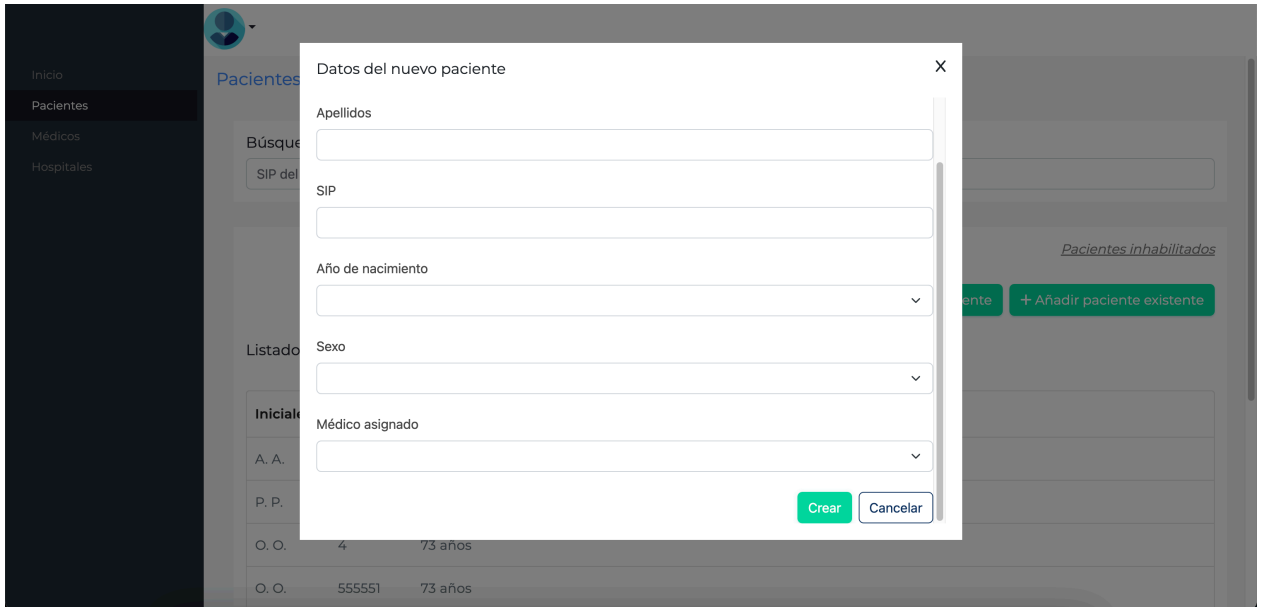


Ilustración 42 AGC-MAC Formulario de creación de un paciente

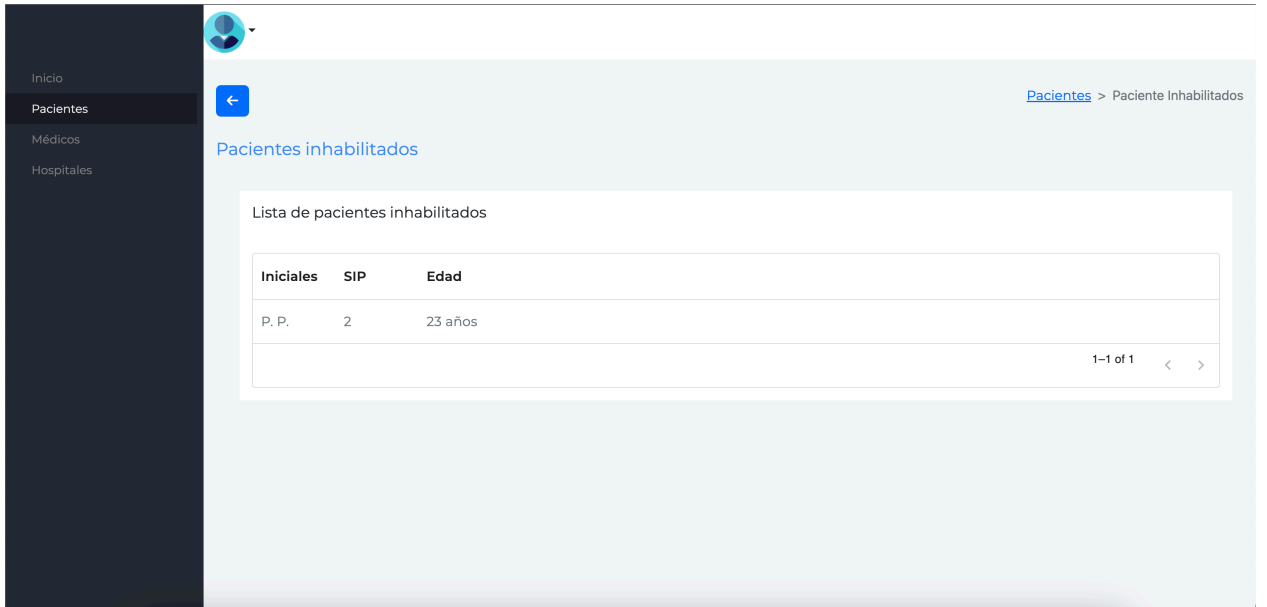


Ilustración 43 AGC-MAC Ventana Pacientes inhabilitados

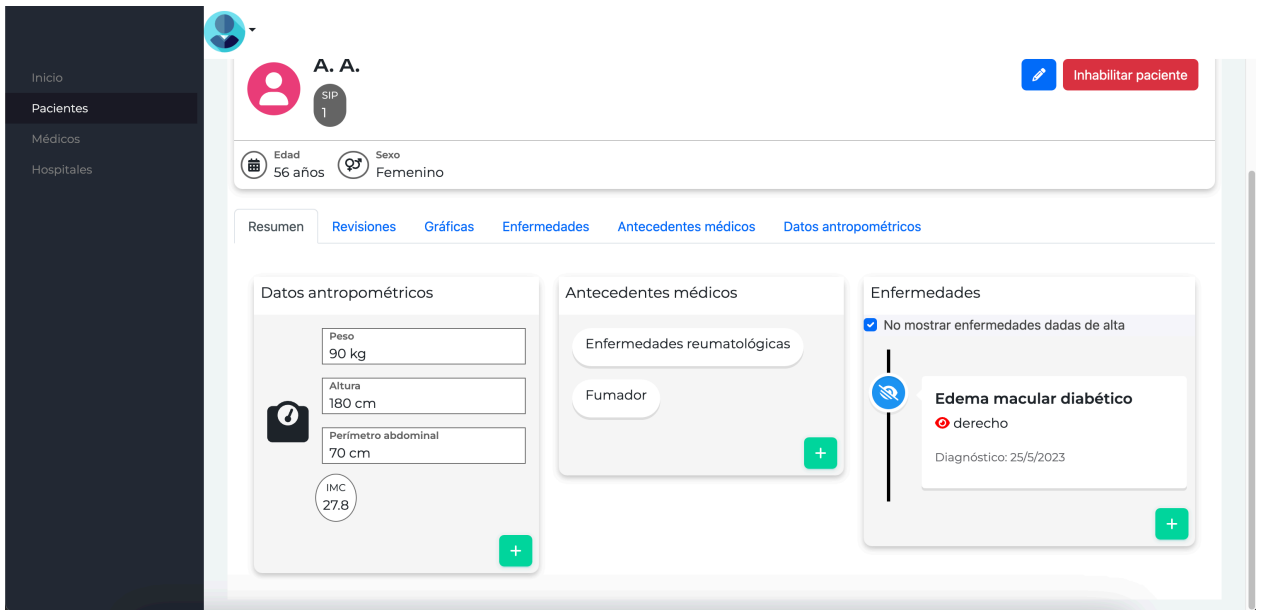


Ilustración 44 AGC-MAC Ventana Resumen de paciente

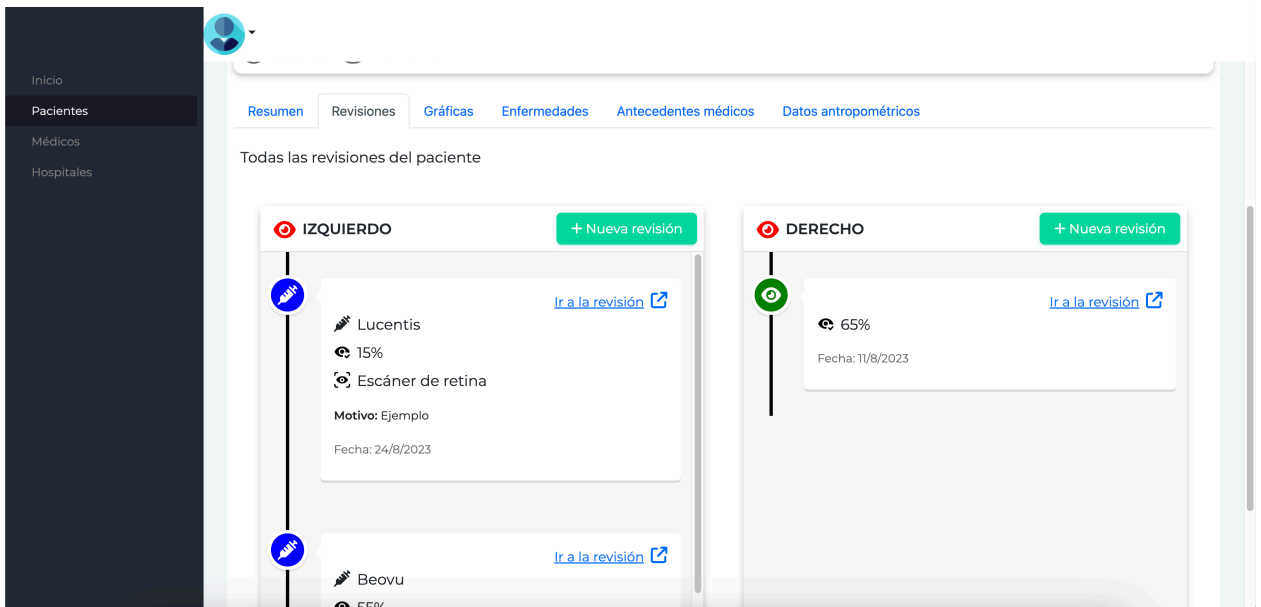


Ilustración 45 AGC-MAC Ventana Revisiones de paciente



Ilustración 46 AGC-MAC Ventana Gráficas de paciente

Paciente

A. A. SIP 1 Inhabilitar paciente

Edad: 56 años Sexo: Femenino

Enfermedades

+ Nueva enfermedad

Nombre	Descripción	Ojo afectado	Fecha de diagnóstico	Alta	Acciones
Edema macular diabético	ooo	DERECHO	25/5/2023	No	✎ ✖

