



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Diseño y Desarrollo de una Plataforma Web para el estudio
de Cardiomiopatías Familiares

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Pérez Gómez, Ana María

Tutor/a: Pastor López, Oscar

Director/a Experimental: GARCIA SIMON, ALBERTO

CURSO ACADÉMICO: 2021/2022

Resumen

Hoy en día la informática está presente prácticamente en todos los campos. En los últimos tiempos ha tomado una gran importancia en el ámbito de la salud, donde gracias a ella se han conseguido grandes avances además de haber mejorado notablemente la fiabilidad de los diagnósticos gracias a la recolección y estudio de la gran cantidad de datos clínicos recogidos y filtrados.

En este contexto de trabajo se ha diseñado y desarrollado un módulo web dónde los profesionales de la salud podrán visualizar, añadir y editar los datos de sus pacientes afectados por cardiomiopatías y sus familiares, además de las relaciones entre ellos. Todo ello se visualizará en forma de árbol genealógico incluyendo la información más relevante para el posterior diagnóstico del paciente. Este módulo web permitirá a los médicos analizar la información de manera más clara y concisa además de ayudar en el correcto diagnóstico gracias al estudio familiar. Este módulo forma parte de una aplicación web dónde se recogen todos los datos necesarios para los profesionales de la salud.

Para su desarrollo se ha seguido una metodología Design Science realizando un estudio del contexto y de los objetivos para la correcta creación del artefacto a implementar. Además, su diseño está basado en un modelo conceptual cubriendo todos los requisitos a cumplir.

Palabras clave: Modelado Conceptual, Informática médica, Sistema de Gestión de Cardiopatías, Ingeniería de Sistemas de Información

Resum

Hui dia la informàtica és present pràcticament en tots els camps. En els últims temps ha pres una gran importància en l'àmbit de la salut, on gràcies a ella s'han aconseguit grans avanços a més d'haver millorat notablement la fiabilitat dels diagnòstics gràcies a la recollida i estudi de la gran quantitat de dades clíniques recollides i filtrats.

En aquest context de treball s'ha dissenyat i desenvolupat un mòdul web on els professionals de la salut podran visualitzar, afegir i editar les dades dels seus pacients afectats per cardiomiopaties i els seus familiars, a més de les relacions entre ells. Tot això es visualitzarà en forma d'arbre genealògic incloent la informació més rellevant per al posterior diagnòstic del pacient. Aquest mòdul web permetrà als metges analitzar la informació de manera més clara i concisa a més d'ajudar en el correcte diagnòstic gràcies a l'estudi familiar. Aquest mòdul forma part d'una aplicació web on es recullen totes les dades necessàries per als professionals de la salut.

Per al seu desenvolupament s'ha seguit una metodologia Design Science realitzant un estudi del context i dels objectius per a la correcta creació de l'artefacte a implementar. A més, el seu disseny està basat en un model conceptual cobrint tots els requisits a complir.

Paraules clau: Modelatge Conceptual, Informàtica mèdica, Sistema de Gestió de Cardiopaties, Enginyeria de Sistemes d'Informació

Abstract

Nowadays computer science is present in practically all fields. In recent times it has taken on great importance in the field of health, where great advances have been achieved in addition to having significantly improved the reliability of diagnoses thanks to the collection and study of the large amount of clinical data collected and filtered.

In this context of work, a web module has been designed and developed where health professionals can visualize, add and edit the data of their patients affected by cardiomyopathies and their families, as well as the relationships between them. All this will be displayed in the form of a family tree including the most relevant information for the subsequent diagnosis of the patient. This web module will allow doctors to analyze the information more clearly and concisely as well as help in the correct diagnosis thanks to the family study. This module is part of a web application where all the necessary data for health professionals is collected.

For its development, a Design Science methodology has been followed, carrying out a study of the context and objectives for the correct creation of the artifact to be implemented. In addition, its design is based on a conceptual model covering all the requirements to be met.

Keywords: Conceptual Modeling, Medical Informatics, Cardiopathies Management System, Information Systems Engineering

Índice General

CAPÍTULO 1: INTRODUCCIÓN	9
1.1 Motivación.....	9
1.2 Metodología	12
1.3 Estructura	14
CAPÍTULO 2: INVESTIGACIÓN DEL PROBLEMA	16
2.1 OGMIOS.....	16
2.2 Genómica y enfermedades.....	18
2.3 Modelo Conceptual	23
2.4 CRD.....	26
2.5 Genogramas	30
2.6 Tecnologías	32
CAPÍTULO 3: DISEÑO DE LA SOLUCIÓN	35
3.1 Ampliación Modelo Conceptual.....	35
3.2 Requisitos	41
3.3 Implementación	45
3.3.1 Diseño.....	45
3.3.2 Implementación	54
3.3.3 Integración	57
3.4 Workflow	60
CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN	62
CAPÍTULO 5: CONCLUSIONES	64
5.1 Trabajos Futuros	66
BIBLIOGRAFÍA	68
ANEXO	71



Índice de Figuras

Figura 1: framework Design Science	13
Figura 2: ciclo de Diseño Design Science.....	14
Figura 3: fases del ciclo cardíaco.....	19
Figura 4: anatomía del corazón	20
Figura 5: comparación de intervalos QT en electrocardiograma	21
Figura 6: estructura fichero VCF	24
Figura 7: estructura de un Contig.....	24
Figura 8: modelo conceptual de OGMIOS	25
Figura 10: vista de la aplicación CRD inicial	27
Figura 9: estructura CRD	27
Figura 11: arquitectura de aplicación CRD	29
Figura 12: simbología de genograma	31
Figura 13: stack de tecnologías	34
Figura 14: modelo conceptual Pedigree GA4GH.....	36
Figura 15: fragmento inicial modelo conceptual.....	36
Figura 16: opción 1 ampliación modelo conceptual OGMIOS	38
Figura 17: opción 2 ampliación modelo conceptual OGMIOS	39
Figura 18: opción 3 y modelo conceptual final ampliación OGMIOS	40
Figura 19: mockup adición pestaña Genograma	46
Figura 20: mockup vista general genograma.....	48
Figura 21: mockup vista leyenda genograma	49
Figura 22: mockup vista formulario genograma.....	50
Figura 23: mockup vista adición de familiares	50
Figura 24: estructura de la aplicación con los archivos del genograma	57
Figura 25: frontend - Ramas del workflow del proyecto.....	60
Figura 26: backend - Ramas del workflow del proyecto	60

Índice de Tablas

Tabla 1: comparativa de librerías para implementación genograma	33
Tabla 2: especificación de requisitos	45
Tabla 3: plantillas genograma	47
Tabla 4: equivalencia Modelo conceptual e Interfaz de Usuario.....	51
Tabla 5: equivalencia de requisitos con elementos mockup.....	54
Tabla 6: acciones probadas en la plataforma web.....	63

Índice de Fragmentos de Código

Fragmento de código 1: creación del árbol inicial.....	55
Fragmento de código 2: creación del genograma.....	55
Fragmento de código 3: funciones manejadoras del genograma ...	55
Fragmento de código 4: hooks useEffect del genograma.....	56
Fragmento de código 5: devolución de la pestaña genograma	56
Fragmento de código 6: backend - Consulta obtención genograma	58
Fragmento de código 7: backend - Resolver de getGenogram	58
Fragmento de código 8: backend - Driver getGenogram.....	58
Fragmento de código 9: backend - Mutation de actualización genograma	58
Fragmento de código 10: backend - Resolver de updateGenogram	59
Fragmento de código 11: backend - Driver updateGenogram.....	59
Fragmento de código 12: frontend - Actualización consulta genograma	59
Fragmento de código 13: frontend - Mutation de exportación genograma	59
Fragmento de código 14: actualización script Base de datos tabla CardioDiagnosis	60

CAPÍTULO 1:

Introducción

1.1 Motivación

Comprender el genoma humano en su totalidad ha sido una tarea imposible desde su descubrimiento hasta que finalmente se ha conseguido publicar su secuenciación al 100% este 31 de marzo de 2022 [1]. El entendimiento del genoma es clave para responder una pregunta para la que no se ha podido encontrar respuesta todavía: *¿Por qué somos como somos?* Aunque se ha teorizado una posible respuesta desde diversas áreas tan dispares como la filosofía o la biología, la genómica constituye uno de los campos más prometedores para poder responder a esta pregunta.

Nos encontramos en un contexto científico donde, por primera vez en la historia, el reto de comprender el genoma humano se ha convertido en una meta alcanzable gracias a los avances en las técnicas de secuenciación [2][3]. Uno de los muchos beneficios de estos avances es el hecho de que el diagnóstico genético haya dado lugar al éxito de nuevos y prometedores modelos como la medicina de precisión, un enfoque que intenta prevenir las enfermedades y tratarlas considerando las características de cada individuo. Todo el dominio generado a partir de estos avances ha dado como resultado un mayor conocimiento de qué es lo que nos hace ser como somos.

Además, se ha llegado al punto en el que el paradigma tradicional de la medicina no es capaz de dar respuesta al hecho de que la variedad genética de cada individuo influye de manera drástica en la evolución de las enfermedades y la respuesta a los tratamientos y medicamentos existentes. Por ello, no debemos basarnos únicamente en la sintomatología de los pacientes, sino que se requiere tener una estrategia mucho más dirigida, en la cual la selección de los tratamientos sea individualizada. Estas necesidades constituyen el caldo de cultivo que ha dado lugar al nacimiento de la medicina de precisión [4][5]. Aunque existen muchas definiciones de medicina de precisión, algunas de ellas con una intención más comercial que científica, podemos definirla como la estratificación (en base a características genéticas) de pacientes para proporcionar mejores resultados en el área de la salud. Por ejemplo, discriminando medicamentos o ajustando la dosis de estos.

En nuestro largo viaje para explorar el código humano, estamos en su comienzo, y cuando alcancemos nuestra meta, lograremos uno de los avances más importantes en la historia de la humanidad, la comprensión completa del genoma. Con ello accederemos a un conocimiento absoluto además de permitirnos manipular nuestro código genético a nuestra propia voluntad. Con este descubrimiento, vemos las dos caras de una misma moneda, ya que su uso en el tratamiento de enfermedades y la mejora de las dificultades de la vida humana siempre ha sido y será un objetivo aceptado y deseable de los aspectos éticos, mientras que, si se aplica para la manipulación de rasgos genéticos como, por ejemplo, el color de ojos, la inteligencia o la fuerza pueden provocar un debate en el que percibimos objeciones a la invasión de la moralidad.

Por otro lado, en referencia al contexto tecnológico actual podemos destacar la importancia de la bioinformática, la informática aplicada a la salud y el auge del *Big Data*, donde los médicos se encuentran en una situación de auge de las nuevas tecnologías en el que las aplicaciones para uso médico están a la orden del día. Además, nos encontramos en un momento de la historia en el que se tratan inmensas dimensiones de datos con gran heterogeneidad, lo que está desbordando a las capacidades intuitivas de los profesionales de la salud.

Con respecto a la bioinformática podemos decir que se trata de una disciplina científica de investigación, desarrollo y aplicación de soluciones de cálculo para tratar los datos biológicos. Esta es la ciencia que apoya las pruebas biológicas a través del uso reciente de progreso en técnicas y cálculos estadísticos.

En cuanto a la informática de la salud se puede definir como el uso de la tecnología de la información en el sector sanitario para almacenar, compartir, transferir y analizar conocimientos y datos clínicos. Uno de los usos más reconocibles de la informática médica es la visualización del historial médico de un paciente en línea. La informática en salud incluye la tecnología electrónica y de la información utilizada en la atención al paciente, práctica también conocida como computación clínica, que se define como la implementación y evaluación de sistemas de información y comunicación destinados a mejorar la salud y el cuidado del paciente y la relación entre él y su médico.

Por último, el *Big Data* aplicado a la salud nos permite digerir y procesar datos estructurados y no estructurados, permitiéndonos avanzar en la investigación médica y de atención primaria, que pueden determinar la efectividad de nuestros datos, afectar la causa de algunas enfermedades y brindar nuevas y mejores soluciones. El *Big data* permite a los profesionales de la salud realizar diagnósticos más precisos y seguros, comparando miles de datos en minutos, junto con los síntomas del paciente y las posibles afecciones relacionadas. A través de su uso en la toma de decisiones clínicas, fomenta el desarrollo de la medicina predictiva para identificar enfermedades que los pacientes pueden experimentar en el futuro.

En ambos contextos surge la necesidad de desarrollar una plataforma inteligente para facilitar y promover el acceso a los datos clínicos y genómicos a los investigadores y expertos de la salud.

Con el desarrollo de esta tesis se planea hacer aportaciones en estos ámbitos y así ayudar en el camino de comprensión del genoma humano. Para ello, se ha trabajado en el marco de un proyecto científico (cuyos detalles se explican más adelante) que pretende hacer llegar a los hospitales valencianos un modelo de medicina de precisión en el campo de la cardiología y la oncología pediátrica, concretamente en el desarrollo de este trabajo nos centraremos en el ámbito cardiológico.

Este trabajo de final de grado es un proyecto de investigación, el cual forma parte del proyecto OGMIOS dirigido por el Instituto Valenciano de Investigación en Inteligencia Artificial (VRAIN) de la Universitat Politècnica de València. En dicho proyecto, Vrain realiza su aportación desde dos puntos: i) Mejorando la gestión de datos de pacientes y ii) interpretando el genoma de los pacientes además de generar informes que soporten la toma de decisiones de los médicos. Este TFG participa en la primera aportación mencionada.

En esta tesis se ha diseñado y desarrollado un módulo web para la creación y visualización de genogramas, que se pueden describir como representaciones gráficas (en forma de árbol) de las relaciones familiares en los cuales, además, se recoge el historial médico de cada miembro, entre otras características de interés [6]. En nuestro caso sirven como ayuda al estudio de la predisposición genética a enfermedades cardiovasculares que pueden dar lugar a muerte súbita gracias a la introducción de datos relevantes de los familiares de los pacientes. Este módulo forma parte de un sistema de información inteligente, llamado CRD, desarrollado por el grupo de investigación PROS de Vrain, que permite asistir al experto clínico en el proceso diagnóstico.

Concretamente, la creación de este módulo web surge de la necesidad médica de los doctores que forman parte del proyecto OGMIOS de obtener estudios familiares de sus pacientes mediante la creación y visualización de genogramas a través de la aplicación web. Es por ello por lo que, como resultado del desarrollo de este proyecto, se ha obtenido una parte esencial de la aplicación web que contribuye a la realización de un correcto diagnóstico clínico basado en el código genético de cada paciente. Todo ello les permitirá identificar procesos biológicos, emocionales y sociales de las familias de sus pacientes a lo largo de diversas generaciones.

1.2 Metodología

Para el desarrollo de esta tesis se ha seguido la metodología de investigación *Design Science* propuesta por Roel J. Wieringa [7].

Esta metodología se basa en el diseño y la investigación de artefactos en un contexto dado y el estudio de la interacción entre ellos. Con esta definición diferenciamos los componentes fundamentales de *Design Science*: el objeto de estudio y las dos actividades principales. Por un lado, el artefacto en el contexto dado se corresponde al objeto de estudio mientras que la función de diseñar e investigar la interacción entre ellos serían las dos actividades principales.

Cabe destacar que un artefacto por si mismo no resuelve ningún problema y que, dependiendo del contexto en el cuál se encuentre, ayudará de una manera u otra. Por ello, se necesita una comprensión previa del contexto donde se va a desarrollar dicho artefacto. Tras la correcta identificación del problema, *Design Science* hace uso de diferentes métodos para su resolución. En primer lugar, los problemas de diseño (su objetivo es crear o mejorar un artefacto que resuelva un problema), tratados por el ciclo de diseño; en segundo lugar, las preguntas de conocimiento (su objetivo es obtener respuestas de conocimiento sin crear ningún artefacto), respondidas mediante el ciclo empírico.