1-1 of 1 < >

Ilustración 47 AGC-MAC Ventana Enfermedades de paciente

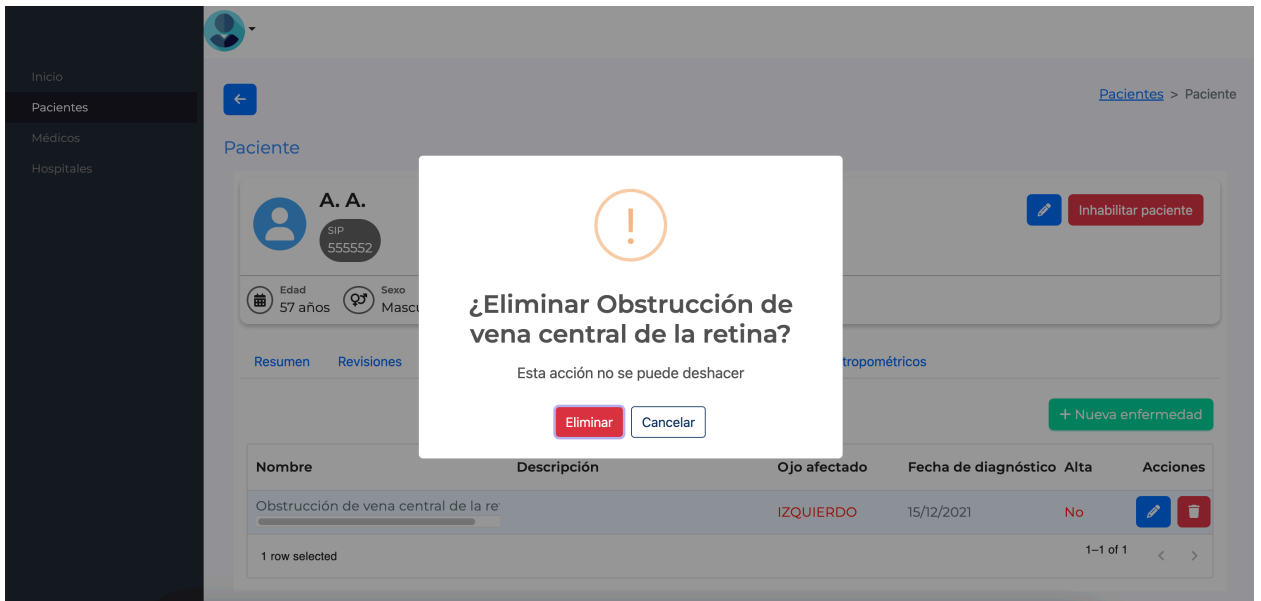


Ilustración 48 AGC-MAC Eliminación de una enfermedad

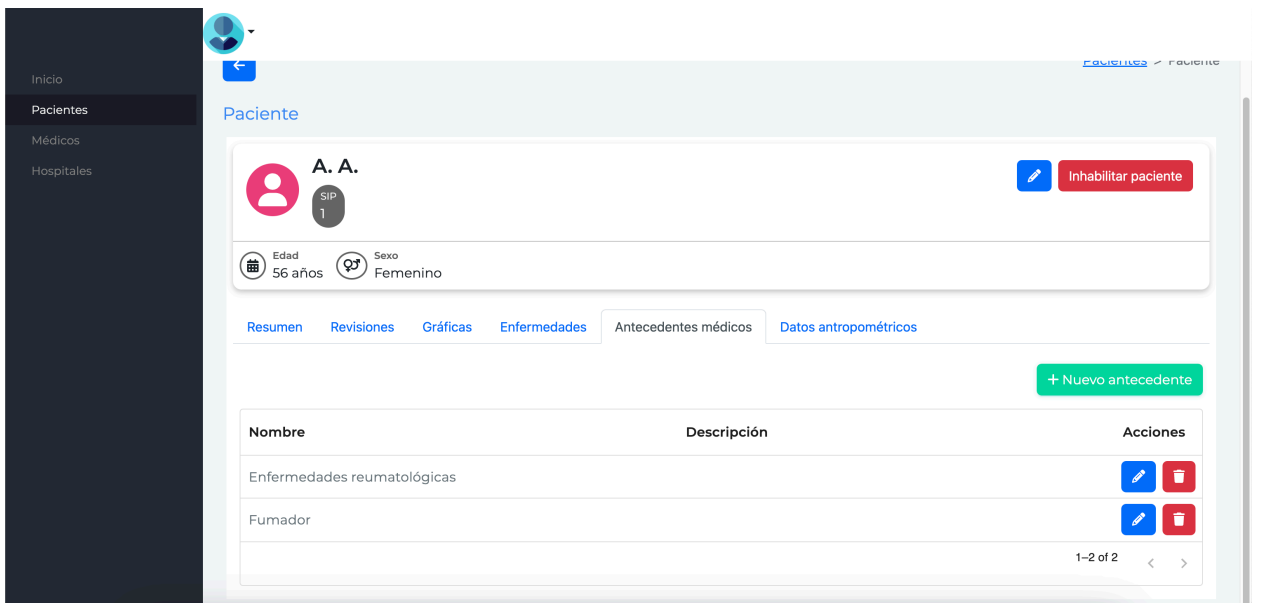


Ilustración 49 AGC-MAC Ventana Antecedentes médicos de paciente

Inicio
Pacientes
Médicos
Hospitales

Pacientes > Paciente

Paciente

A. A. SIP 1 Inhabilitar paciente

Edad 56 años Sexo Femenino

Resumen Revisiones Gráficas Enfermedades Antecedentes médicos Datos antropométricos

+ Nuevo dato

Fecha	Peso	Altura	Perímetro abdominal	IMC	Acciones
22/8/2023	90	180	70	27.8	

1-1 of 1 < >

Ilustración 50 AGC-MAC Ventana Datos antropométricos de paciente

Inicio
Pacientes
Médicos
Hospitales

Pacientes > Paciente > Revisiones > Revisión

Revisión

Datos de la revisión

Médico Testing Testing Testing Hospital Hospital Virgen de los Lirios Fecha 24/8/2023

Coste total 500 €

Motivo
Ejemplo

Observaciones
Hola

Datos del paciente

Ilustración 51 AGC-MAC Ventana Revisión

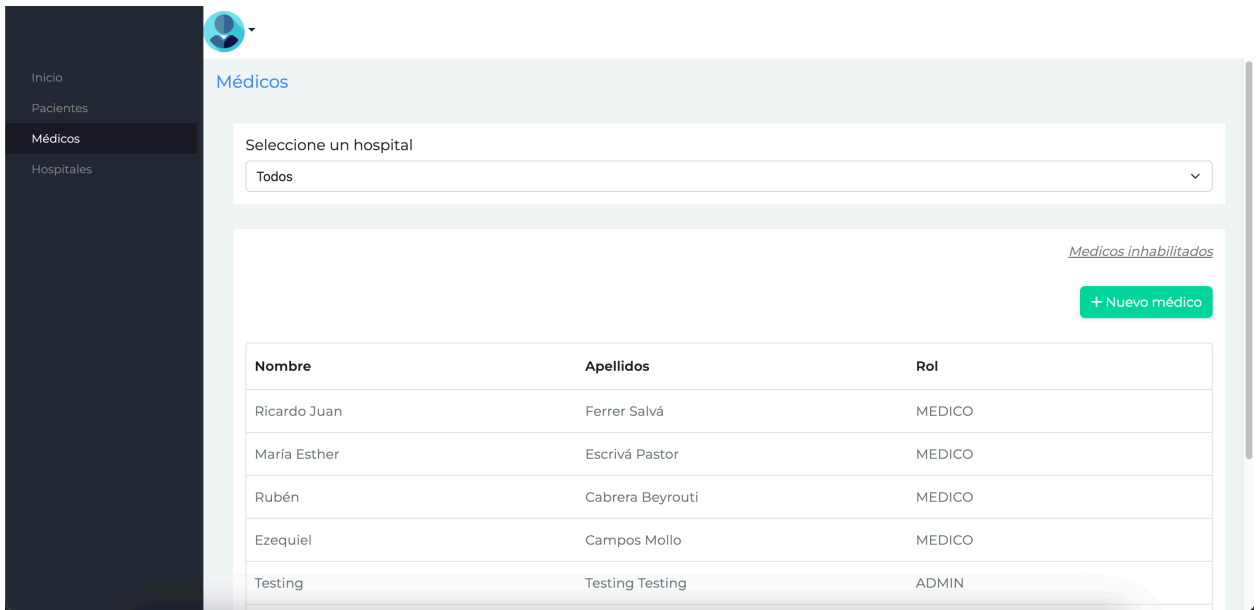


Ilustración 52 AGC-MAC Ventana médicos

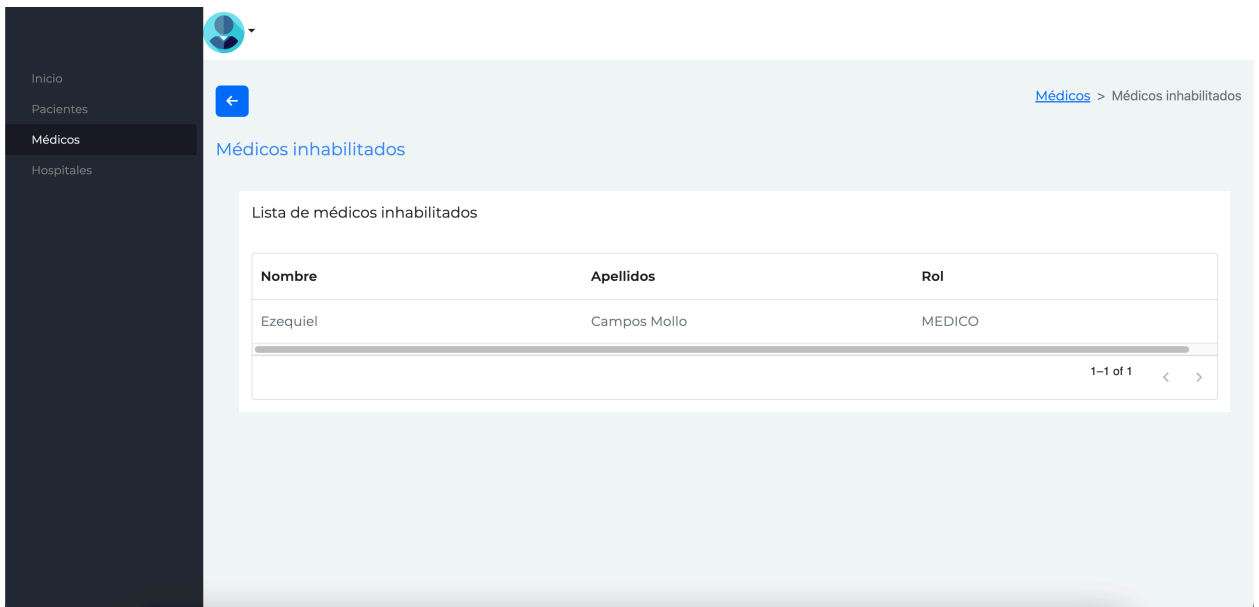


Ilustración 53 AGC-MAC Ventana Médicos inhabilitados

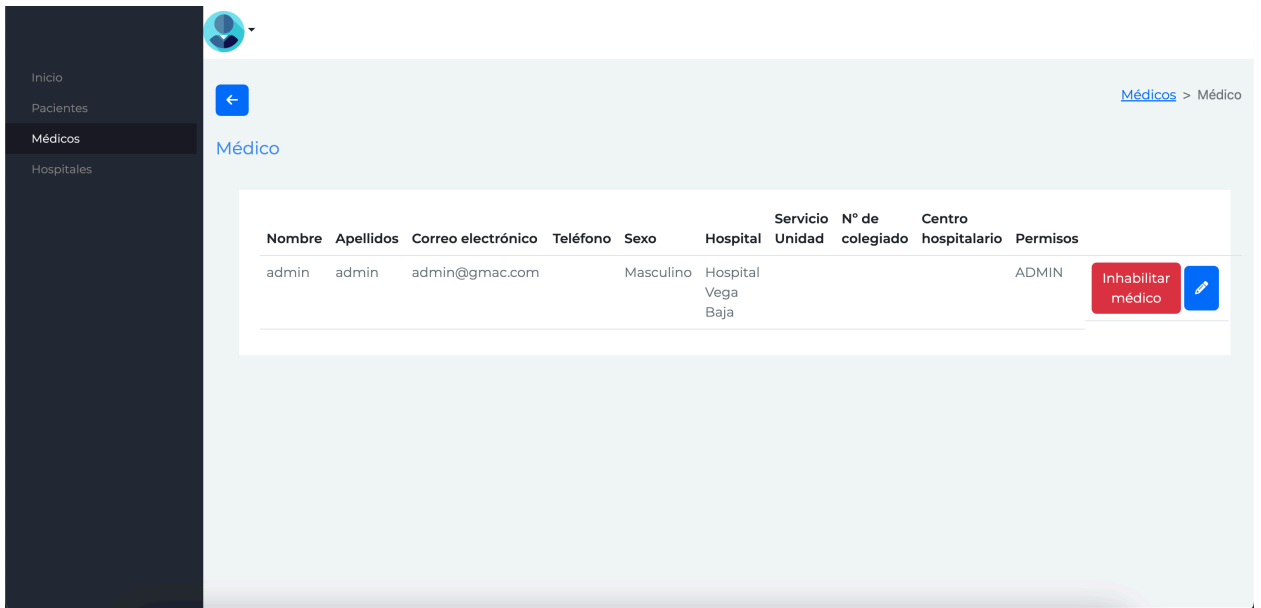


Ilustración 54 AGC-MAC Ventana Médico

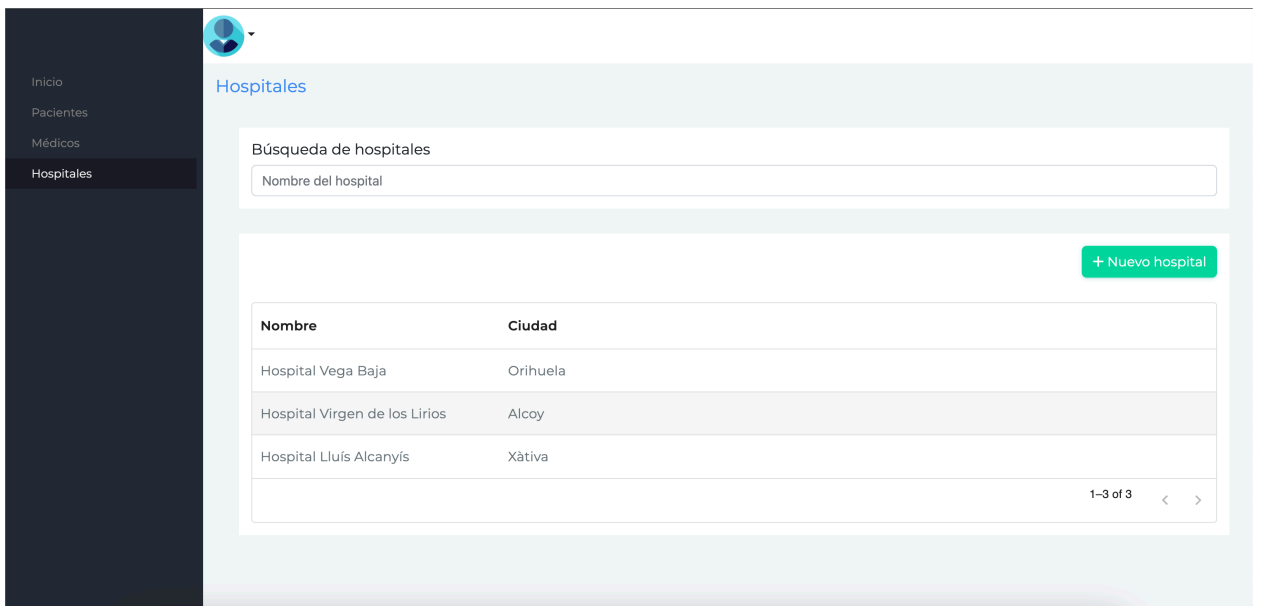


Ilustración 55 AGC-MAC Ventana Hospitales

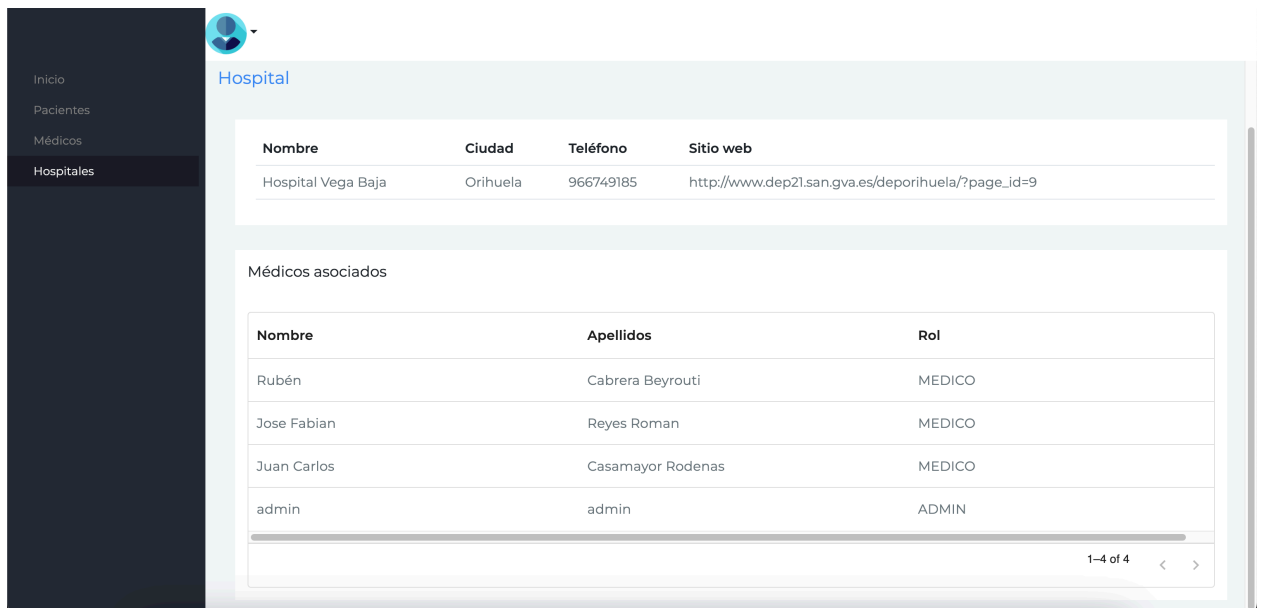


Ilustración 56 AGC-MAC Ventana Hospital

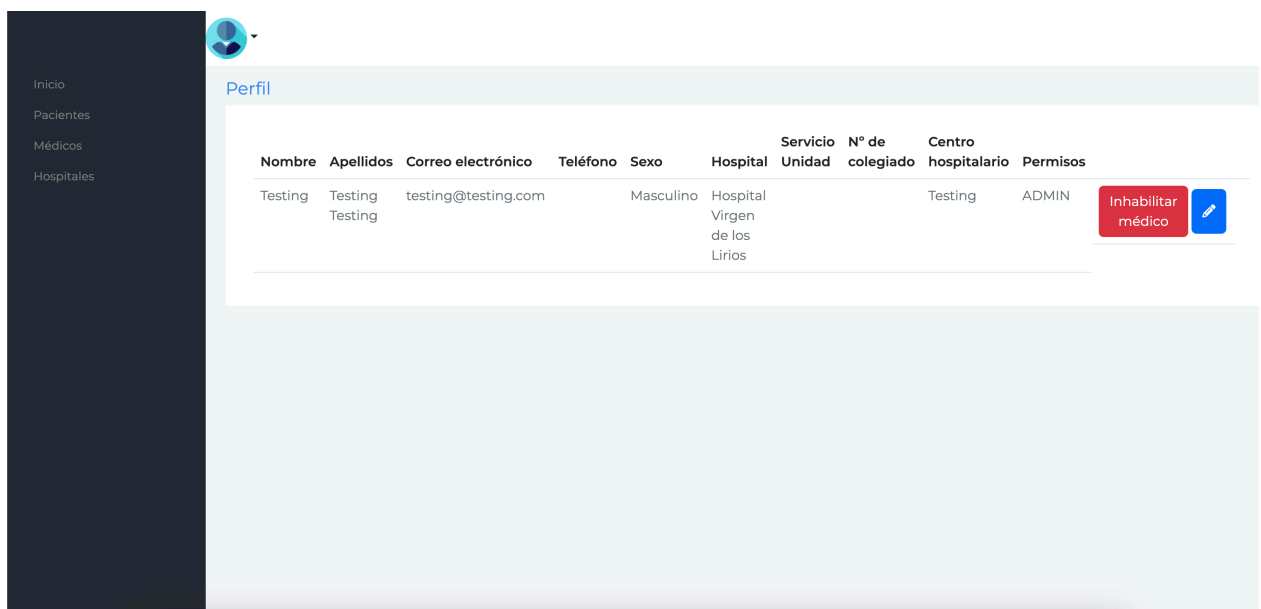


Ilustración 57 AGC-MAC Ventana Perfil

9.2. Experiencia en el desarrollo

Con el desarrollo del AGC-MAC se ha podido aplicar y extender los conocimientos de la ingeniería del *software*, así como el de ciertas tecnologías. Extender un *software* obliga a entender cómo funciona cada una de sus partes para poder tomar decisiones más fundamentadas sobre nuevos diseños. También resalta la importancia de construir un *software* profesional, entendible y mantenible, algo que se ha tenido en cuenta durante todo el desarrollo.

Entre otras cosas, la separación de *frontend* y *backend* en el G-MAC ha permitido cambiar Angular por React, aunque no ha sido una tarea rápida. Definitivamente, migrar una aplicación web de cero ha sido todo un reto positivo en el que se ha aprendido más acerca de React y las fortalezas y debilidades de cada uno. También se ha aprendido en tecnologías de la parte de servidor, ya que partir un *backend* fuerte evita construir las bases de un proyecto entero. En

cambio, se han estudiado las tecnologías usadas y se ha extendido según los requisitos. Por ejemplo, los ORM, Sequelize en este caso, ha sido un gran descubrimiento para la comunicación con la base de datos mediante modelos.

Sin embargo, la parte más interesante ha sido realizar tareas de investigación del problema, la representación de este y la especificación de requisitos. Es la primera vez que se hemos puesto estas actividades en práctica en un proyecto real y ha sido una muy buena experiencia para continuar con ellas. Aunque se han encontrado dificultades en el proceso, poder desarrollar un *software* más usable, con funcionalidades de análisis y con un nuevo modelo de datos orientado a agregación de datos, ha sido servido para mejorar como ingeniero de *software*.

Aparte de la experiencia técnica, hay otros aspectos que también se han visto reforzados. En primer lugar, el mayor conocimiento del ámbito clínico y de las enfermedades maculares, los problemas con los que se encuentran los especialistas para tratarlas y las implicaciones que tiene en los pacientes. En segundo lugar, tener reuniones frecuentes con expertos en el tema y trabajar de una manera más profesional en todos los aspectos ha sido enriquecedor para no solo para mejorar como ingeniero de *software*, sino también para desarrollar habilidades importantes que van más allá de la ingeniería.

Capítulo 10

Conclusiones

En este capítulo se presentan las conclusiones que se pueden extraer del presente Trabajo de Fin de Máster. En primer lugar, en la sección **10.1.** se explican las contribuciones principales de este trabajo en base a los objetivos establecidos en la sección Objetivos. Para finalizar la memoria, en la sección **10.2.** se plantean unas posibles líneas de trabajo futuro.

10.1. Contribuciones principales

Este Trabajo de Fin de Máster se definió con el objetivo de extender y madurar una solución *software* (G-MAC) para la gestión y análisis de pacientes con enfermedades maculares. Un *software* que sea usable por los médicos especialistas en oftalmología y genere conocimiento relevante que pueda influir en la toma de decisiones de tratamientos más personalizados y eficientes. Para ello, la aplicabilidad clínica de los estudios genómicos de este tipo de enfermedades se planteó como el camino a seguir.