Este TFG va a seguir un ciclo de diseño. Dicho ciclo comienza evaluando los objetivos de las partes interesadas para conseguir, gracias a la creación de un artefacto, mejorar un problema identificado en un contexto, todo ello satisfaciendo los requerimientos establecidos. El ciclo de diseño consiste en una iteración de tres fases: Investigación del problema, Diseño de la solución y Validación de la solución.

Para este trabajo, en primer lugar, se estudiará el contexto social y el de conocimiento. Además, se investigará acerca del contexto en el que nos encontramos para el desarrollo del artefacto. Este contexto sería la aplicación web principal desarrollada como cuaderno de datos para la introducción de datos clínicos (CRD), de donde surge la necesidad de poder realizar estudios familiares a partir de árboles genealógicos. Por ello, el artefacto que se va a desarrollar es el módulo para la creación y visualización de genogramas. Todo ello constituye el marco de trabajo con el que se va a trabajar, mostrado de forma esquemática en la Figura 1.

El objetivo principal de esta tesis es:

Desarrollar un módulo web para la creación y visualización de genogramas

Tras la correcta comprensión del contexto y el objetivo principal surgen cinco preguntas de investigación que se resolverán a lo largo del desarrollo del TFG:

1. ¿Qué tecnologías se pueden utilizar para el desarrollo del artefacto?
2. ¿El modelo actual del proyecto OGMIOS da soporte al módulo a desarrollar?
3. ¿Cuáles son los requisitos del módulo a desarrollar?
4. ¿Cómo implementar un genograma?
5. ¿Resulta de utilidad para los profesionales de la salud?

Una vez finalizada la tesis, tras su desarrollo y habiendo obtenido las respuestas a todas las preguntas de investigación, se conseguirán dos contribuciones:

1. La ampliación del modelo conceptual de OGMIOS.
2. La ampliación de la plataforma web de uso clínico.

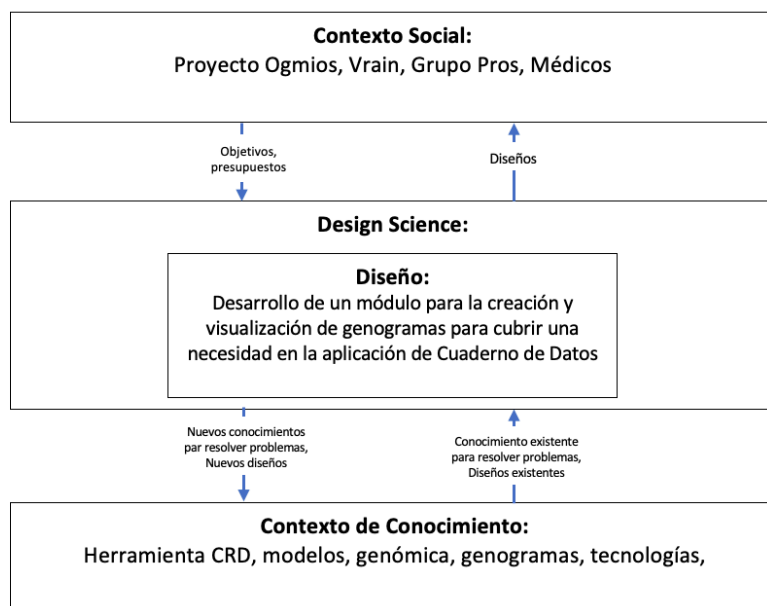


Figura 1: framework Design Science

1.3 Estructura

Los contenidos de la tesis están estructurados basándose en la metodología elegida, ya que las tres fases de desarrollo del ciclo de diseño van a ser quienes guíen el índice de este trabajo. Por ello, en cada una de las secciones centrales de este TFG (sección 2, 3 y 4), se resolverán las preguntas asociadas a cada fase del ciclo de diseño mostradas en la Figura 2. Además, el desarrollo de cada una de las partes nos permitirá responder a las preguntas de investigación originadas en el punto anterior.

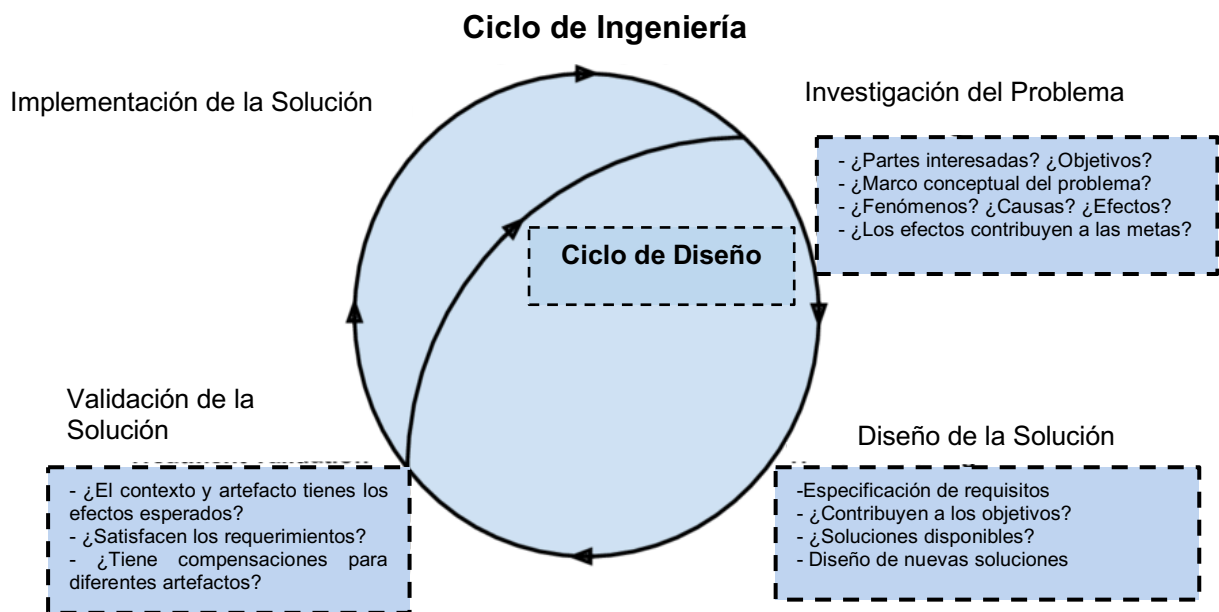


Figura 2: ciclo de Diseño Design Science

Por tanto, tras la introducción, nos encontraremos con el [Capítulo 2](#) que equivale a la Investigación del Problema, donde se realizará un estudio descriptivo para la correcta comprensión del dominio. Se definirán los conceptos más importantes que facilitarán el seguimiento de este TFG. Se comenzará hablando del proyecto OGMIOS, seguido de la genómica y las enfermedades que se tratan en dicho proyecto, después se expondrá el modelo conceptual de OGMIOS, tras ello se explicará qué es y cómo se ha implementado la aplicación del proyecto llamada CRD en la cual se incluirá nuestro módulo web, también se definirán los genogramas y su nomenclatura y por último se hará un análisis de las tecnologías empleadas para el desarrollo de esta plataforma web. Además, en el desarrollo de esta parte responderemos a la primera pregunta de investigación.

En el [Capítulo 3](#) de esta tesis se llevará a cabo el diseño de la solución, donde partiendo del modelo conceptual del proyecto OGMIOS se ampliará

dicho modelo y se realizarán la especificación de requisitos y a partir de ello se realizará el diseño, la implementación y posterior integración de la solución propuesta, también se expondrá el *workflow* de este trabajo. Asimismo, se dará lugar a la resolución de las preguntas de investigación 2, 3 y 4.

En la [sección cuarta](#) se realizará la validación de la solución mediante las opiniones y sensaciones de los usuarios que van a hacer uso del módulo web desde el punto de vista tanto de la facilidad de uso como de la usabilidad y accesibilidad. Todo ello nos permitirá resolver la última pregunta de investigación propuesta.

Para finalizar, en la última parte de este trabajo, [capítulo 5](#), además de responder de forma explícita a cada una de las preguntas de investigación, se recogerán las conclusiones donde se expondrá todo lo aprendido en base a tres ejes: el eje tecnológico, el eje metodológico y el eje científico.

CAPÍTULO 2:

Investigación del Problema

2.1 OGMIOS

Tras los últimos avances en las tecnologías de secuenciación surge la necesidad de integrar la información genómica de cada paciente junto con sus datos clínicos. En este marco de trabajo nace el proyecto OGMIOS.

El objetivo principal de este proyecto, liderado por el Instituto Valenciano de Investigación en Inteligencia Artificial (VRAIN) de la Universitat Politècnica de València, concretamente por el grupo PROS, es la creación de una plataforma comercial de interpretación de datos genómicos de pacientes para ayudar al personal clínico en la prevención, diagnóstico y tratamiento de la enfermedad. Para materializar ese objetivo dos dimensiones esenciales deben de ser correctamente sincronizadas: experiencia clínica y tecnologías de última generación. Por ello, el equipo que desarrolla el proyecto incluye científicos líderes en el campo de la inteligencia artificial de la Universidad Politécnica de Valencia (UPV), servicios médicos de los hospitales más importantes de la Comunidad Valenciana como el Hospital Universitario y Politécnico La Fe, el Hospital Clínico Universitario de Valencia y el Hospital Alicante, además de una empresa líder en su sector (Bionos Biotecnología).

Tras el logro del objetivo principal de esta plataforma de referencia se conseguirá dar respuesta a dos preguntas fundamentales en la práctica clínica:

1. ¿Puede un contexto genómico determinado definir la predisposición a padecer enfermedad?
2. ¿Son las variantes identificadas en un contexto específico susceptibles de producir otras enfermedades?

La correcta respuesta a estas preguntas es esencial para asegurar la viabilidad de una medicina de precisión.

OGMIOS se desarrolla basándose en un modelo conceptual común que facilita la estandarización y la interoperabilidad de los datos además de utilizar tecnologías de última generación para la gestión avanzada de datos que

integran técnicas de tres tipos: modelado conceptual, de soporte ontológico fundacional como modelo básico de conocimiento, y de inteligencia artificial explicable.

El sistema de información inteligente desarrollado por OGMIOS genera una tecnología que se nutre de:

- Una plataforma de secuenciación que permite obtener la información genómica asegurando que se cumplen los más exigentes estándares de calidad.
- Un soporte ontológico sólido que permite definir claramente los actores, los conceptos y las entidades implicadas en el proceso de diagnóstico clínico y genético, resolviendo así los problemas de ambigüedad e interoperabilidad que suelen afectar a este dominio tan complejo.
- Un conocimiento clínico especializado tanto en oncología infantil como en predisposición genética al desarrollo de enfermedades cardiovasculares.
- Una arquitectura moderna integrativa de información genómica que permite obtener todo el conocimiento asociado a cada variante a partir de las distintas fuentes de datos existentes, solventando los problemas de heterogeneidad y dispersión que éstas presentan.
- Un soporte metodológico riguroso que permite evaluar la calidad y relevancia de la evidencia recogida, asegurando de esta forma que las decisiones se tomen en base a los datos más fiables, actuales y precisos.
- Un sistema de actualización continua que permite monitorizar la aparición de nueva evidencia que pueda afectar de forma significativa a las decisiones tomadas sobre un diagnóstico ya realizado, manteniendo actualizado el sistema de información, así como los procesos analíticos y de toma de decisión.

Y que proporciona:

- Un proceso automatizado de interpretación y clasificación de variantes basado en los estándares más actuales utilizados por la comunidad.
- Un soporte analítico inteligente que, utilizando tecnologías de ciencia de datos e inteligencia artificial, permite cruzar la evidencia científica con los datos clínicos del paciente para identificar con precisión aquellas variantes directamente implicadas en el desarrollo de los síntomas presentados.
- Mecanismos de interacción con el usuario que guían en todo el proceso diagnóstico y facilitan la toma de decisiones.

Inicialmente el sistema de información se utilizará para el estudio de la predisposición genética al cáncer pediátrico y a las cardiopatías familiares con riesgo de muerte súbita. Una vez validada y puesta en funcionamiento en el mercado, la plataforma podrá ser adaptada a otras áreas clínicas.

2.2 Genómica y enfermedades

Los dos ámbitos clínicos seleccionados para tratar en el proyecto OGMIOS han sido los de oncología pediátrica y cardiología. Ya que, además de constituir dos campos de interés clínico evidente, se consideran especialmente atractivos debido a la importancia y necesidad de una respuesta precisa en el campo de la predisposición genética al cáncer y a las cardiopatías familiares con riesgo de muerte súbita.

Comenzando por las Cardiomiopatía familiares, que es el ámbito clínico en el que mi proyecto va a aportar conocimiento, se conoce como una enfermedad del músculo cardíaco. Esta enfermedad aumenta el tamaño del corazón o lo hace más grueso y rígido de lo normal. Hay casos en los que los pacientes que sufren esta enfermedad no se dan cuenta de estar afectados, sin embargo, en otros casos suele desarrollar síntomas como insuficiencia cardíaca, ritmos cardíacos anormales o incluso paro cardíaco.

Enfermedades como los infartos, la hipertensión arterial y algunas infecciones entre otras, también pueden causar cardiomiopatía. Aunque en muchos pacientes se desconoce la causa de la enfermedad en algunos tipos de cardiomiopatía son hereditarias. El tratamiento para estas puede incluir medicinas, cirugía, otros procedimientos médicos, además de cambios en el estilo de vida [8].

Existen diferentes tipos de miocardiopatía, para este proyecto nos hemos centrado en la Miocardiopatía Dilatada, Miocardiopatía Hipertrófica y el Síndrome de QT largo.

En primer lugar, la miocardiopatía dilatada (MCD) se caracteriza por la dilatación y daño en la contracción del ventrículo izquierdo o ambos ventrículos, se observa una alteración en la geometría normal de estos, con tendencia a la esfericidad y aumento de los diámetros diastólicos y sistólicos, así como sus volúmenes, con un deterioro en la función ventricular sistólica, y un volumen del latido marcadamente disminuido, todos estos conceptos y movimientos cardíacos se puede observar en la Figura 3¹. Además de ser la tercera causa más frecuente de fallo cardíaco después de la enfermedad coronaria y la hipertensión, es la causa más frecuente de daño miocárdico primario, así como

¹ Figura 3, imagen extraída de: <http://bombacardiaca.blogspot.com/p/ciclo-cardiaco.html>

de muerte súbita y trasplante cardíaco. La miocardiopatía dilatada puede ser familiar, de origen genético, secundarias a inflamación e infección, debido a sustancias tóxicas, desórdenes metabólicos, anomalías inmunológicas o de origen idiopático, entre otras muchas causas[9].

Hasta un 35% de los casos de cardiomiopatía dilatada están determinados genéticamente (cardiomiopatía dilatada familiar), y suele ser hereditaria de transmisión autosómica dominante, donde autosómico significa que el gen en cuestión está localizado en uno de los cromosomas no sexuales (es decir, del cromosoma número 1 al 22) y dominante que con que se herede el gen anormal de solo uno de los padres, se puede presentar la enfermedad [10]. Por ello, la evolución de la enfermedad puede verse influenciada positivamente por la identificación precoz de los portadores.

De esta forma es importante realizar un estudio genético de los pacientes obteniendo así [11]:

- Un diagnóstico genético-molecular preciso que proporcione información esencial, especialmente para formas de MCD con penetrancia o expresividad variable.
- Mejor estimación del riesgo y pronóstico del paciente en particular.
- Identificación de casos de alto riesgo de muerte súbita cardíaca.
- Ajuste e inicio oportuno de la terapia médica.
- Identificación de otros familiares en riesgo de desarrollar la enfermedad.