Tras el desarrollo de este TFM, se puede concluir si ha contribuido a solucionar el problema y en qué manera. Se ha desarrollado un prototipo de Sistema de Información (SI), el AGC-MAC, mucho más usable para la gestión de pacientes, que además puede aportar una utilidad real con un mejor y rápido entendimiento de la evolución y población de pacientes. De este modo, i) permitiendo a los médicos tomar decisiones más rápidas y, en consecuencia, ahorrar tiempo en las consultas y, ii) posibilidad de realizar estudios epidemiológicos sobre este tipo de enfermedades. Además, se ha planteado una funcionalidad clínico-genómica que pretende personalizar los tratamientos y así, intentar reducir el elevado gasto en los fármacos para este tipo de enfermedades.

Para llevarlo a cabo, se ha puesto el modelado conceptual en el foco como base para desarrollar las soluciones, con la extensión del Modelo Conceptual de la Patología Macular que ha refinado el dominio y ha potenciado el análisis de los datos. También se ha unido el MCPM con el Modelo Conceptual del Genoma que sirve como base para desarrollar las funcionalidades clínico-genómicas, así como se ha realizado una labor de investigación de la DMAE para entender mejor el componente genético de este tipo de enfermedades.

También se ha mejorado la implementación del prototipo i) actualizándola a las últimas tecnologías de la industria como React, ii) aumentando la robustez frente a errores con validaciones de modelo y test de integración de la API, y iii) cambio el esquema relacional de base de datos a la agregación de datos, y iv) detectando y corrigiendo vulnerabilidades de seguridad.

10.2. Trabajo futuro

Con el final de este Trabajo de Fin de Máster, se finaliza la versión **2.0** del *software* para la gestión y análisis de pacientes con enfermedades maculares y establece una base para la inclusión de la genómica en el ámbito clínico-genómico. A partir de este punto, se abren nuevas líneas de trabajo, a corto y largo plazo.

- *Extensión del Modelo Conceptual de la Patología Macular (MCPM)*: aunque se ha realizado una extensión del primer modelo, se podría extender aún más con especificaciones de las enfermedades, ya que cada una tiene sus particularidades. También se podrían añadir los planes de tratamiento que se definen para tratar un paciente y los nuevos métodos de diagnóstico o seguimiento que surjan. En definitiva, investigar más sobre el dominio de las enfermedades maculares y representarlo en el MCPM.
- *Realizar estudios epidemiológicos*: actualmente el *software* permite la visualización de gráficas integrando las variables de enfermedad, sexo, edad y antecedentes médicos. Con esta funcionalidad, se podría registrar el historial completo de un número relevantes de pacientes y realizar estudios epidemiológicos sobre las enfermedades maculares.
- *Desarrollo la versión 3.0 de la solución*: en el punto en que se encuentra el AGC-MAC, es necesario realizar más reuniones con expertos para tener un *feedback* más completo que ayude no solo a corregir los posibles problemas de uso, sino también a conocer aún más el dominio para el desarrollo de una versión **3.0**. Una nueva versión que ofrezca funcionalidades que no han sido posible en este TFM por limitaciones temporales como: visualización de la imagen del escáner, gestión de medicamentos, gráficas de todo tipo como, por ejemplo, el gasto hospitalario por enfermedad, y la internacionalización del *software*. Aunque para dar un salto importante, una futura versión podría, finalmente, implementar funcionalidades clínico-genómicas o incluso, integrar Inteligencia Artificial Explicable (IAE), siguiendo la línea del proyecto de innovación.
- *Transferencia del producto software*: una vez demostrada su utilidad en la práctica asistencial, se procedería a la difusión del producto en congresos para la venta e implantación del *software* en cualquier hospital, público o privado, que requiera de una licencia. Para llegar a este punto, ha de ser un producto muy maduro y con los mercados CE.
- *Publicación del trabajo*: los resultados obtenidos en este trabajo de investigación *software* se pretende publicarlos en alguna de las siguientes conferencias y/o congresos: CAiSE, ER, RCIS o CMLS@ER.