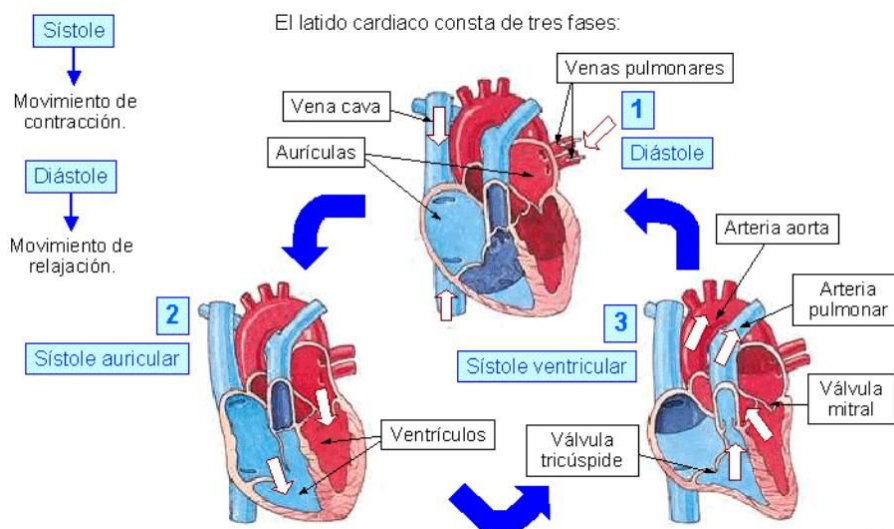


Figura 3: fases del ciclo cardíaco

En cuanto a la Miocardiopatía Hipertrófica (MCH) la OMS la designa como una enfermedad genética, caracterizada por un proceso único de hipertrofia muscular primaria en la cual el miocardio resulta engrosado, además, personas con MCH también sufren de un arreglo anormal de las células musculares del corazón, como se muestra en la Figura 4². Todo ello dificulta la salida de la sangre del corazón [12]. El diagnóstico de esta enfermedad se apoya en los resultados de un examen físico y un electrocardiograma (ECG), además de un examen principal basado en una ecocardiografía.

Este tipo de miocardiopatía generalmente se transmite de padres a hijos, hay un 50% de probabilidades de que los hijos de padres con esta condición hereden una mutación genética por la enfermedad, por ello es la enfermedad cardiovascular hereditaria más común con una prevalencia del 0,1% a 0,2% en la población general, es decir, 1 de cada 500 habitantes [9]. Afecta a mujeres y hombres por igual y además es causa de muerte en todas las edades.

Con frecuencia, la MCH no se consigue diagnosticar debido a que algunas veces, esta ocurre sin un bloqueo significativo de la sangre y muchas personas con la enfermedad no presentan síntomas y esto les permite llevar vidas normales sin problemas significativos. Aún así algunos de los síntomas más comunes de la miocardiopatía son [13]:

- Dificultad para respirar.
- Dolor de pecho.
- Desvanecimiento.
- Sensación de un ritmo cardíaco rápido.
- Murmullo del corazón, que un doctor puede detectar al escuchar tu corazón.

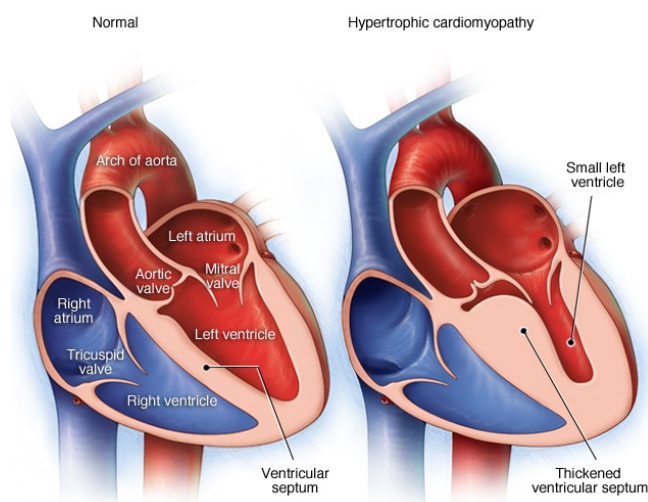


Figura 4: anatomía del corazón

² Figura 4, imagen extraída de: <https://cirugiacardiovascular.com.mx/operacion-a-corazon-abierto/enfermedades-cardiovasculares/miocardiopatia-hipertrofica/>

Por último, referido a cardiomiopatías, el Síndrome de QT largo (SQTL) es una afección que puede hacer que el corazón lata rápido y de manera caótica, se caracteriza por la prolongación del intervalo QT en el electrocardiograma, una mayor propensión a desmayos repentinos, convulsiones y arritmias ventriculares, que en ocasiones puede culminar en fibrilación ventricular y en el peor de los casos en muerte súbita. La prevalencia de la enfermedad es de 1 cada 2000 habitantes. El diagnóstico de la SQTL se basa en los síntomas, la duración del intervalo QT, el electrocardiograma, la historia clínica y familiar y las pruebas genéticas [14].

El síndrome de QT largo es un trastorno del ritmo cardíaco causado por anomalías en el sistema de recarga eléctrica del corazón, sin embargo, por el contrario, a los otros dos tipos de cardiomiopatías mencionados anteriormente, en este tipo de enfermedad la estructura del corazón es normal.

Cuando el corazón funciona correctamente, bombea sangre al cuerpo con cada latido. Para bombear esta sangre, las cavidades del corazón se contraen y relajan, una acción coordinada controlada por el sistema eléctrico de este. Las señales eléctricas (llamadas impulsos) viajan desde la parte superior hasta la parte inferior del corazón y hacen que este se contraiga y lata. Después de cada latido, el sistema se recarga para prepararse para el siguiente latido.

En el síndrome de QT largo lo que ocurre es que el sistema eléctrico del corazón tarda más de lo normal en recargarse entre latidos. Este retraso, que suele observarse en un electrocardiograma, se llama intervalo QT largo, se considera así cuando el QT es mayor a 470ms en hombres y mayor a 480ms en mujeres, como se muestra en la Figura 5³.

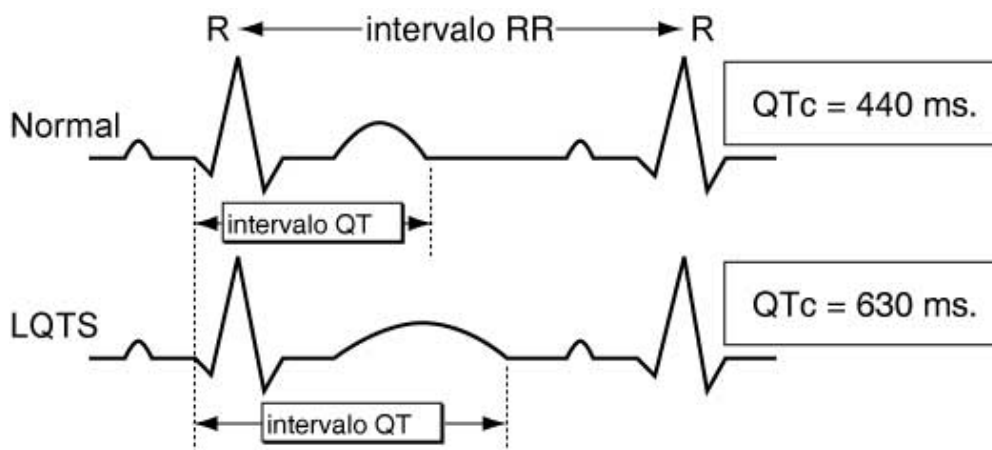


Figura 5: comparación de intervalos QT en electrocardiograma

³ Figura 5, imagen extraída de: <https://temas.sld.cu/medicamentoterapeutica/tag/taquicardia/>

Dependiendo de la causa, el síndrome de QT largo se puede agrupar en dos categorías principales:

- *Síndrome de QT largo congénito*, es decir, si naces con la enfermedad. Algunas formas de SQTl son el resultado de una mutación genética que se transmite de padres a hijos.
- *Síndrome de QT largo adquirido*, se da debido a una afección médica de fondo o un medicamento. Este tipo suele ser reversible cuando se identifica la causa que lo originó.

A partir de los 40 años de edad las personas que sufren el SQTl tienen una mayor probabilidad de sufrir los síntomas de esta afección [15]. Sin embargo, los signos del síndrome de QT largo congénito pueden aparecer en los bebés durante las primeras semanas o meses de vida, en la infancia o más tarde. También existen los casos en los que nunca se presentan síntomas.

Finalmente, en cuanto al cáncer infantil, se define como aquel que se presenta entre el nacimiento y los 14 años de edad. Estos cánceres son muy raros y difieren de cómo se forman y propagan los cánceres de adultos, además de su tratamiento y la respuesta a estos. Los tipos más comunes de cáncer infantil son: la leucemia, los tumores cerebrales y de la médula espinal, el linfoma, el neuroblastoma, el tumor de Wilms (un tipo de cáncer de riñón), el retinoblastoma (cáncer en la retina) y los cánceres de huesos y tejidos blandos [16].

Alrededor de 1.100 niños enferman de cáncer cada año en España [17]. Aunque no es muy frecuente, el cáncer infantil tiene un gran impacto en la sociedad ya que se trata de un problema de salud importante. Aunque los datos y resultados han mejorado en el largo camino recorrido, el cáncer es la principal causa de muerte por enfermedades infantiles en los países desarrollados.

Por desgracia, el cáncer infantil no puede ser prevenido, por ello es de suma importancia la prontitud del diagnóstico y el tratamiento. Para impulsar la mejora continua en la calidad de la atención, del diagnóstico y posterior tratamiento los sistemas de datos sobre el cáncer infantil son fundamentales.

La probabilidad de supervivencia de este tipo de cáncer supera el 80 % en los países desarrollados y está por debajo del 30 % en los países de ingresos bajos y medios. Estas tasas de supervivencia más bajas en los países de ingresos bajos y medianos se deben al diagnóstico tardío, la incapacidad para hacer un diagnóstico preciso, la falta de acceso al tratamiento, el abandono de los regímenes de tratamiento y la muerte por toxicidad (efectos secundarios tóxicos) [16]. Por ello, se puede enfatizar la importancia y mejora que supone el acceso a tecnologías y fármacos esenciales.

A diferencia de los cánceres en adultos, se desconoce la causa de la gran mayoría de los cánceres infantiles. Se han realizado numerosos estudios para tratar de identificar las causas, pero muy pocos cánceres en estos grupos de edad son causados por factores ambientales o relacionados con el estilo de vida. Aún así, según datos actuales, cerca del 10% de los niños con cáncer tienen una predisposición genética [\[15\]](#).

2.3 Modelo Conceptual

Como se ha mencionado anteriormente OGMIOS es un proyecto basado en el modelado conceptual. El modelo realizado por el grupo PROS, mostrado en la Figura [8](#), ha servido como entorno de trabajo para numerosas publicaciones del equipo, en las cuales el modelado ha propiciado la mejora en la gestión de datos del genoma [\[18\]](#), la comprensión del genoma humano desde un enfoque del modelado holístico [\[19\]](#) y para la identificación de datos genómicos relevantes apoyando a la medicina de precisión [\[20\]](#). Asimismo, ha marcado las pautas para el desarrollo e implementación de la aplicación del cuaderno de datos.

Tras la comprensión del modelo, se observa que el eje principal es el paciente, del que se recogen los datos más relevantes. El paciente, dependiendo de su enfermedad, puede pertenecer a la especialidad de cardiología o de oncología. Según su clasificación se tienen en cuenta los datos específicos a la afección. En caso de tratarse de un paciente de cardiología, se les realiza una prueba de sangre y un estudio genético. Mientras que si se trata de un paciente de oncología se debe incluir los resultados del test de Ripperger, de Mippog y de CMMRD, todos ellos basados en los criterios correspondientes. Cada paciente, independientemente del ámbito al que pertenezca, cuenta con su propio diagnóstico. En el caso de tratarse de un enfermo de oncología se tiene un diagnóstico más amplio donde se describe el tipo de cáncer y si se trata de un paciente con recaída.

A partir de cada enfermo y de sus características se pueden realizar estudios en los que se incluyan reportes describiendo la evolución del paciente en la enfermedad. En el estudio realizado es importante el análisis de las muestras recogidas a cada enfermo, ya que a partir de la secuenciación de dichas muestras se obtiene un fichero VCF, el cual almacena las variaciones de la secuencia de genes y su información y tiene un formato como el mostrado en la Figura [6](#).

Las variantes VCF que forman el fichero además de especificar a que cromosoma pertenecen (chrom), en qué posición se encuentran (pos), cuáles son las secuencias bases del genoma de referencia (ref) y los alelos alternativos (alt) cuentan también con el campo Filter, el campo Info y el Contig, explicados en los siguientes párrafos.


```

FORMATO ##fileformat=VCFv4.1
RECOMEN- ##fileDate=20090805
DABLES ##source=myImputationProgramV3.1
        ##reference=file:///seq/references/1000GenomesPilot-NCBI36.fasta
OTROS ##contig=<ID=20,length=62435964,assembly=B36,md5=f126cdf8a6e0c7f379d618ff66beb2da,species="Ho
mo sapiens",taxonomy=x>
        ##phasing=partial
INFO ##INFO=<ID=NS,Number=1,Type=Integer,Description="Number of Samples With Data">
        ##INFO=<ID=DP,Number=1,Type=Integer,Description="Total Depth">
        ##INFO=<ID=AF,Number=A,Type=Float,Description="Allele Frequency">
        ##INFO=<ID=AA,Number=1,Type=String,Description="Ancestral Allele">
        ##INFO=<ID=DB,Number=0,Type=Flag,Description="dbSNP membership, build 129">
        ##INFO=<ID=H2,Number=0,Type=Flag,Description="HapMap2 membership">
FILTER ##FILTER=<ID=q10,Description="Quality below 10">
        ##FILTER=<ID=s50,Description="Less than 50% of samples have data">
FORMAT ##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
        ##FORMAT=<ID=GQ,Number=1,Type=Integer,Description="Genotype Quality">
        ##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Read Depth">
        ##FORMAT=<ID=HQ,Number=2,Type=Integer,Description="Haplotype Quality">
CABECERA #CHROM POS ID REF ALT QUAL FILTER INFO FORMAT NA00001
DATOS 20 14370 rs6054257 G A 29 PASS NS=3;DP=14;AF=0.5;DB;H2 GT:GQ:DP:HQ 0|0:48:1:51,51
        20 17330 . T A 3 q10 NS=3;DP=11;AF=0.017 GT:GQ:DP:HQ 0|0:49:3:58,50
        20 1110696 rs6040355 A G,T 67 PASS NS=2;DP=10;AF=0.333,0.667;AA=T;DB GT:GQ:DP:HQ 1|2:21:6:23,27
    
```

Figura 6: estructura fichero VCF

Por un lado, en el campo Filter se especifican los filtros que han conseguido pasar las variaciones pudiendo tomar como valor: PASS, si la variación ha superado todos los filtros impuestos, el nombre del filtro que no ha superado o un punto (.) en caso de no habersele aplicado ningún filtro.

Por otro lado, el campo Info del fichero VCF, permite definir información adicional referente a una posición del cromosoma en la que se han encontrado variaciones, pueden referirse a características como por ejemplo calidades de las lecturas, del alineamiento, número de alelos alternativos, etc.

Por último el *Contig* [21], donde se definen una serie de secuencias superpuestas de ADN utilizadas para hacer un mapa físico que reconstruye la secuencia original de ADN de un cromosoma o una región de este, como se muestra en la Figura 7⁴.

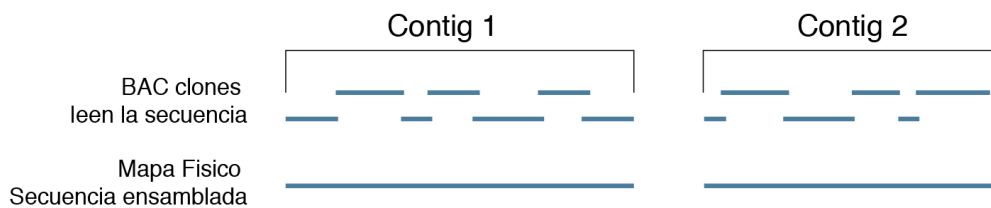


Figura 7: estructura de un Contig

Toda esta información rige el diseño de la aplicación web CRD, en ella se almacena toda la información descrita, necesaria para que los profesionales clínicos puedan realizar un correcto diagnóstico con la aplicación del tratamiento adecuado. Esta aplicación y este modelo han sido el punto de partida para el desarrollo de nuestro modulo web, y posterior incorporación en la aplicación principal.

⁴ Figura 7, imagen extraída de: <https://www.genome.gov/es/genetics-glossary/C%C3%B3ntig>

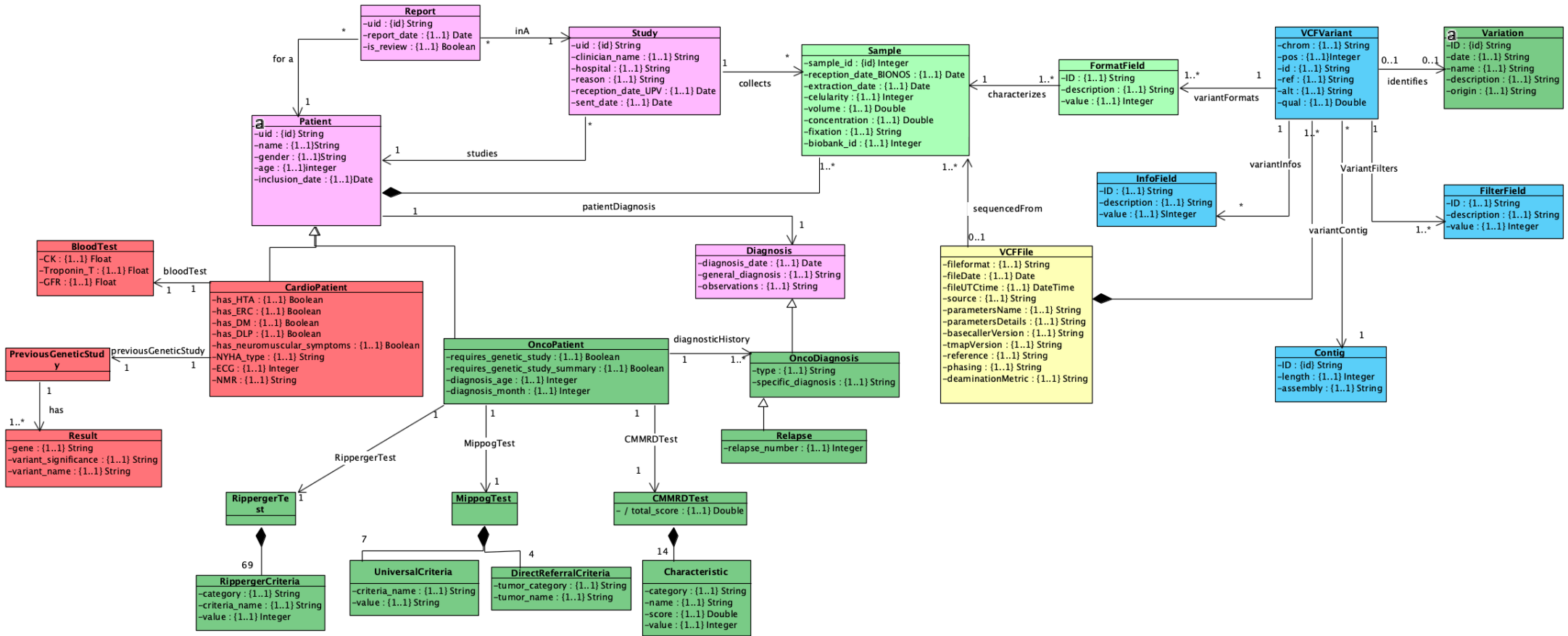


Figura 8: modelo conceptual de OGMIOS



2.4 CRD

El CRD (Cuaderno de datos), es una aplicación web que tiene como objetivo cubrir las necesidades tecnológicas de los profesionales de la salud, dando soporte para la introducción de datos y visualización de los estudios realizados a los pacientes. Esta aplicación permite añadir a cada enfermo y visualizar su diagnóstico, las muestras que se les ha realizado, el estudio genético previo y el informe final.

En la implementación de esta aplicación nos encontramos con una estructura dividida entre el *backend* y el *frontend*. Centrándonos en el desarrollo del *frontend*, tenemos la carpeta *src* que contiene todo el código. Dentro de ella nos encontramos con las carpetas: *components*, *layouts*, *lib*, *pages*, *router* y *theme*. Esta estructura la podemos observar en la Figura 9 y se va a explicar a continuación:

- Dentro de la carpeta **Components**, se almacenan todos los componentes que son compartidos por páginas o por otros componentes.
- La carpeta **Layouts** contiene el código de la estructura que se comparte entre las diferentes páginas de la aplicación.
- **Lib** es donde se encapsula toda la lógica, código e implementación de las distintas librerías, en la implementación del CRD se han usado: **aggrid** para las tablas, **auth0** para la autenticación y **graphql** para la conexión.
- En **Pages** encontramos la implementación tanto de la página principal como de los componentes que la forman, se pueden diferenciar tres componentes: **ListPatients** donde nos permite visualizar y gestionar todos los pacientes registrados, **NewPatient** donde se crean los pacientes y **PatientDetails** que es la vista donde se muestran todos los detalles del paciente.
- En **Router** se usa para la redirección de una página a otra y además implementa *lazy loading*.
- La carpeta **Theme** donde se generan todos los códigos de colores y de paletas usados en la aplicación.

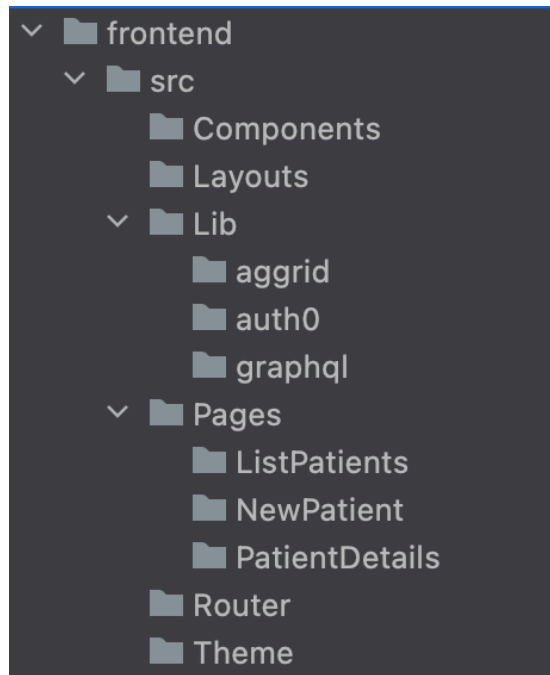


Figura 10: estructura CRD

En la Figura 10 podemos observar la interfaz de la aplicación principal con los componentes definidos e implementados en *Pages*. En la que encontramos la pestaña de crear Nuevo Paciente, la de Listar Pacientes y la ventana de los Detalles del Paciente en la cual se diferencian tres pestañas llamadas *tabs*: Diagnóstico, Muestras y Estudio Genético Previo.

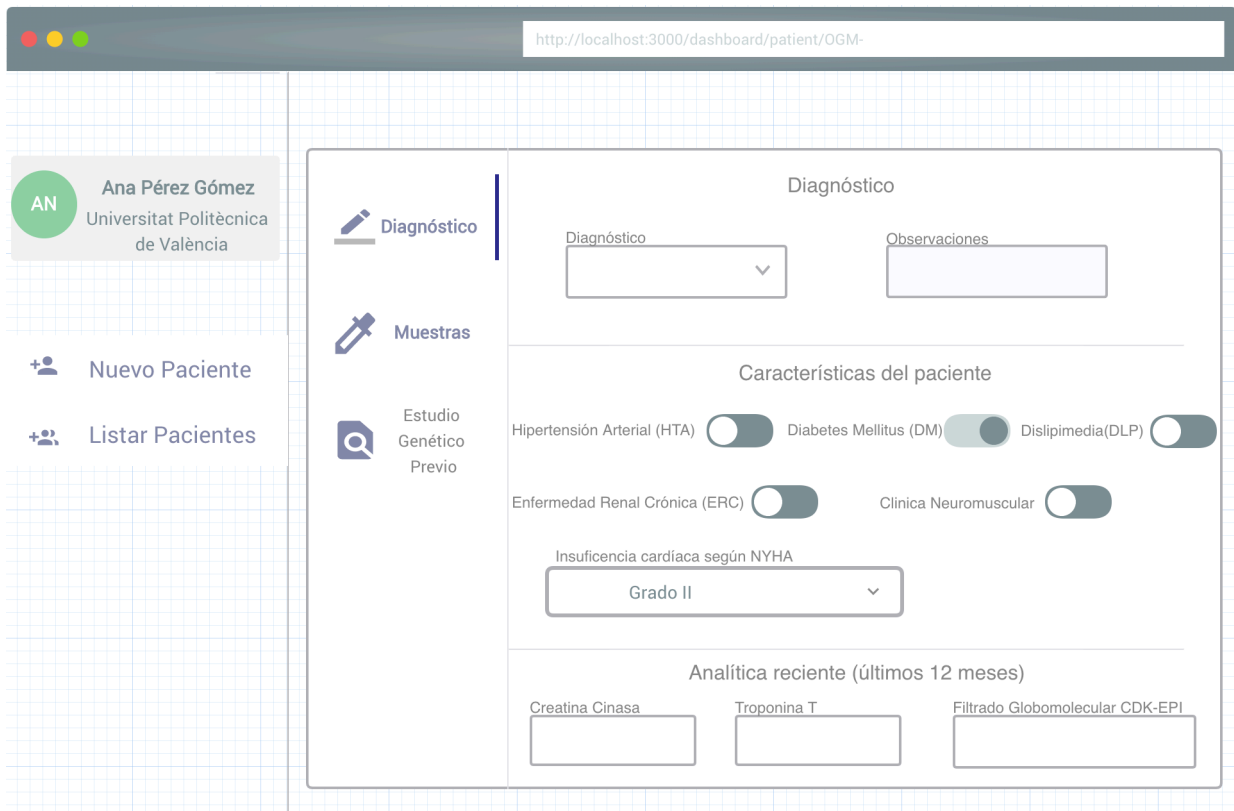


Figura 9: vista de la aplicación CRD inicial

Para la implementación de esta aplicación se ha usado como lenguaje de programación **JavaScript** (JS), ya que es un lenguaje de alto nivel, dinámico e interpretado, está orientado a objetos, basado en prototipos, es imperativo y débilmente tipado, se le conoce como el lenguaje de programación *frontend* por excelencia. Además, presenta un modelo de ejecución asíncrono, esto quiere decir que solo consta de un subproceso o hilo posibilitando que se ejecute una sola cosa a la vez, por ello se pueden realizar largas solicitudes de red sin bloquear el hilo principal y que no haya que preocuparse por los problemas de concurrencia. Aún así, gracias a no ser bloqueante y presentar un bucle de eventos para operaciones de entrada y salida, JavaScript es altamente concurrente [22]. Junto al **HTML** y a **CSS**, es una de las tres tecnologías esenciales en la implementación de páginas webs, su papel en ello es añadir interactividad a estos sitios. Gracias a todas las características que presenta JS y a su modelo de ejecución este lenguaje de programación es la mejor opción para el desarrollo de esta aplicación web.

El entorno de ejecución utilizado es **Node.js**, se trata de un entorno de JavaScript, es de código abierto, multiplataforma y se ejecuta del lado del servidor.

Para el desarrollo del *frontend* se ha utilizado **React** como librería de JS para la creación de forma sencilla de la interfaz de usuario interactiva [23]. React permite actualizar y renderizar de manera eficiente los componentes cuando los datos cambian en función del diseño realizado. Cuenta con vistas declarativas que permiten que el código sea más predecible, y por ello, fácil de depurar. Además, crea componentes encapsulados que manejan su propio estado, y los convierte en interfaces de usuario complejas. A diferencia del DOM Real, definido como un modelo del documento de la aplicación web creado por el navegador, sobre el cuál este aplica los cambios necesarios en cada actualización de estado, React cuenta con DOM virtual, una representación del DOM guardada en memoria, que actúa de intermediario entre los estados de la aplicación y los estados del DOM visualizados por los usuarios [24]. Cuando ocurre un cambio en la aplicación web, el DOM virtual interpreta estos cambios y calcula la manera más eficiente de actualizar el DOM para que renderice la menor cantidad de cambios posibles. Además, React ha añadido los *hooks* a su API, estos permiten usar el estado y otras características de interés en los componentes creados con una función sin necesidad de escribir una clase. En React podemos usar dos tipos de componentes: los componentes de clase, los cuales tienen estado (*Statefull Component*) y cuentan siempre con un método `render()` además de tener ciclos de vida y los componentes funcionales que no tienen estado (*Stateless Components*) sino que se tratan de funciones puras sin métodos `render()` [25]. 1/7/22 20:12:00

Por otro lado, en el *frontend* también se ha usado **GraphQL** para escribir consultar y recuperar datos específicos de la base de datos. Es un lenguaje de consulta para APIs que, en lugar de enviar múltiples peticiones independientes, permite a los clientes hacer una sola petición, y obtener la respuesta con todos los campos que necesitan [26]. Con ello se disminuye la cantidad de datos que deben transferirse a través de la red y esto mejora las aplicaciones.

Para el *backend*, se ha usado **MySQL** como sistema de administración de bases de datos SQL [27], el software es de código abierto y las bases de datos son relacionales, es decir, los datos están distribuidos en tablas. Además, el servidor de datos es muy rápido, confiable, escalable y fácil de usar.

En cuanto a la conexión entre el *frontend* y el *backend* se ha usado **Knex.js** como *query builder*. Knex es una librería de node js que procesa las consultas que se hacen a mysql y las preprocesa para que sean más seguras, evitando las inyecciones de sql. Con ello nos da la garantía de que es un código más profesional, seguro y escalable.

Por último, como sistema de control de versiones se ha hecho uso de la herramienta **Gitlab**, debido a que es un entorno colaborativo de trabajo que ofrece el control de versiones basado en GIT, esto permite gestionar el trabajo colaborativo y el código fuente de cualquier proyecto.

La arquitectura de esta aplicación, representada en la Figura 11, cuenta con un servidor gestionado por **VMware** que es un software de gestión de máquinas virtuales. En nuestro caso contamos con tres máquinas. La primera (VM I) es la que gestiona el tráfico de entrada y salida, donde está **Vyos**, un sistema operativo de red de código abierto que permite democratizar la forma de acceso a las redes. Además, se crea una regla NAT de destino (**DNAT**) para enrutar traduciendo la dirección IP de destino de los paquetes que recibe [28]. Las otras dos máquinas virtuales se destinan una para el *frontend* (VM II) y otra para el *backend* (VM III). En la destinada al *frontend* se aloja React que se comunica con Node mediante GraphQL, esta máquina hace consultas seguras mediante Knex a la máquina virtual del *backend* que es donde se encuentra la base de datos. Se ha diferenciado en dos máquinas virtuales el *frontend* y el *backend* para aumentar la seguridad, puesto que, en caso de verse comprometida la máquina virtual destinada al *frontend*, a la cual tienen acceso los usuarios, la máquina del *backend* esté protegida.