Bibliografía

1. Marco Palomares, A. (2020). Desarrollo Dirigido por Modelos para el Análisis y Gestión de los Datos asociados a la Patología Macular. <http://hdl.handle.net/10251/151886>
2. Román, J. F. R., Palacio, A. L., S, A. G., Beyrouiti, R. C., & Pastor, O. (2022, September 16). Integration of clinical and genomic data to enhance precision medicine: a case of study applied to the retina-macula. *Software and Systems Modeling*. <https://doi.org/s10270-022-01039-4>
3. Reyes Román, J. F., Marco Palomares, A., García Simón, A., & Pastor, O. (2021). A Model-Based Application for the Effective and Efficient Management of Data Associated with Retina-Macula Pathology. *Enterprise, Business-Process and Information Systems Modeling*, 366–379. https://doi.org/10.1007/978-3-030-79186-5_24
4. Fleckenstein, M., Keenan, T. D. L., Guymer, R. H., Chakravarthy, U., Schmitz-Valckenberg, S., Klaver, C. C., Wong, W. T., & Chew, E. Y. (2021, May 6). Age-related macular degeneration. *Nature Reviews Disease Primers*, 7(1). <https://doi.org/10.1038/s41572-021-00265-2>
5. Casten, R. J., & Rovner, B. W. (2013, May). Update on depression and age-related macular degeneration. *Current Opinion in Ophthalmology*, 24(3), 239–243. <https://doi.org/10.1097/icu.0b013e32835f8e55>
6. Khachigian, L. M., Liew, G., Teo, K. Y. C., Wong, T. Y., & Mitchell, P. (2023, February 21). Emerging therapeutic strategies for unmet need in neovascular age-related macular degeneration. *Journal of Translational Medicine*, 21(1). <https://doi.org/10.1186/s12967-023-03937-7>
7. Thomas, C. J., Mirza, R. G., & Gill, M. K. (2021, May). Age-Related Macular Degeneration. *Medical Clinics of North America*, 105(3), 473–491. <https://doi.org/10.1016/j.mcna.2021.01.003>
8. Wong, W. L., Su, X., Li, X., Cheung, C. M. G., Klein, R., Cheng, C. Y., & Wong, T. Y. (2014, February). Global prevalence of age-related macular degeneration and disease burden projection for 2020 and 2040: a systematic review and meta-analysis. *The Lancet Global Health*, 2(2), e106–e116. [https://doi.org/10.1016/s2214-109x\(13\)70145-1](https://doi.org/10.1016/s2214-109x(13)70145-1)
9. JB. (2017). Updates on the Epidemiology of Age-Related Macular Degeneration. *Asia-Pacific Journal of Ophthalmology*. <https://doi.org/10.22608/apo.2017251>
10. Brooke. (1996). SUS -- a quick and dirty usability scale. In *Usability Evaluation in Industry* (pp. 189–194). Taylor & Francis Editors: Patrick W. Jordan and Bruce Thomas and Bernard A. Weerdmeester and Ian L. McClelland.
11. Steffen Schmitz-Valckenberg, Moussa A. Zouache, Gregory S. Hageman, Monika Fleckenstein. (2022). From Genes, Proteins, and Clinical Manifestation: Why Do We Need to Better Understand Age-Related Macular Degeneration? *American Academy of Ophthalmology*.

12. Chen, Y., Bedell, M., & Zhang, K. (2010, October 1). Age-related Macular Degeneration: Genetic and Environmental Factors of Disease. *Molecular Interventions*. <https://doi.org/10.1124/mi.10.5.4>
13. Sociedad Español de Retina y Vitreo. <https://serv.es/>
14. Asociación Mácula-Retina. <https://www.macula-retina.es/>
15. American Academy of Ophthalmology. <https://www.aao.org/>
16. Costa Sánchez, M. (2021). Diseño y Desarrollo de una Plataforma para la Gestión de Datos Genómicos: Oráculo Genómico de Delfos. Universitat Politècnica de València. <http://hdl.handle.net/10251/173220>
17. García Simón, A. (2022). Understanding the Code of Life: Holistic Conceptual Modeling of the Genome [Tesis doctoral]. Universitat Politècnica de València. <https://doi.org/10.4995/Thesis/10251/191432>
18. Reyes Román, JF. (2018). Diseño y desarrollo de un Sistema de Información Genómica basado en un modelo conceptual holístico del genoma humano [Tesis doctoral no publicada]. Universitat Politècnica de València. <https://doi.org/10.4995/Thesis/10251/99565>
19. Reyes Román, J. F., Marco Palomares, A., García Simón, A., & Pastor, O. (2021). A Model-Based Application for the Effective and Efficient Management of Data Associated with Retina-Macula Pathology. *Enterprise, Business-Process and Information Systems Modeling*, 366–379. https://doi.org/10.1007/978-3-030-79186-5_24
20. Li, J. Q., Welchowski, T., Schmid, M., Mauschitz, M. M., Holz, F. G., & Finger, R. P. (2019, November 11). Prevalence and incidence of age-related macular degeneration in Europe: a systematic review and meta-analysis. *British Journal of Ophthalmology*; *BMJ*. <https://doi.org/10.1136/bjophthalmol-2019-314422>
21. Wieringa. (n.d.). *Design Science Methodology for Information Systems and Software Engineering*. Springer
22. Pameijer, E. M., Heus, P., Damen, J. A., Spijker, R., Hooft, L., Ringens, P. J., Imhof, S. M., & Van Leeuwen, R. (2022, June 13). What did we learn in 35 years of research on nutrition and supplements for age-related macular degeneration: a systematic review. *Acta Ophthalmologica*; Wiley. <https://doi.org/10.1111/aos.15191>
23. Overview of DERBY and OAKS Studies. (2023, April 14). *AJMC*. <https://www.ajmc.com/view/overview-of-derby-and-oaks-studies>
24. Gene Therapy for Age-Related Macular Degeneration. (n.d.). Haiku. <https://www.ndcn.ox.ac.uk/research/clinical-ophthalmology-research-group/trials/amd-gene-therapy>
25. Klein, R. J., Zeiss, C., Chew, E. Y., Tsai, J. Y., Sackler, R. S., Haynes, C., Henning, A. K., SanGiovanni, J. P., Mane, S. M., Mayne, S. T., Bracken, M. B., Ferris, F. L., Ott, J., Barnstable, C., & Hoh, J. (2005, April 15). Complement Factor H Polymorphism in Age-Related Macular Degeneration. *Science*, 308(5720), 385–389. <https://doi.org/10.1126/science.1109557>

26. Esparza-Gordillo, J., Soria, J. M., Buil, A., Almasy, L., Blangero, J., Fontcuberta, J., & Rodríguez de Córdoba, S. (2004, May 1). Genetic and environmental factors influencing the human factor H plasma levels. *Immunogenetics*, 56(2), 77–82. <https://doi.org/10.1007/s00251-004-0660-7>
27. León Palacio, A. (2019). SILE: A Method for the Efficient Management of Smart Genomic Information [Tesis doctoral]. Universitat Politècnica de València. <https://doi.org/10.4995/Thesis/10251/131698>
28. What is UML | Unified Modeling Language. (n.d.). <https://www.uml.org/what-is-uml.htm>
29. Agencia Española de Medicamentos y Productos Sanitarios. <https://www.aemps.gob.es/>
30. Fight Retinal Blindness! <https://savesightregistries.org/fight-retinal-blindness/>
31. React. <https://es.react.dev/>
32. Sequelize. <https://sequelize.org/>
33. Express.js. <https://expressjs.com/>
34. Node.js. <https://nodejs.org/es>
35. Angular. <https://angular.io/>
36. PM2. <https://pm2.keymetrics.io/>
37. Usability Testing, <https://www.nngroup.com/articles/usability-testing-101/>
38. Figma. <https://www.figma.com/>
39. Visual Paradigm. <https://www.visual-paradigm.com/>
40. ¿Qué es una API REST? | IBM. (n.d.). <https://www.ibm.com/es-es/topics/rest-apis>
41. How To Structure React Projects From Beginner To Advanced. (n.d.). <https://blog.webdevsimplified.com/2022-07/react-folder-structure/>
42. MySQL. <https://www.mysql.com/>
43. Sanidad implanta el sistema Orion Clinic en el Hospital de Elda - Portada - Portal del departamento de Elda. (n.d.). https://elda.san.gva.es/home/-/asset_publisher/c9X9/content/id/8146954
44. ClinVar. <https://www.ncbi.nlm.nih.gov/clinvar/>
45. Ensembl. <https://www.ensembl.org/index.html>
46. LOVD. <https://www.lovd.nl/>
47. HTTP | MDN. (2023, July 24). <https://developer.mozilla.org/es/docs/Web/HTTP>

48. Hasanzad, M., Sarhangi, N., Ehsani Chimeh, S., Ayati, N., Afzali, M., Khatami, F., ... & Aghaei Meybodi, H. R. (2022). Precision medicine journey through omics approach. *Journal of Diabetes & Metabolic Disorders*, 21(1), 881-888.