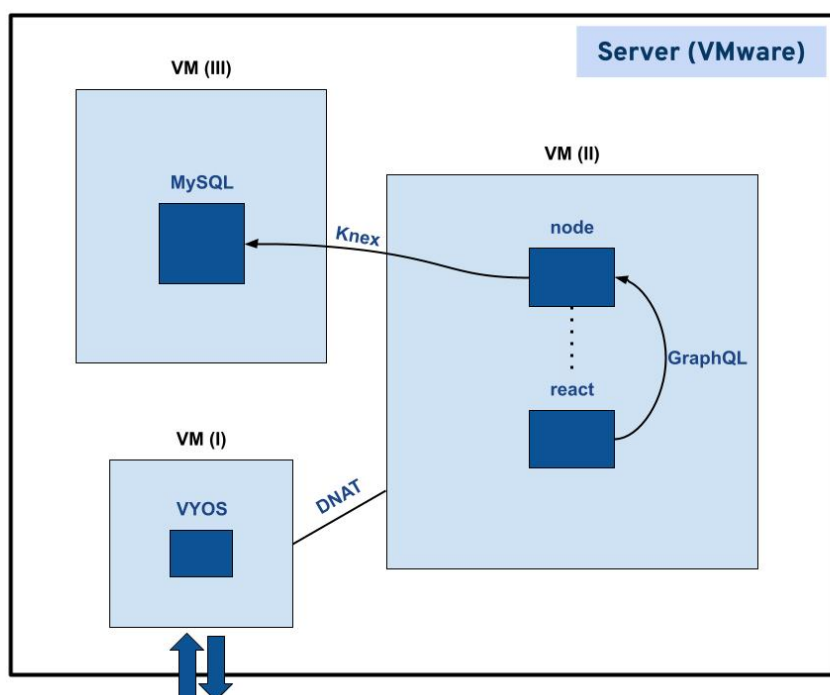


Figura 11: arquitectura de aplicación CRD

2.5 Genogramas

El genograma se define como una representación gráfica de las principales características de un grupo familiar, de al menos tres generaciones, a lo largo del tiempo. Es considerado un instrumento multifuncional y de gran utilidad clínica. Con el genograma se representa información a manera de diagrama de flujo que permite identificar procesos biológicos, emocionales y sociales de una familia a través de diversas generaciones. Es usado en diferentes ámbitos por profesionales como historiadores, antropólogos, biógrafos, médicos familiares, terapeutas familiares, enfermeras, psicólogos, trabajadores sociales, genetistas, etc., quienes lo han adaptado a sus necesidades y objetivos. También se le conoce como árbol genealógico, familiograma, árbol familiar o pedigree humano [\[29\]](#).

Muchos estudios revelan la importancia que tiene la familia y el estudio genético de esta en el proceso de la salud y la enfermedad, ya que gracias a ello se pueden detectar enfermedades hereditarias o causadas por una tendencia en la genética. Además, cuenta con un gran número de ventajas:

- Presentación de un registro médico sistemático.
- Formato gráfico que facilita la lectura.
- Identifica patrones intergeneracionales, biomédicos y psicosociales.
- Permite la evaluación de las relaciones entre el entorno familiar y la enfermedad.
- Reestructura los problemas presentes en los pacientes.
- Contribuye en el campo de la gestión médica y la medicina preventiva.
- Permite hacer evidentes los obstáculos para la cooperación.
- Identifica eventos de la vida que podrían afectar el diagnóstico y tratamiento.
- Revela patrones de enfermedad.
- Promover la educación del paciente.
- Detecta la etapa del ciclo de vida que atraviesa la familia.
- Facilita la formulación de hipótesis terapéuticas.
- Su proceso de elaboración permite establecer una armoniosa relación médico-paciente.
- Es una vía no amenazante para obtener información emocionalmente importante.
- Su elaboración demuestra interés por el paciente y otros familiares.
- Representa la estructura funcional de la familia

La elaboración del genograma consta de tres fases: el trazado de la estructura familiar, el registro de la información sobre la familia y por último, la descripción de la relaciones intrafamiliares [\[30\]](#).

En nuestro caso, el uso de genograma tiene como finalidad determinar el tipo de herencia de una enfermedad dentro de un grupo familiar representando las relaciones entre ellos, señalando los miembros de la familia que tienen determinadas variantes genéticas patógenas, rasgos y enfermedades, como si

se tratase de una radiografía completa de la familia y utilizarlo para la resolución de problemas y prevención en salud individual y conjunta. Al ser un instrumento dinámico debe de actualizarse periódicamente por los cambios que suceden en el transcurso del tiempo.

Dentro de OGMIOS los genogramas se van a aplicar inicialmente para los pacientes con cardiomiopatías, aunque posteriormente se incluirá también para oncología. La interacción con el genograma permitirá a los especialistas de la salud detectar de manera sencilla y basado en el esquema gráfico si el tipo de cardiomiopatía que sufre el paciente es hereditario, que probabilidad hay de que los familiares sufran la enfermedad en un futuro o incluso diagnosticar la enfermedad a individuos que la desconocían permitiendo así una prevención.

La nomenclatura estándar utilizada para representar los elementos de un árbol genealógico se puede ver en la siguiente figura.

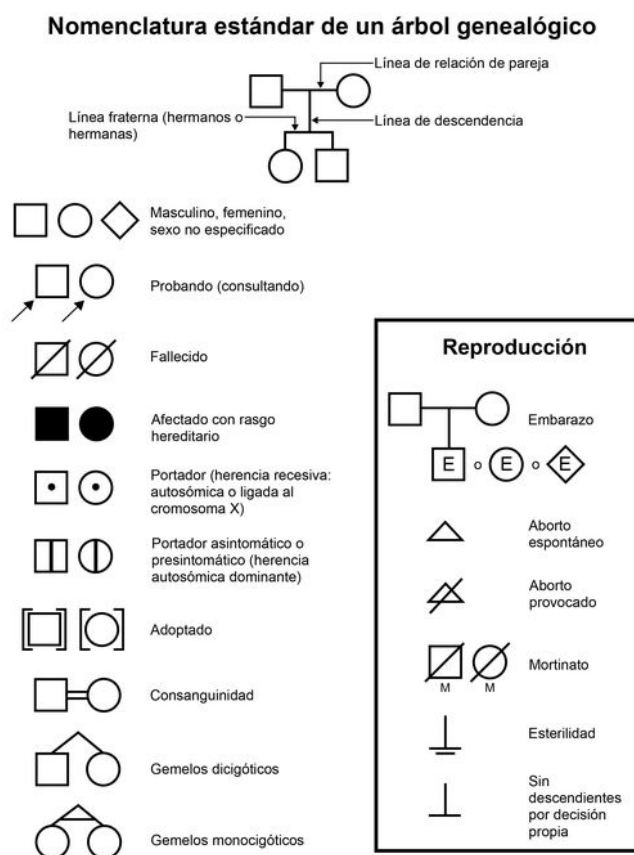


Figura 12: simbología de genograma

Además de la simbología anteriormente explicada en la Figura 12⁵, cada individuo presenta información específica que facilita el contexto del estudio. En nuestro caso la información adicional que nos ha parecido interesante incluir es:

⁵ Figura 12, imagen extraída de: <https://www.cancer.gov/espanol/publicaciones/diccionarios/diccionario-cancer/def/arbol-genealogico>

- Datos personales (nombre, género, edad...)
- Situación con respecto a la enfermedad (probando, portador, afectado, asintomático o sano)

En caso de ser un paciente de oncología:

- Edad de aparición del tumor
- Tipo de tumor y órgano asociado
- En caso de éxitos, especificar la edad

El grado de parentesco, la situación con respecto a la enfermedad, si ha fallecido, si es adoptado y el género se representan según los símbolos especificados en la Figura [12](#).

2.6 Tecnologías

En la creación del módulo web correspondiente al genograma que se desarrolla en este TFG, ha habido que adaptarse a las tecnologías usadas para la implementación de la aplicación principal de la que este módulo va a formar parte (CRD), las cuales han sido mencionadas anteriormente en el punto [2.4](#). Por ello, se ha programado en el lenguaje JavaScript, además de hacer uso de la biblioteca React.

Para el desarrollo del genograma inicialmente hemos tenido que investigar sobre las librerías disponibles que cumplieran con nuestras necesidades y fueran interoperables con React, entre ellas destacan: **GoJS** y **Balkan FamilyTreeJS**

GoJS es una biblioteca de JavaScript con la que se puede crear fácilmente gráficos interactivos en navegadores web modernos. Admite plantillas gráficas y vincula datos de propiedades de objetos de gráficos a datos de modelos. Todo lo que necesita hacer es guardar y recrear un modelo el cual consta de objetos JavaScript simples que contienen las propiedades necesarias para su aplicación. Muchas herramientas y comandos predefinidos implementan el comportamiento estándar requerido por la mayoría de los diagramas [\[31\]](#). La personalización de la apariencia y el comportamiento depende de la configuración de dichas propiedades.

En referencia a la implementación para genogramas que proporciona GoJS, se generan nodos con colores predefinidos siguiendo la estructura según el parentesco familiar. Proporciona funciones que convierten un valor de atributo en un color de pincel o geometría de forma, para así ser agregado al nodo que representa a la persona dependiendo de las características. Las propiedades del nodo que representan a las personas en esta librería se definen según los siguientes atributos: el nombre de la persona, el sexo, la clave de la madre, la clave del padre, la esposa o el marido de la persona y una matriz de los atributos o marcadores que tiene la persona.

En cuanto a Balkan FamilyTreeJS permite crear una aplicación de árbol genealógico con un cuadro que representa las relaciones familiares en una estructura de árbol convencional. Además de tener un diseño llamativo Family Tree JS se ejecuta en cualquier servidor que admita HTML, incluso se puede ejecutar localmente desde un sistema de archivos, no depende de ninguna biblioteca de JavaScript de terceros y funciona en todos los navegadores modernos utilizando SVG para la representación de gráficos [\[32\]](#).

Algunas de las características más importantes de esta librería son: admite datos locales y datos remotos (JSON), proporciona efectos de expandir/contraer, permite alinear el gráfico hasta en 8 orientaciones, presenta soporte panorámico y *zoom*, contiene edición para formulario, posibilita la personalización de los nodos, proporciona barras de desplazamiento y contiene exportador del genograma.

Tras habernos documentado a cerca de las dos librerías en cuestión realizamos un análisis *Fit and Gap* para decidir cuál de las dos librerías se amolda mejor a nuestras necesidades:










Requisitos/Librerías	GoJS	Balkan FamilyTreeJS
Exportación a múltiples formatos		
Diseño intuitivo		
Documentación disponible		
Código abierto		
Posibilidad de adición de múltiples propiedades		
Formulario visible con los atributos de cada nodo		
Inserción de nodos desde la interacción web		
Edición de nodos desde la interacción web		
Visualización intuitiva y atractiva		

Tabla 1: comparativa de librerías para implementación genograma

Diseño y desarrollo de una plataforma web para el estudio de cardiomiopatías familiares

Como se puede observar la librería que cumple con la mayoría de nuestros requisitos es Balkan FamilyTreeJS y por tanto la que hemos usado para el desarrollo de este módulo web.

A continuación, se expone un diagrama que ofrece una vista completa de las diferentes tecnologías que se han ido explicando y por tanto componen el *stack* resultante de este proyecto.

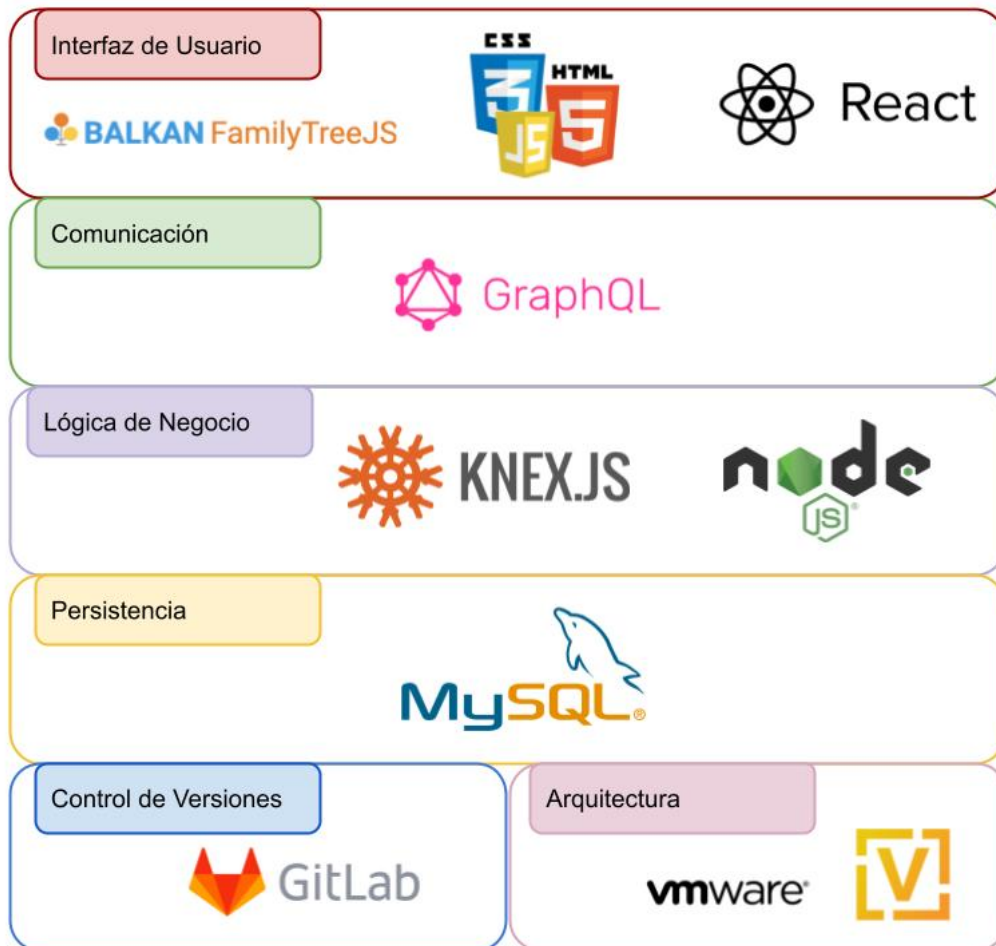


Figura 13: stack de tecnologías

CAPÍTULO 3:

Diseño de la Solución

3.1 Ampliación Modelo Conceptual

Viendo que el modelo inicial de OGMIOS, presentado en el apartado [2.3](#) de esta tesis, concretamente en la Figura [8](#), se puede observar que no cubre la implementación de nuestra plataforma web. Por tanto, se han creado tres opciones para la ampliación de dicho modelo conceptual que contemple el desarrollo de nuestro modulo web. Los tres modelos disponibles realizados se basan en el estándar de *Pedigree* GA4GH, ya que este permite la representación de familias más grandes y complejas, así como el intercambio computable del historial de salud familiar.

El estándar de *Pedigree* GA4GH surge de la necesidad para escalar el análisis genómico a familias grandes y complejas a partir del estudio de información genealógica y familiar de alta calidad, inequívoca y computable. Los datos genealógicos se representan actualmente en formatos heterogéneos que con frecuencia dan como resultado el uso de formatos de mínimo común denominador (PED) o formatos JSON personalizados para la transferencia de datos. Los modelos de datos básicos, como el estándar de HL7 FHIR, no son compatibles con los árboles genealógicos, pero existe un borrador de extensión para admitirlos que el estándar GA4GH tiene como objetivo evaluar y ampliar [\[33\]](#). La estandarización de la forma en que los sistemas representan la estructura familiar permitirá a los pacientes compartir esta información más fácilmente entre los sistemas de atención médica y ayudar a las herramientas de software a utilizar dicha información para mejorar el análisis y el diagnóstico del genoma.

El formato PED, anteriormente nombrado, es un archivo de texto simple con seis columnas: ID, un campo de sexo binario, el fenotipo (singular) y los genotipos SNP. Puede representar un trío padre-hijo básico, y eso puede cubrir muchos casos de uso. Sin embargo, no puede representar gemelos, cosas como adopción o donantes, embarazo, estado vital, fenotipos múltiples y procedencia de datos. Todo este tipo de datos es importante para el asesoramiento genético y las evaluaciones de riesgo, donde son valiosas las representaciones más ricas de las relaciones.

Pedigrí GA4GH incorpora de forma nativa PED para permitir la interoperabilidad con herramientas heredadas y a la vez amplia los atributos para el estudio genético. En la Figura 14⁶ podemos observar el modelo de GA4GH, el cual ha servido como base para el diseño de nuestra ampliación del modelo de OGMIOS.

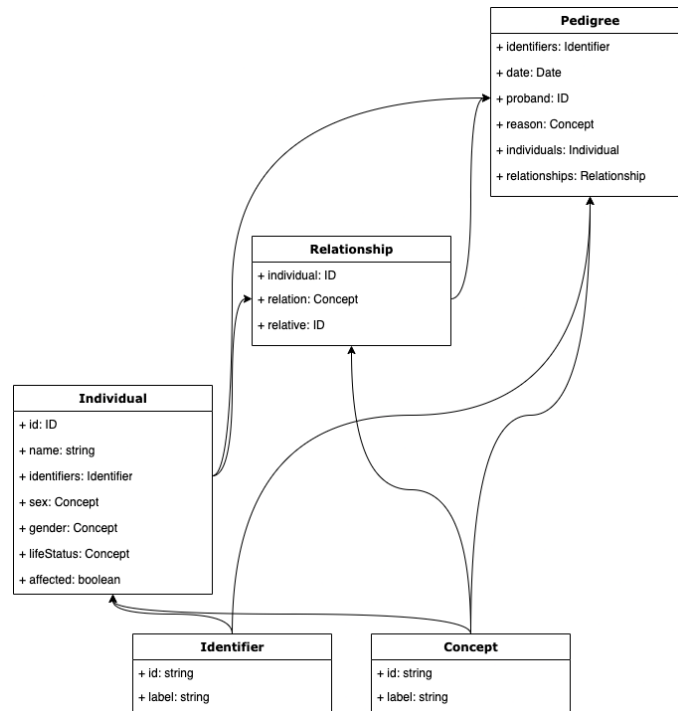


Figura 14: modelo conceptual Pedigree GA4GH

Observamos que el estándar GA4GH cuenta con cinco clases: *Pedigree*, *Relationship*, *Individual*, *Identifier* y *Concept*.

Por otro lado, nuestro borrador inicial, sin el estudio previo del estándar GA4GH, buscaba representar a cada familiar mediante la clase *Relative* y el grado de parentesco entre dos familiares como una clase de asociación *Relationship* que surge de la relación recursiva sobre las distintas instancias de cada familiar. La información que se debe incluir para cada familiar es diferente en función de que se trate de un paciente de oncología o un paciente de cardiología, por lo que cada instancia de la clase *Relative* tiene la posibilidad de tener asociada información adicional, representada por la clase *AdditionalData*, que puede ser específica de oncología (*OncoData*) o de cardiología (*CardioData*).

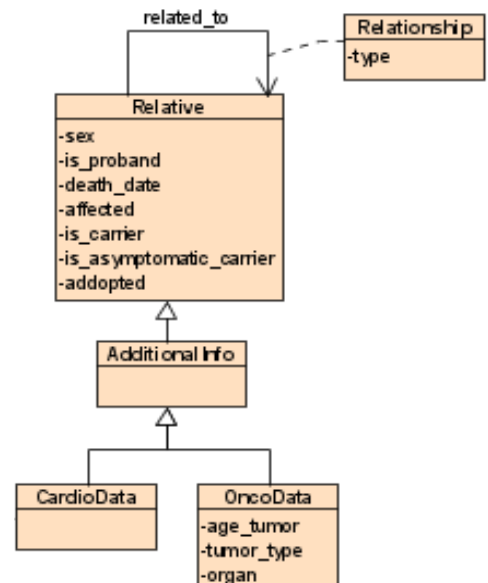


Figura 15: fragmento inicial modelo conceptual

⁶ Figura 14, imagen extraída de: <https://github.com/GA4GH-Pedigree-Standard/pedigree/>

Ya teniendo en cuenta tanto el estándar de Pedigree como nuestro boceto inicial, decidimos diferenciar la clase *Individual* del GA4GH en dos clases: paciente (*Patient*) y familiar (*Relative*), donde dependiendo de la clase a la que pertenezca el individuo contará con unos atributos u otros. Por otro lado, tanto la clase Identifier como Concept del estándar se van a incluir directamente como atributos de las clases relacionadas. Por último, la clase *Relationship* y la clase *Pedigree* se conservan, aunque en el caso de *Relationship* pasa a ser una clase de asociación.

Partiendo de este punto llegamos a la creación de los tres modelos candidatos a ampliación del modelo original de OGMIOS. El **primer modelo** propuesto fue el representado en la Figura 16, se observa que la clase *Patient* hereda de la clase *Relative*, ya que se puede decir que el paciente es un familiar, además de compartir los atributos que define a un familiar. En este modelo se encontraron inconvenientes ya que todos los atributos que pertenecen a *Relative* no los debe de tener el paciente, como, por ejemplo:

- el estado de vida y fecha de defunción, ya que el paciente debe de estar vivo,
- el estado de la enfermedad, debido a que el paciente no hay duda de que sufre la enfermedad,
- si es adoptado, ya que para hacer un estudio genético el paciente no debe de ser adoptado.

Por ello se llegó a la creación del **segundo modelo**, mostrado en la Figura 17, donde se ha suprimido la herencia de *Patient* a *Relative* y se creado una relación de asociación entre ellos indicando que un paciente puede tener muchos familiares. Aún así este modelo también cuenta con inconvenientes que pueden mejorarse, ya que se repiten muchos de los atributos en ambas clases, atributos que podrían reutilizarse como: el nombre, la edad, el sexo y la fecha de nacimiento.

Tras esto se diseñó el **tercer y último modelo** conceptual para la ampliación de la aplicación web del proyecto OGMIOS, representado en la Figura 18, que contempla la creación de genogramas. En este modelo definitivo se ha decidido crear una clase de asociación más genérica, *Person*, donde incluir los atributos comunes tanto a *Patient* como a *Relative*, de esta forma tenemos contenidos en la clase *Person* los atributos comunes: nombre, edad, sexo y fecha de nacimiento; y en las clases *Relative* y *Patient* los atributos específicos en caso de tratarse de paciente o de familiar.

Como se puede observar el modelo final, define un Pedigree formado por personas. Estos individuos pueden ser tanto pacientes como familiares, entre ellos están relacionados por un parentesco, además, dependiendo del grupo al que pertenezcan se tendrán en cuenta unas características u otras, aunque como se mencionado anteriormente cuentan con atributos comunes. A partir de este modelo actualizado de OGMIOS se va a proceder a la implementación del modulo web correspondiente al genograma familiar de cada paciente.

Diseño y desarrollo de una plataforma web para el estudio de cardiomiopatías familiares

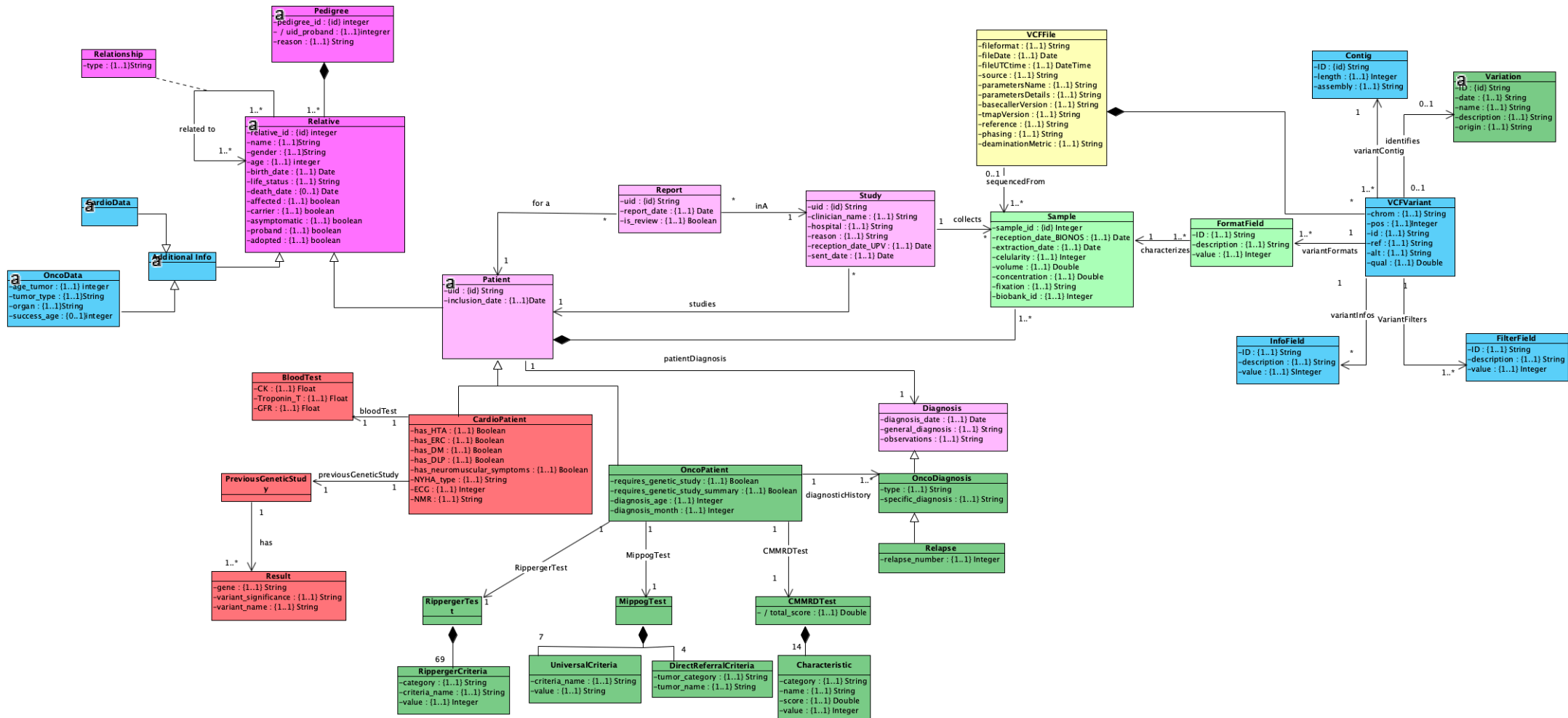


Figura 16: opción 1 ampliación modelo conceptual OGMIOS

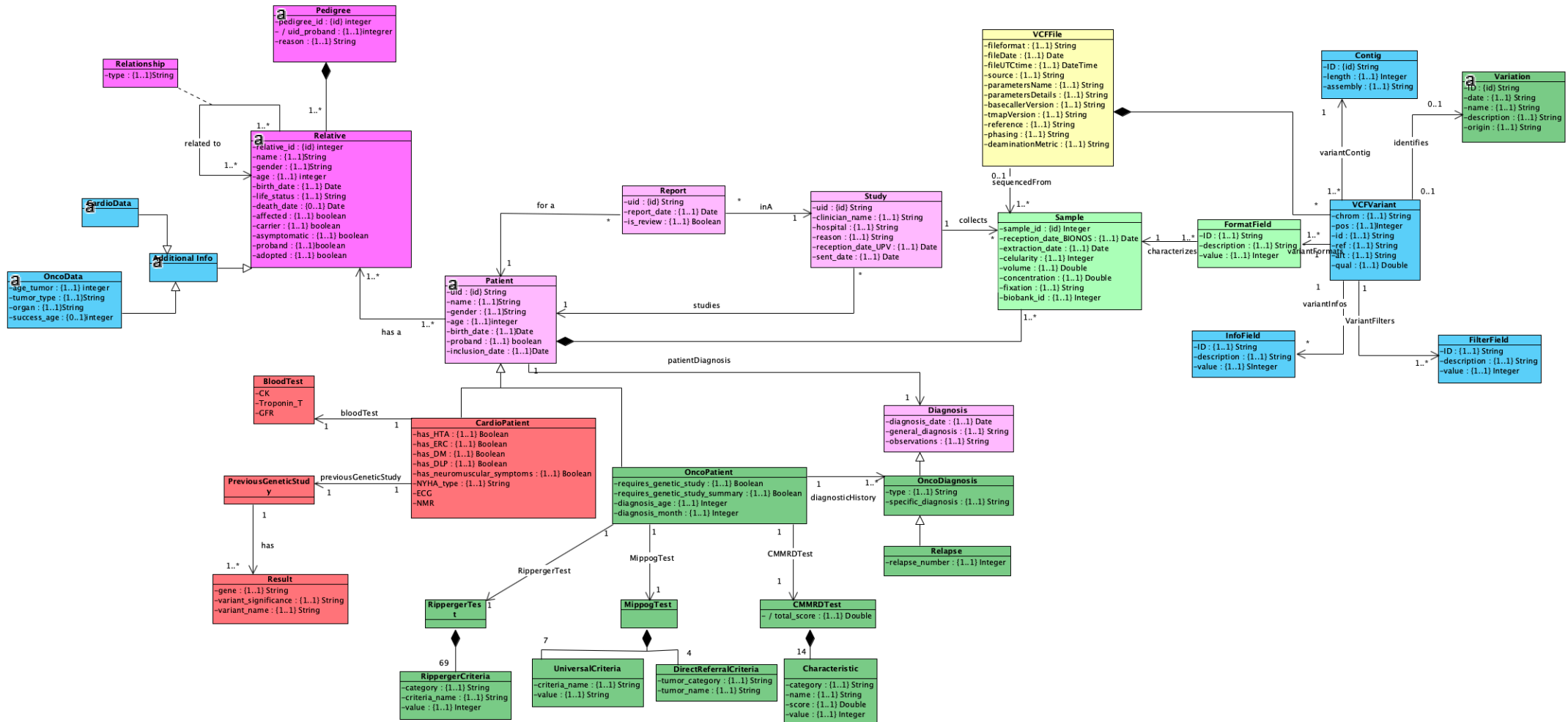


Figura 17: opción 2 ampliación modelo conceptual OGMIOS



Diseño y desarrollo de una plataforma web para el estudio de cardiomiopatías familiares