Anexo I – Prototipo G-MAC

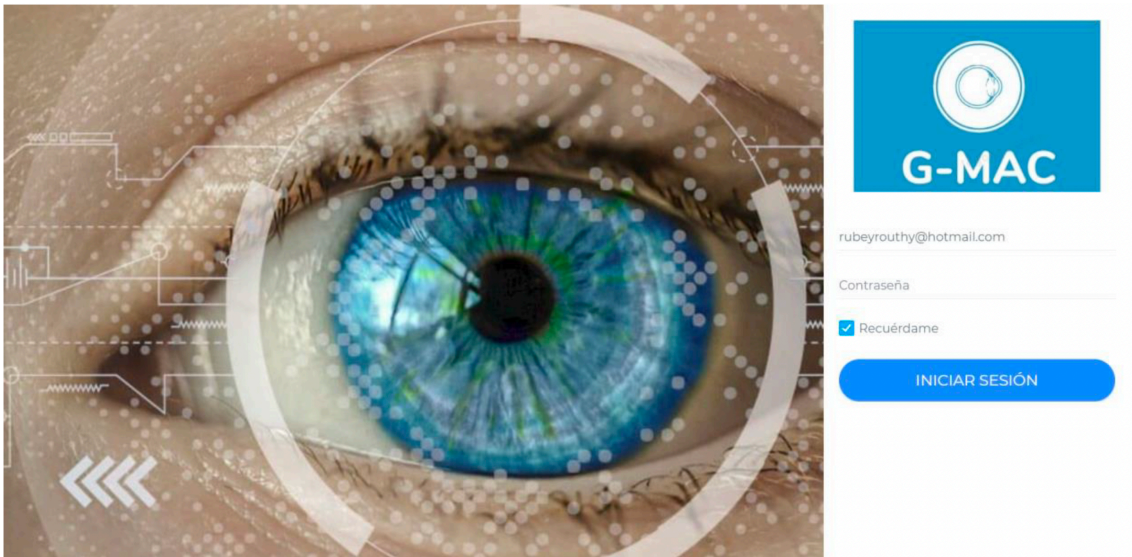


Ilustración 58 G-MAC Login o inicio de sesión

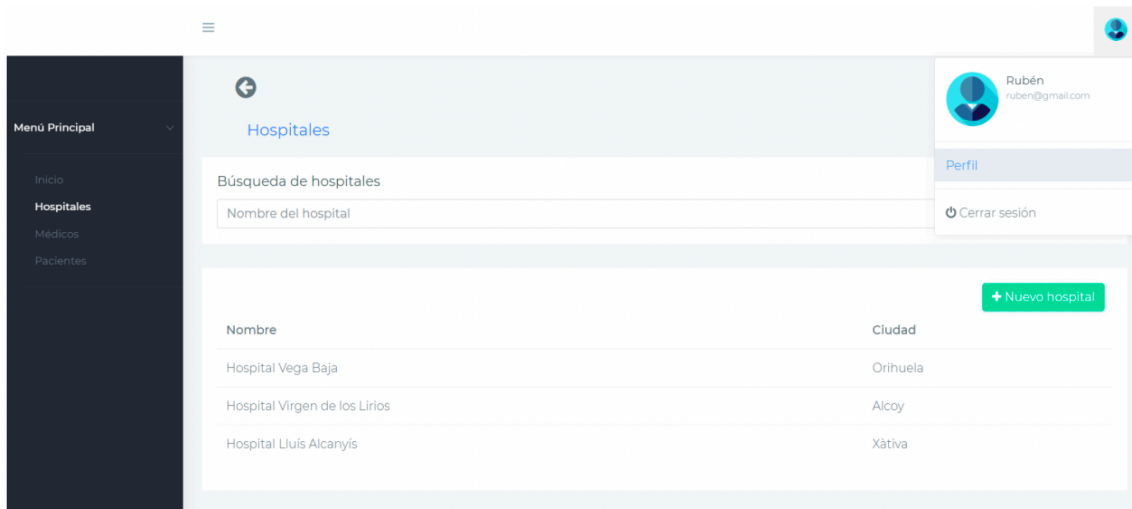


Ilustración 59 G-MAC Ventana hospitales

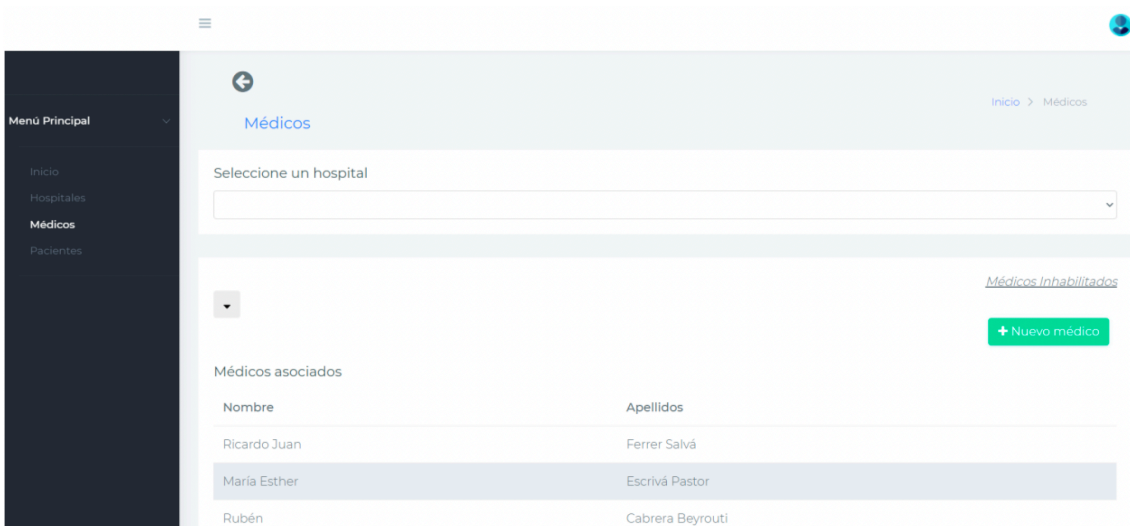


Ilustración 60 G-MAC Ventana médicos

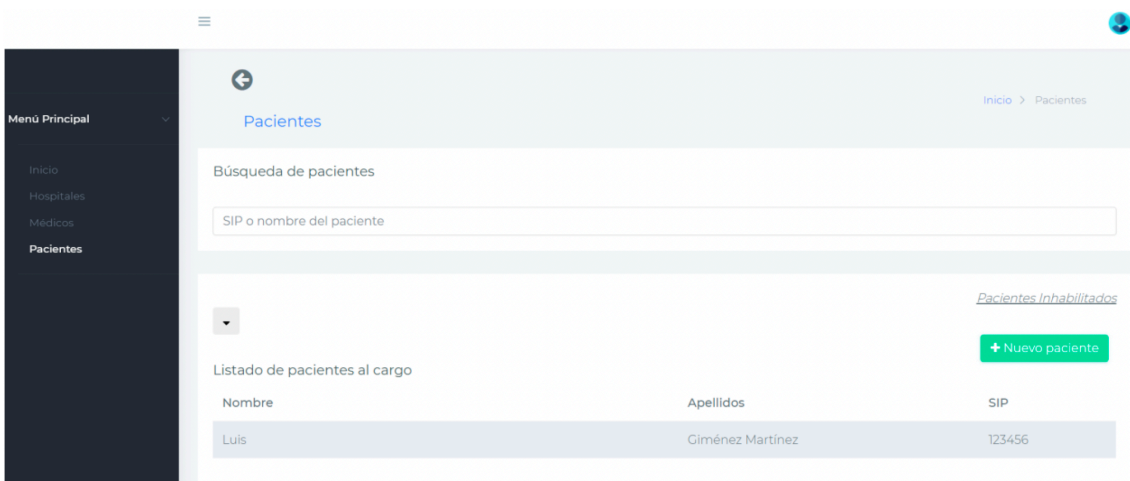


Ilustración 61 G-MAC Ventana Pacientes

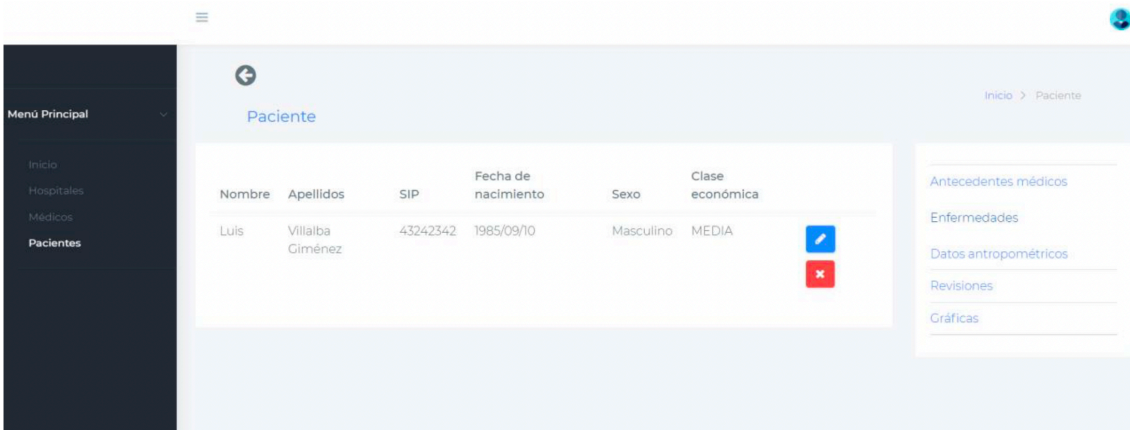


Ilustración 62 G-MAC Ventana detalle del paciente

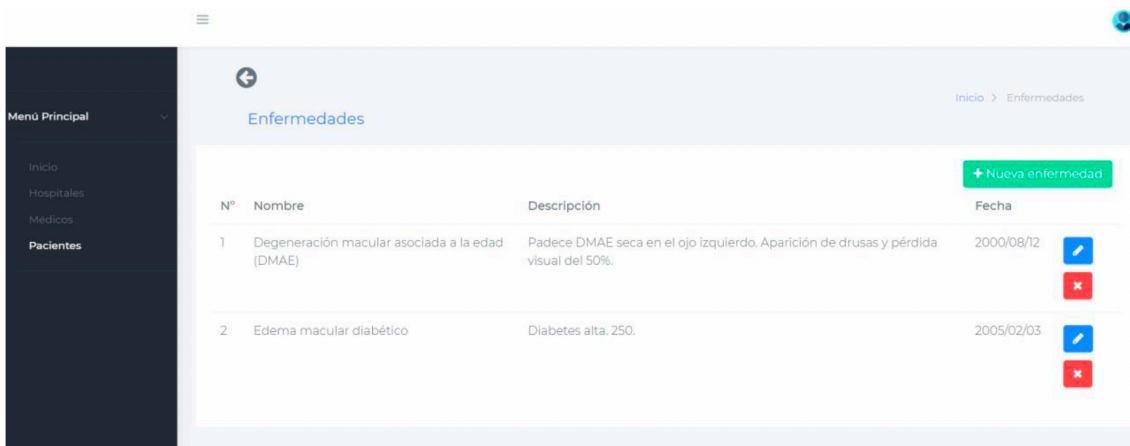


Ilustración 63 G-MAC Ventana Gestión de enfermedades de un paciente

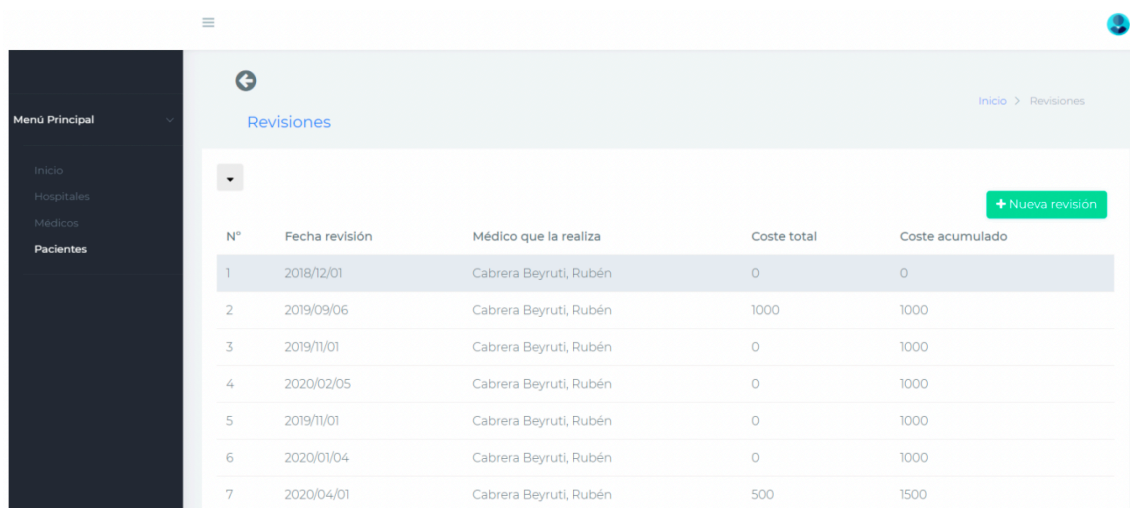


Ilustración 64 G-MAC Listado de revisiones

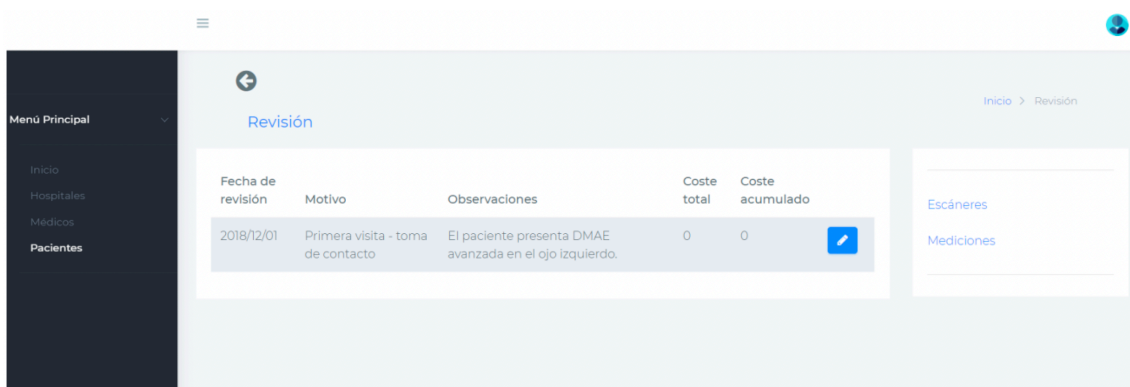


Ilustración 65 G-MAC Ventana detalle de una revisión

Menú Principal

- Inicio
- Hospitales
- Médicos
- Pacientes**

Inicio > Escáneres de retina

Escáneres de retina

Seleccione un ojo

IZQUIERDO

+ Nuevo escáner de retina

N°	Fecha	Edema macular quístico	Grosor retina (micras)	Puntos reflectantes	Líquido subretiniano	Afectación capa interna	Afectación capa externa	Tipo membrana neovascular
1	2019/09/06	No	10	No	No	Sí	No	TIPO 2

Ilustración 66 G-MAC Gestión de escáneres de retina

Menú Principal

- Inicio
- Hospitales
- Médicos
- Pacientes**

Inicio > Mediciones

Mediciones

Seleccione un ojo

IZQUIERDO

+ Nueva medición

N°	Fecha	Agudeza visual	Percepción de luz	Movimiento de manos	Cuenta dedos
1	2018/12/01	0.7	Sí	Sí	Sí
2	2019/09/06	0.6	Sí	Sí	Sí
3	2019/11/01	0.9	Sí	Sí	Sí
4	2020/02/05	0.9	Sí	Sí	Sí
5	2020/07/01	0.8	Sí	Sí	Sí

Ilustración 67 G-MAC Gestión de mediciones

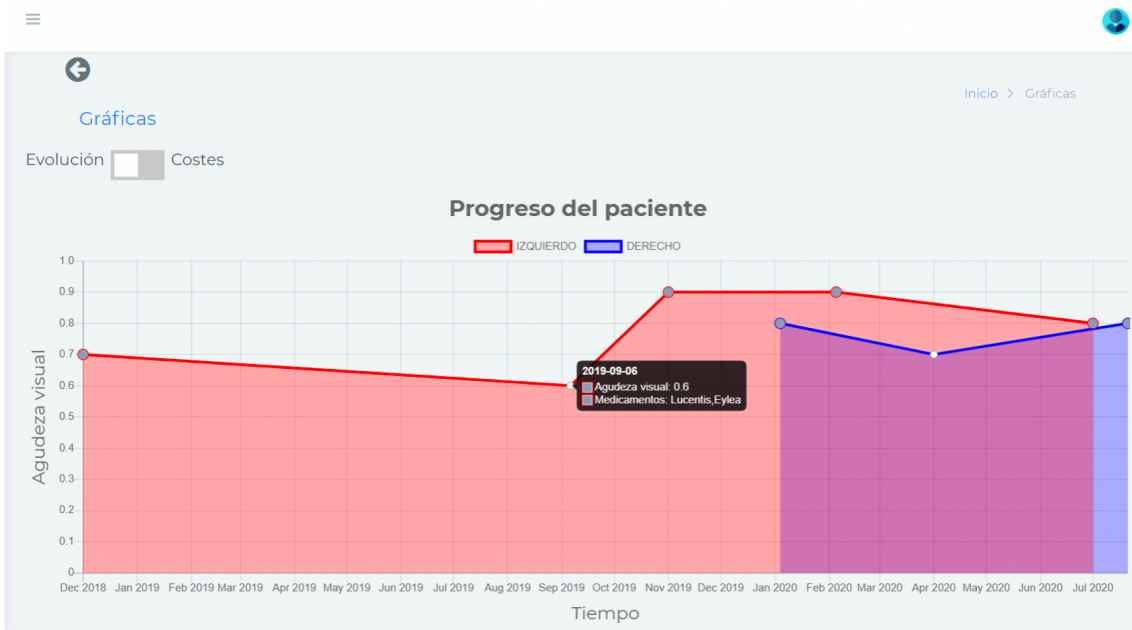


Ilustración 68 G-MAC Visualización de gráficas

Anexo II – Instrucciones de la prueba de usabilidad

- **Tratamiento de un paciente nuevo al sistema:**

Llega a la consulta un paciente por primera vez

1. Ir a Pacientes
2. Crear paciente
3. Ver paciente
4. Ver sexo y edad del paciente
5. Añadir dato antropométrico (Peso y altura solo)
6. Añadir antecedente médico: diabetes
7. Diagnosticar Edema macular diabético en ambos ojos
8. Crear una primera revisión con una medición y escáner de retina a cada ojo.
Se decide un medicamento y el paciente vuelve al cabo de un mes para su aplicación.
9. Ver detalle de primeras revisiones a cada ojo.
10. Volver a la página del paciente y aplicar medicamento diferente en cada ojo.
*Repetir para los siguientes dos meses.
El siguiente mes (4a consulta), se realiza una medición de cada ojo. Uno de los dos mejora mucho.*