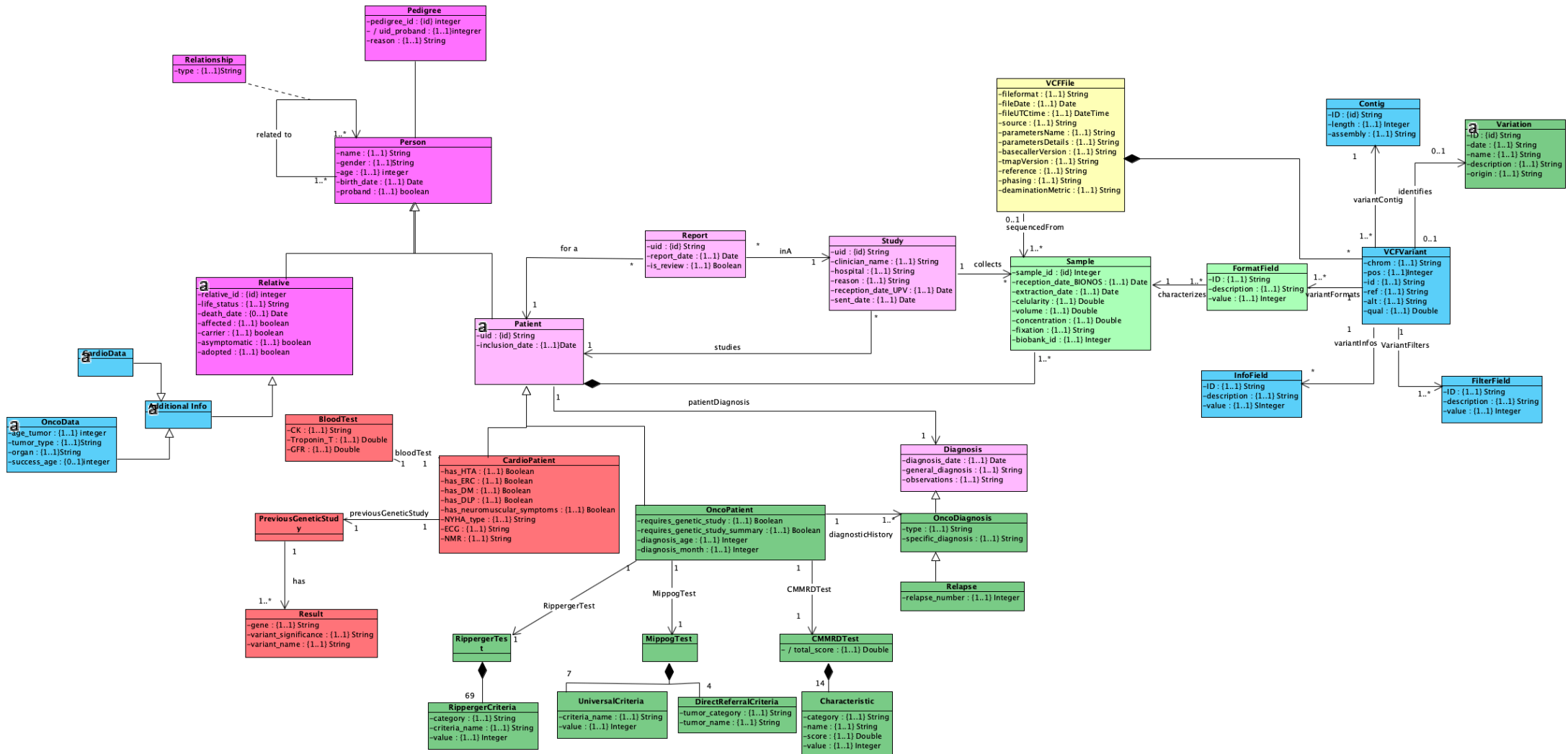


Figura 18: opción 3 y modelo conceptual final ampliación OGMIOS

3.2 Requisitos

Para el desarrollo correcto de la solución antes de comenzar con la implementación se ha llevado a cabo un estudio profundo de las necesidades tecnológicas que presentan los profesionales de los hospitales que forman parte del proyecto. Por ello se han tenido varias reuniones con los doctores:

- **Primera reunión** para la comprensión del contexto y el problema, además de la presentación del equipo clínico al frente: Detallamos con los profesionales de la salud en que punto se encontraban tecnológicamente, de dónde surgía el problema presentado, qué necesidades tenían, cómo creían que se podían cubrir dichas necesidades, que expectativas tenían del proyecto, qué alcance pensaban que tendría el desarrollo de este proyecto y qué aproximaciones de tiempos estimaban para las diferentes fases del proyecto, además de concretar la cantidad de datos clínicos que querían compartir para la aportación de conocimiento al proyecto.
- **Segunda reunión** para realizar las propuestas de ampliación del modelo conceptual de OGMIOS: Una vez habiendo analizado y entendido el contexto y el problema desarrollamos las tres propuestas de borradores para la ampliación del modelo conceptual, las cuales se mostraron a los médicos para su mejora y ampliación gracias a la aportación de una visión más clínica aclarando qué datos son necesarios recoger como atributos de las clases del modelo, tanto para el paciente como para los familiares.
- **Tercera reunión** para la validación del modelo definitivo de OGMIOS: Tras la mejora y perfeccionamiento del borrador del modelo propuesto en la anterior reunión, se eligió la propuesta final y se explicó a los médicos llegando a la aceptación y validación por parte de todos los miembros. Tras esta reunión se consiguió el modelo actual de OGMIOS, mostrado y explicado en el apartado anterior en la Figura [18](#), el cual contempla la creación de la plataforma para la visualización de genogramas.
- **Cuarta reunión** para la presentación de bocetos de la app por parte de los doctores, mostrando la idea de lo que esperaban como resultado final: Habiendo dado por finalizada la fase de modelado se comenzó con el diseño de la plataforma web, para ello sugerimos a los médicos que nos facilitaran unos bocetos a cerca de cómo les gustaría que se mostrara el estudio familiar mediante genogramas en la aplicación. Estos bocetos nos sirvieron como idea sobre qué conceptos visuales les serían más útiles y fáciles de utilizar, además de debatir a cerca de los problemas y limitaciones tecnológicas que encontramos en sus propuestas.

- **Quinta reunión:** para definir la funcionalidad: Tras haber comprendido qué esperaban los doctores de la aplicación y más concretamente de nuestro modulo web, se comenzó con la definición de las funcionalidades y el análisis de requisitos.

Para la realización de este análisis diferenciamos tres tipos de requisitos:

- **Requisitos de interfaz:** formato con el que la aplicación se comunica con su entorno.
- **Requisitos funcionales:** Servicios que el sistema debe proporcionar.
- **Requisitos no funcionales:** Restricciones que afectan al sistema.

Cada requisito se completa con los siguientes campos:

- **Identificador:** cada requisito está identificado de manera inequívoca. Como buen identificador es único además de distinguir la categoría a la que pertenece:

Para los requisitos de interfaz de usuario: **RIU-x**

Para los requisitos funcionales: **RF-x**

Para los requisitos no funcionales: **RNF-x**

- **Nombre:** indica la funcionalidad que presenta el requisito de manera escueta. Es identificativo.
- **Prioridad:** especifica la importancia que tiene dicho requisito. Puede ser: Alta, Media o Baja.
- **Descripción:** se detalla de manera breve las especificaciones que engloba el requisito.
- **Solución:** se propone de manera sencilla la manera de resolver el requisito.

Id	Nombre	Prioridad	Descripción	Solución
RIU-1	Usabilidad	Alta	Interfaz fácil e intuitiva que permita al usuario hacer uso de ella de forma cómoda	Usar colores predictivos para las plantillas y acciones correctamente especificadas
RIU-2	Ajuste en pantalla	Media	El genograma deberá ajustarse automáticamente para mostrarse completo en pantalla	Scroll del genograma

RIU-3	Navegación en pantalla	Alta	Se debe de poder desplazar por el genograma tanto en horizontal como en vertical	Barras laterales
RIU-4	Información del sistema	Media	El sistema debe de informar al usuario en cada acción de si ha sido completada con éxito o ha habido fallos	Implementación <i>snackbar</i> para indicar si se ha guardado correctamente en la base de datos
RIU-5	Uso del lenguaje del Usuario	Media	El sistema debe usar el lenguaje del usuario, de forma clara y con los términos adecuados	Español como idioma utilizado y uso de tecnicismos conocidos por los médicos
RIU-6	Color	Media	La interfaz del módulo de genograma seguirá la gama de colores de la aplicación	Aplicar a botones y cabeceras el color azul establecido como principal en la carpeta <i>Theme</i> del CRD
RF-1	Menú edición y visualización de datos	Media	Se debe de poder acceder a un menú para visualizar y editar los datos de cada individuo	Menú en cada nodo que enlace con la edición y visualización de los datos del individuo
RF-2	Obtención de datos individuo	Alta	Cada nodo debe de presentar una ficha asociada a los datos del individuo	Formulario rellenado con los datos del individuo
RF-3	Edición de datos individuo	Alta	Cada nodo debe presentar una ficha en la que poder editar los datos de cada individuo	Formulario con los campos definidos para rellenar los datos
RF-4	Libertad para eliminar o cancelar acciones	Alta	Cada acción tanto de adición de nodos como de adición de datos de cada individuo se debe poder cancelar	Botón para eliminar los nodos y cancelar la introducción de datos en cada nodo
RF-5	Actualización de datos	Alta	El sistema debe permitir al usuario actualizar y guardar los datos cambiados de las fichas de los individuos	Botón de guardar en la ficha de cada individuo
RF-6	Plantilla del nodo en función a características	Alta	El sistema mostrará la customización de cada nodo en función de las características seleccionadas	Se aplicarán a los nodos las plantillas definidas en función a las características que se seleccionen en las fichas de cada individuo

Diseño y desarrollo de una plataforma web para el estudio de cardiomiopatías familiares

RF-7	Buscador individuo por nombre	Baja	El sistema debe permitir encontrar un nodo mediante la búsqueda del nombre del individuo	Barra de búsqueda por nombre del nodo
RF-8	Avatar de individuo	Baja	El sistema mostrará un icono de avatar intuitivo y representativo para cada individuo en función de sus características	Icono asociado a cada ficha de paciente en función al sexo y edad
RF-9	Selección de estado de enfermedad excluyente	Media	El sistema debe de permitir la selección de una de las cinco características definidas como estado de enfermedad (probando, afectado, portador, asintomático o sano)	Mostrar un <i>select</i> con las cinco opciones en la ficha de cada individuo
RF-10	Género excluyente	Media	El sistema debe permitir la selección de uno de los dos sexos (hombre o mujer)	Mostrar un <i>select</i> con las dos opciones en la ficha de cada individuo
RF-11	Fechas con calendario	Baja	El sistema debe facilitar la introducción de datos con formato de fecha	Mostrar un calendario para la selección de fechas
RF-12	Adopción <i>tick</i>	Baja	El sistema debe mostrar de manera clara e intuitiva si el individuo es adoptado o no	Mostrar un <i>checkbox</i> para indicar adopción
RF-13	Identificar nodo por nombre	Alta	El sistema debe de permitir identificar cada nodo a que individuo corresponde	Mostrar el nombre del individuo asociado a cada nodo
RF-14	Minimización de árbol	Baja	El sistema tiene que minimizar el genograma dependiendo del espacio disponible	Icono en los nodos que tengan más ramas familiares, pero no se muestren por cuestión de espacio
RF-15	Adición de nuevos individuos	Alta	El sistema debe permitir al usuario la adición de nodos en función a los nodos ya creados: mujer, marido, hijos y/o padres...	Botón en cada nodo que permita crear nuevos nodos con parentesco
RF-16	Registro con nombre obligado	Alta	El sistema debe permitir el registro de nuevos nodos y de sus características siempre y cuando se registre el nombre	El sistema solo guardará y actualizará la información del nodo si ha registrado el nombre del individuo
RF-17	Leyenda disponible	Media	El sistema debe de facilitar toda la información necesaria para ayudar al usuario a comprender el código de colores y formas de las plantillas del genograma	Botón que muestre la leyenda del genograma

RF-18	Exportación a base de datos	Alta	El sistema debe de permitir al usuario guardar los datos del genograma	Botón de guardar que salve toda la información de los nodos
RNF-1	Persistencia	Alta	La implementación de los datos debe ser persistente y por eso estará almacenada en un base de datos	Cuando el usuario seleccione el botón de guardar el sistema exportará la información a la base de datos asociada
RNF-2	Acceso restringido	Alta	Solo los usuarios identificados podrán acceder al genograma de todos los pacientes	La opción del genograma solo estará disponible cuando el usuario se haya identificado

Tabla 2: especificación de requisitos

3.3 Implementación

3.3.1 Diseño

Tras la ampliación del modelo conceptual de OGMIOS y la especificación de requisitos se puede comenzar con la etapa de diseño de la solución, ya que nuestra implementación estará basada en ellos.

Como se ha indicado anteriormente en el apartado [2.4](#), la aplicación del Cuadernos de datos cuenta con la vista de Nuevo paciente, Listar pacientes y Detalles del paciente, en cuya vista, tras haber seleccionado el paciente, nos encontramos con las tres *tabs*: Diagnóstico, Muestras y Estudio Genético Previo, en cada cual se realiza una funcionalidad.

A la hora de realizar el diseño de la plataforma web decidimos añadir una cuarta pestaña a las *tabs* de la vista PatientDetails en la que visualizar el Genograma, como se muestra en la Figura [19](#). En esta pestaña implementamos el genograma con sus acciones.

A partir del modelo conceptual desarrollado y del análisis de requisitos realizado se ha realizado el diseño del módulo mediante *mockups*.

Dentro de la pestaña del genograma se muestra el genograma del paciente en el cual se puede interactuar añadiendo familiares, editando datos de las fichas, accediendo a la leyenda para comprender el significado de cada forma y color, visualizando la información de cada persona y guardando todos los cambios que se hayan realizado.

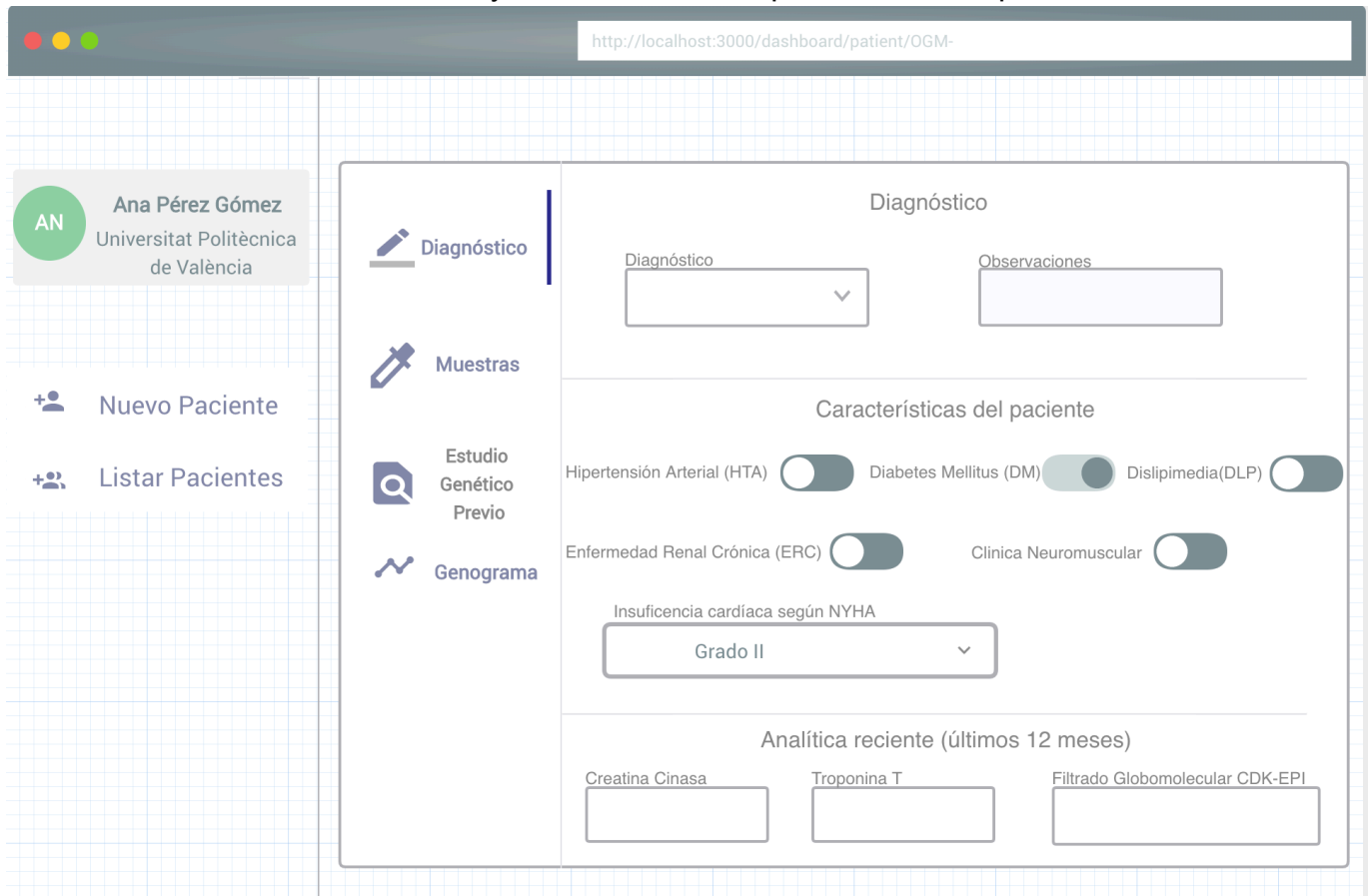




Figura 19: mockup adición pestaña Genograma

Para ello se han establecido el diseño de las plantillas de los nodos para representar el estado de cada familiar y paciente, todo ello en colaboración con los médicos además de apoyarnos en la definición de genograma y de su nomenclatura, nombradas en la Figura 12 del apartado 2.5.