11. Dar de alta ese ojo.
12. Ver gráficas de evolución del paciente.

- **Tratamiento de un paciente con tratamientos previos pero nuevo para el médico:**

Llega a la consulta un paciente nuevo para el médico, pero ya ha sido tratado por otros médicos.

1. Añadir paciente existente (SIP: 3) (Paciente creado previamente a la prueba)
2. Buscar por SIP
3. Ver resumen
4. Ver evolución de la agudeza visual
5. Ver último medicamento aplicado en el historial de revisiones
6. Aplicar una dosis más del último medicamento

- **Edición de los datos incorrectos de un paciente:**

Llega a la consulta un paciente con datos incorrectos y se han de corregir.

1. Añadir paciente existente
2. Buscar por SIP (SIP: 3) (Paciente creado previamente a la prueba)
3. Ver paciente
4. Editar año de nacimiento de 1950 a 1951
5. Ver enfermedades
6. Eliminar DMAE
7. Cambiar edema macular pseudofáquico por edema macular diabético.
8. Ver antecedentes médicos
9. Eliminar fumador
10. Cambiar hipertensión por diabetes
11. Ver datos antropométricos
12. Cambiar altura del último dato de 165 a 175.
13. Ver revisiones
14. Eliminar la revisión vacía por error
15. Ver detalle nueva última revisión
16. Abrir edición de revisión
17. Cancelar edición de la revisión

18. Volver a la lista de pacientes
- **Médico con 10 pacientes:**
 1. Ir a Inicio
 2. Ver y entender todas las gráficas de población de paciente.
 - **Cambio de contraseña:**
 1. Cambiar la contraseña del usuario

Anexo III – Fragmentos de código

Backend

En el Fragmento de código 1 se muestra la configuración de Sequelize para trabajar con tres bases de datos según el entorno: producción, desarrollo o *test*.

```
require('dotenv').config();
...
module.exports = {
  development: {
    username: process.env.MYSQL_DEV_USER,
    password: process.env.MYSQL_DEV_PASSWORD,
    database: process.env.MYSQL_DEV_DB_NAME,
    host: process.env.MYSQL_DEV_HOST,
    dialect: 'mysql',
  },
  test: {
    username: process.env.MYSQL_TEST_USER,
    password: process.env.MYSQL_TEST_PASSWORD,
    database: process.env.MYSQL_TEST_DB_NAME,
    host: process.env.MYSQL_TEST_HOST,
    port: process.env.MYSQL_TEST_PORT,
    dialect: 'mysql',
    logging: false,
  },
  production: {
    username: process.env.MYSQL_USER,
    password: process.env.MYSQL_PASSWORD,
    database: process.env.MYSQL_DB_NAME,
    host: process.env.MYSQL_HOST,
    dialect: 'mysql',
    logging: false,
  },
};
```

Fragmento de código 1 Configuración de las bases de datos en Sequelize

El Fragmento de código 2 muestra la definición del modelo «Médico» en Sequelize. En esta parte también se definen las asociaciones con otros modelos y los *scopes* como, por ejemplo en este caso, uno para excluir la contraseña cuando se obtiene un modelo de la base de datos mediante los *finders* de Sequelize.

```

'use strict';
module.exports = (sequelize, DataTypes) => {
  const Medico = sequelize.define("Medico", {
    nombre: {
      type: DataTypes.STRING,
      allowNull: false
    },
    apellidos: {
      type: DataTypes.STRING,
      allowNull: false
    },
    correo: {
      type: DataTypes.STRING,
      allowNull: false,
      unique: true,
      validate: {
        | isEmail: true
      }
    },
    password: {
      type: DataTypes.STRING,
      allowNull: false
    },
    sexo: {
      type: DataTypes.STRING,
      defaultValue: 'Indefinido',
      validate: {
        | isIn: [['Indefinido', 'Masculino', 'Femenino']]
      }
    },
    telefono: DataTypes.INTEGER,
    servicio_unidad: DataTypes.STRING,
    num_colegiado: DataTypes.INTEGER,
    centro: DataTypes.STRING,
    deBaja: {
      type: DataTypes.BOOLEAN,
      allowNull: false,
    },
    rol: {
      allowNull: false,
      type: DataTypes.STRING,
      isIn: [['MEDICO', 'ADMIN']]
    }
  }, {
    freezeTableName: true, // Model tableName will be the same as the model name
    timestamps: false // Omits creating two extra table columns: created_at and other
  });

  Medico.addScope(
    'defaultScope', {
      attributes: {exclude: ['password']}
    }
  )
  Medico.addScope(
    'includePassword', {
      attributes: {include: ['password']}
    }
  )
  // Relations
  Medico.associate = function(models) {
    Medico.belongsToMany(models.Paciente, {as: 'pacientes', through: 'PacienteMedico', freezeTableName: true, timestamps: false});
    Medico.hasMany(models.Revision);
    Medico.belongsTo(models.Hospital, {as: 'hospital', freezeTableName: true, timestamps: false, foreignKey: 'hospitalId'});
  }

  return Medico;
};

```

Fragmento de código 2 Definición del modelo “Medico” en Sequelize

Los *finders* de Sequelize que permiten realizar consultas a la base de una forma declarativa mediante los modelos. Por ejemplo, en el Fragmento de código 3 se observa el controlador del

punto de entrada de la API del que se obtienen los antecedentes médicos de un paciente, el cual hace uso de los *finders*. Simplemente se hace un *findAll* sobre el modelo «PacienteAntecedente» (añadido en el AGC-MAC) y se incluyen los modelos «Paciente» y «Antecedente» en la consulta para que también los devuelva en la respuesta, la cual tiene una estructura como la del ejemplo del Fragmento de código 4.

```
/******  
 * ANTECEDENTES MÉDICOS  
*****  
// Obtiene todos antecedentes de un Paciente  
exports.getAntecedentesByPaciente = (req, res) => {  
  
  const pacienteId = req.params.id;  
  
  PacientesAntecedente.findAll({  
    include: [  
      {  
        model: Paciente,  
        as: "paciente",  
        where: { id: pacienteId }  
      },  
      {  
        model: Antecedente,  
        as: "antecedente"  
      }  
    ],  
  })  
    .then(data => {  
      res.send(data);  
    })  
    .catch(err => {  
      res.status(500).send({  
        message:  
          err.message || "Algún error mientras se obtenían los antecedentes."  
      });  
    });  
};
```

Fragmento de código 3 Controlador de la API para la obtención de los antecedentes de un paciente


```
[
  {
    "id": 1,
    "descripcion": "",
    "pacienteId": 1,
    "antecedenteId": 5,
    "paciente": {
      "edad": 56,
      "id": 1,
      "iniciales": "A. A.",
      "sip": 1,
      "year_nac": 1967,
      "sexo": "Femenino",
      "deBaja": false
    },
    "antecedente": {
      "id": 5,
      "nombre": "Enfermedades reumatológicas"
    }
  },
  {
    "id": 2,
    "descripcion": "",
    "pacienteId": 1,
    "antecedenteId": 1,
    "paciente": {
      "edad": 56,
      "id": 1,
      "iniciales": "A. A.",
      "sip": 1,
      "year_nac": 1967,
      "sexo": "Femenino",
      "deBaja": false
    },
    "antecedente": {
      "id": 1,
      "nombre": "Fumador"
    }
  }
]
```

Fragmento de código 4 Respuesta de la API en formato JSON de la consulta de los antecedentes médicos de un paciente

En el Fragmentos de código 5 se muestra un ejemplo de los *seeders* de Sequelize que se han agregado al AGC-MAC. Este ejemplo muestra la facilidad para añadir medicamentos al sistema.

```

'use strict';
module.exports = {
  up: (queryInterface, Sequelize) => {
    return queryInterface.bulkInsert('Medicamento',
    [{
      nombre: 'Lucentis',
      descripcion: 'Descripcion 1',
      precio: 500
    },
    {
      nombre: 'Eylea',
      descripcion: 'Descripcion 2',
      precio: 500
    },
    {
      nombre: 'Avastin',
      descripcion: 'Descripcion 3',
      precio: 500
    },
    {
      nombre: 'Dexametasona',
      descripcion: 'Descripcion 4',
      precio: 1000
    },
    {
      nombre: 'Iluvien',
      descripcion: 'Descripcion 4',
      precio: 1000
    },
    {
      nombre: 'Beovu',
      descripcion: 'Descripcion 4',
      precio: 1000
    }
    ], {});
  },

  down: (queryInterface, Sequelize) => {
    return queryInterface.bulkDelete('Medicamento', null, {});
  }
};

```

Fragmentos de código 5 Seed de Sequelize para poblar las tablas de los medicamentos

Frontend

Siguiendo el mismo ejemplo de los antecedentes médicos, el frontend también los define los modelos para saber cómo interpretar los objetos recibidos de la API. Los nombres de las propiedades deben ser consistentes con el *backend*. De lo contrario, el *frontend* leerá un valor indefinido.

```

import { Antecedente } from "./antecedente.model";
import { Paciente } from "./paciente.model";

export interface PacienteAntecedente {
  id: number
  paciente: Paciente,
  antecedente: Antecedente,
  descripcion: string,
}

```

Fragmento de código 6 Definición del modelo PacienteAntecedente en el frontend

Los componentes de la interfaz solicitan obtener recursos necesarios o modificarlos a través de sus servicios, que llaman a las APIs necesarias para cumplir con sus solicitudes. El Fragmento de código 7 muestra algunas funciones de las que se compone el servicio de los antecedentes médicos de un paciente. Este servicio construye la URI correspondiente según el recurso solicitado y la pasa al objeto «gMacApi», el cual realiza la llamada al *backend*.

```

import gMacApi from 'apis/gmac';
import { Antecedente } from 'models/antecedente.model';
import { PacienteAntecedente } from 'models/pacienteAntecedente';

const baseUrl = '/antecedentes';

const pacienteAntecedentesService = {
  cargarAntecedentesDePaciente: (id : number) : Promise<PacienteAntecedente[]>=>{

    let url = '/pacientes/' + id + '/antecedentes';

    return gMacApi.get<PacienteAntecedente[]>(url);
  },

  añadirAntecedente : async (pacienteId : number, antecedente: Antecedente) =>{

    let url = '/pacientes/' + pacienteId + '/antecedentes';

    return gMacApi.post<Antecedente>(url, antecedente);
  },
}

```

Fragmento de código 7 Servicios de frontend para los antecedentes médicos de un paciente

Finalmente, en el Fragmento de código 8 corresponde al archivo TSX que construye la página del detalle de un hospital. En este ejemplo se pueden ver los puntos fuertes de React, que son los siguientes:

- *Componentización de interfaces*: componentes que se pueden importar e incrustar en cualquier punto de la interfaz. Esto ayuda a la reusabilidad y a la depuración del código. En este ejemplo encontramos el componente «ListaMedicos», que devuelve una tabla

paginada con la lista de unos médicos pasados como argumento. Este componente en concreto también se usa en la página de Médicos.

- *Estados*: cada componente puede tener un o más estados. Este ejemplo tiene dos estados: el hospital y los médicos del hospital («medicosHospital»). Cuando alguno de los estados cambia, se actualizan los componentes que dependan de él. En este caso, si los estados están indefinidos, se muestra una animación de carga. En cambio, si tienen un valor, se muestran con los componentes correspondientes. Otro caso donde es de más interés es en la creación de objetos (p.e. pacientes) y la consecuente actualización de la lista.
- *Hooks*: los *hooks* es una característica de React complicada de explicar brevemente, pero en resumen, permite gestionar los estados extrayendo la lógica de los componentes y así, seguir el principio de *Single Responsibility*. Este ejemplo utiliza los *hooks* «useHospital» y «useMedicosByHospital», que devuelven los estados y realizan las llamadas a los servicios correspondientes.

```

import PageTitle from "components/PageTitle";
import { Container } from "react-bootstrap";
import DatosHospital from "../../features/hospitales/DatosHospital";
import { useHospital } from "hooks/useHospital";
import { useParams } from "react-router-dom";
import CustomBreadCrumb from "components/CustomBreadCrumb";
import ListaMedicos from "features/medicos/ListaMedicos";
import { useMedicosByHospital } from "hooks/useMedicosByHospital";
import { ColorRing } from "react-loader-spinner";

function HospitalPage() {
  const hospitalId = parseInt(useParams().hospitalId || "");
  const { hospital } = useHospital(hospitalId);
  const { medicosHospital } = useMedicosByHospital(hospitalId);

  return (
    <Container fluid className="main d-flex flex-column align-items-center">
      <CustomBreadCrumb data={{paths:["/hospitales/", "/hospitales/" + hospitalId], titles:["Hospitales", "Hospital"]}}/>
      <PageTitle texto = "Hospital"/>
      <Container fluid className="main-page-container">
        {
          hospital === undefined ?
          <Container fluid className="d-flex flex-row justify-content-center">
            <ColorRing
              visible={true}
              height="80"
              width="80"
              ariaLabel="blocks-loading"
              wrapperStyle={{}}
              wrapperClass="blocks-wrapper"
              colors={['#e15b64', '#f47e60', '#f8b26a', '#abbd81', '#849b87']}
            />
          />
          :
          <DatosHospital data ={hospital}/>
        }
      </Container>
      <Container fluid className="main-page-container">
        <Container fluid className="d-flex flex-column align-items-start">
          <h4>Médicos asociados</h4>
          </Container>
          {
            medicosHospital === undefined ?
            <Container fluid className="d-flex flex-row justify-content-center">
              <ColorRing
                visible={true}
                height="80"
                width="80"
                ariaLabel="blocks-loading"
                wrapperStyle={{}}
                wrapperClass="blocks-wrapper"
                colors={['#e15b64', '#f47e60', '#f8b26a', '#abbd81', '#849b87']}
              />
            </Container>
            :
            <ListaMedicos data={medicosHospital}/>
          }
        </Container>
      </Container>
    </Container>
  );
}

export default HospitalPage;

```

Fragmento de código 8 Página del hospital con React