Se decidió establecer las siguientes plantillas para los diferentes estados:

Característica	Opciones	Representación
Sexo	Mujer	
	Hombre	

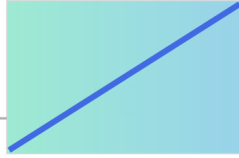

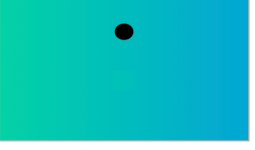


Estado de vida	Fallecido	
	Vivo	Sin elementos añadidos
Adoptado	Es adoptado	
Estado de enfermedad	Sano	Sin elementos añadidos
	Afectado	
	Portador	
	Asintomático	
	Probando	

Tabla 3: plantillas genograma

Cabe destacar que tanto el sexo, como el estado de enfermedad, el de vida y la adopción son complementarios, es decir, en cada nodo se añade una plantilla de cada característica según las opciones que cumpla el individuo. Sin embargo, dentro de las opciones de cada característica solo se puede cumplir con una, es decir, es excluyente. Por ejemplo, el estado de enfermedad solo puede cumplir con uno de los cinco estados definidos al igual que el sexo o el estado de vida.

Tras la definición de las plantillas se pasa a realizar los diseños de los mockups para las diferentes ventanas.

Para la correcta comprensión del desarrollo de los bocetos se han recuadrado cada componente destacable dándole asimismo un nombre para posteriormente indicar que requisitos y parte del modelo cubre cada elemento.

Comenzamos con la pantalla principal que en nuestro caso sería la pantalla que muestra el genograma completo registrado para el paciente seleccionado. En todo momento se han tenido en cuenta los requisitos y el modelo conceptual para el desarrollo del diseño, cumplimentando así todas las necesidades, además como se muestra en la Figura 20 se ha hecho uso de las plantillas definidas para cada estado como se ha explicado en la Tabla 3.

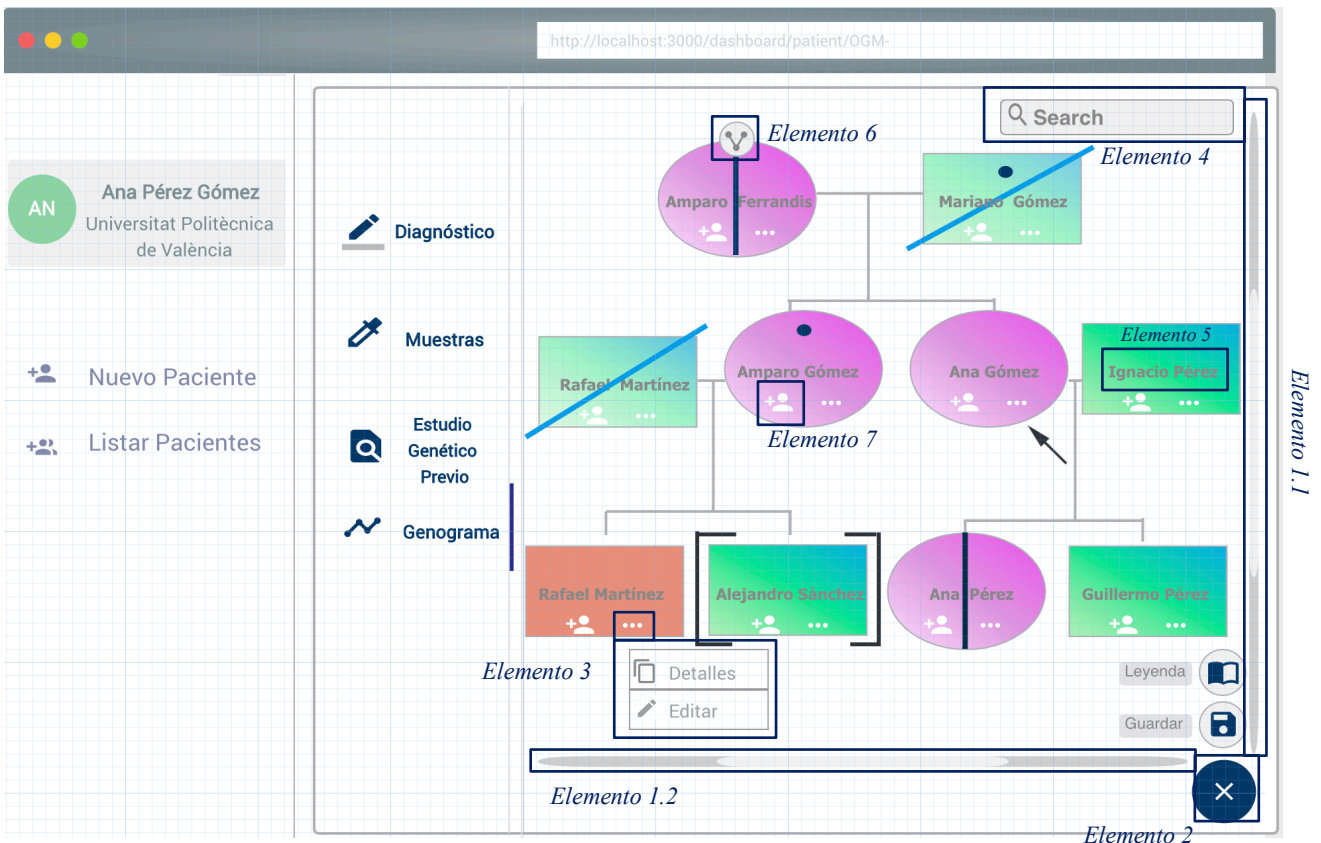


Figura 20: mockup vista general genograma

Como se puede observar en la Figura 21 en la página principal se ha añadido un botón expandible que permite recoger los botones de Guardar y de Leyenda, en los cuales, como indica su nombre, en el primero se salva toda la información del genograma y en el segundo se muestra una imagen descriptiva de las plantillas que ayuda a la correcta comprensión del genograma.

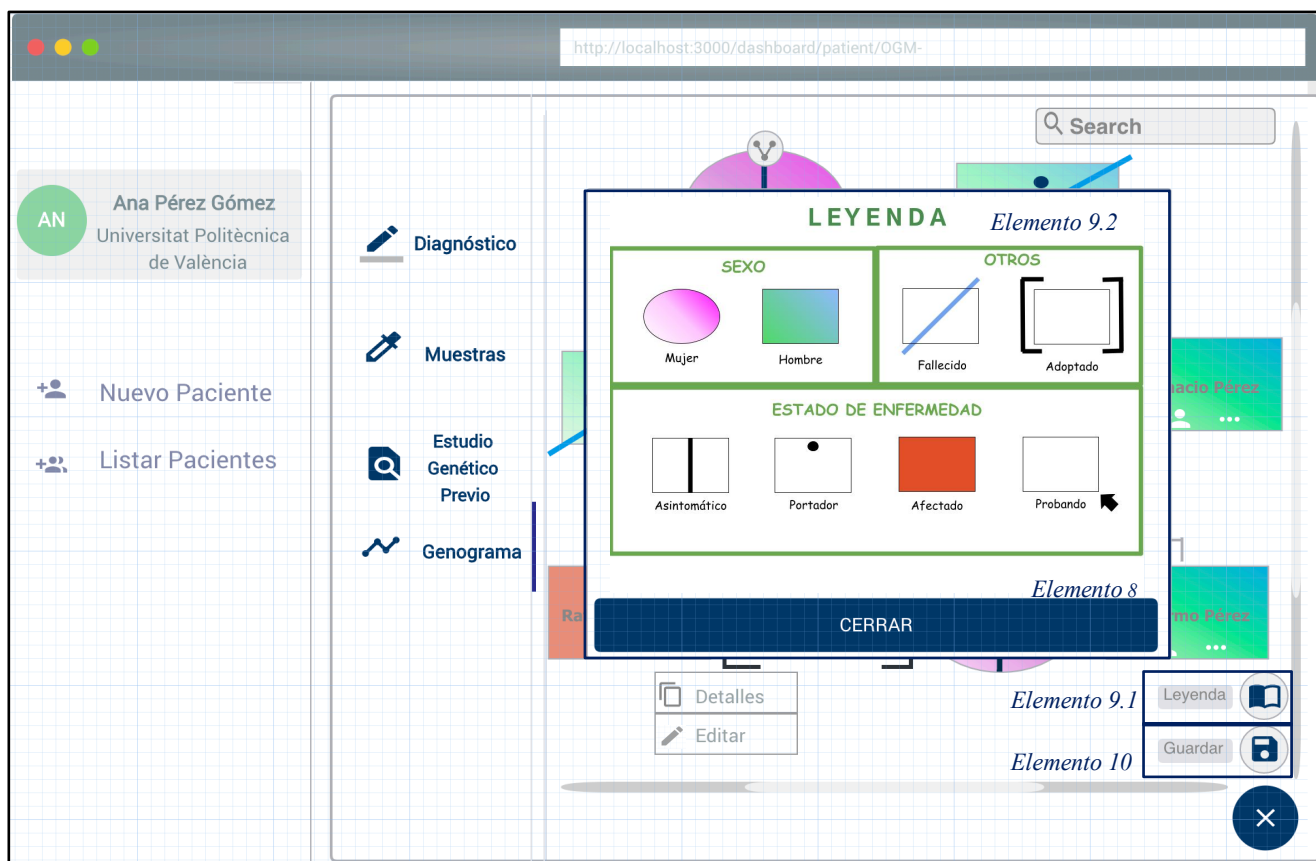


Figura 21: mockup vista leyenda genograma

La siguiente ventana diseñada, presentada en la Figura 22, corresponde con el formulario para cada nodo en el que se recoge la información necesaria de cada individuo, a este formulario se puede acceder tanto seleccionando la opción Editar (Elemento 3 Figura 20) como haciendo *click* encima de cada nodo.

Dentro del formulario nos encontramos con la opción de poder editarlo, actualizando los diferentes campos definidos, los cuales corresponden con los atributos establecidos en el modelo conceptual. Para el tipo de cada atributo del formulario se han elegido los más a fines en función a los requisitos definidos en el apartado 3.2.

Como se puede observar en el boceto se cuenta con un botón de eliminar formulario completo y por tanto también el nodo del árbol. Además, tiene un botón de cancelar en el que se cancelarán los cambios realizados y un botón de Guardar y Cerrar en el que se actualizarán los campos con la información y por ende también se actualizarán las plantillas del nodo en función a las características cambiadas.

Diseño y desarrollo de una plataforma web para el estudio de cardiomiopatías familiares

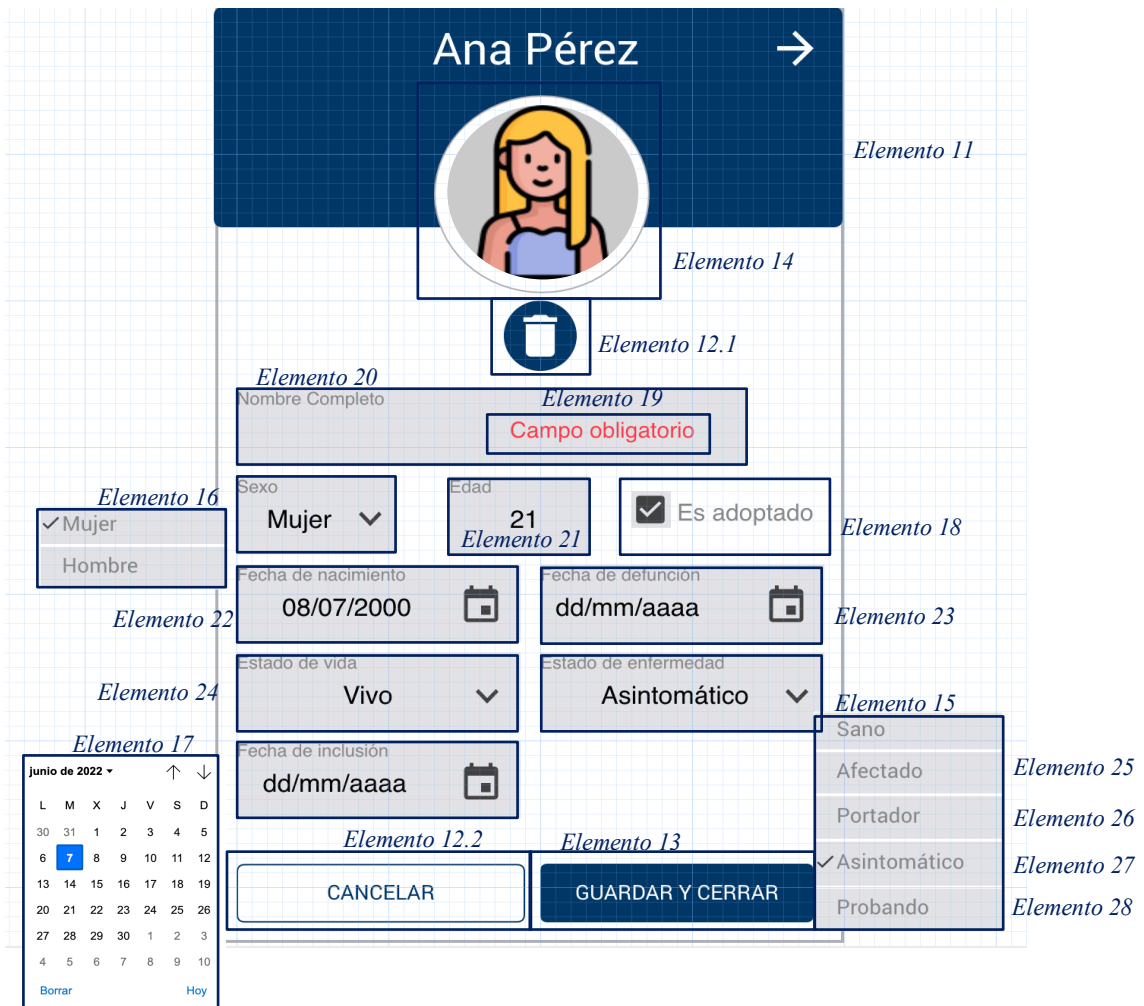


Figura 22: mockup vista formulario genograma

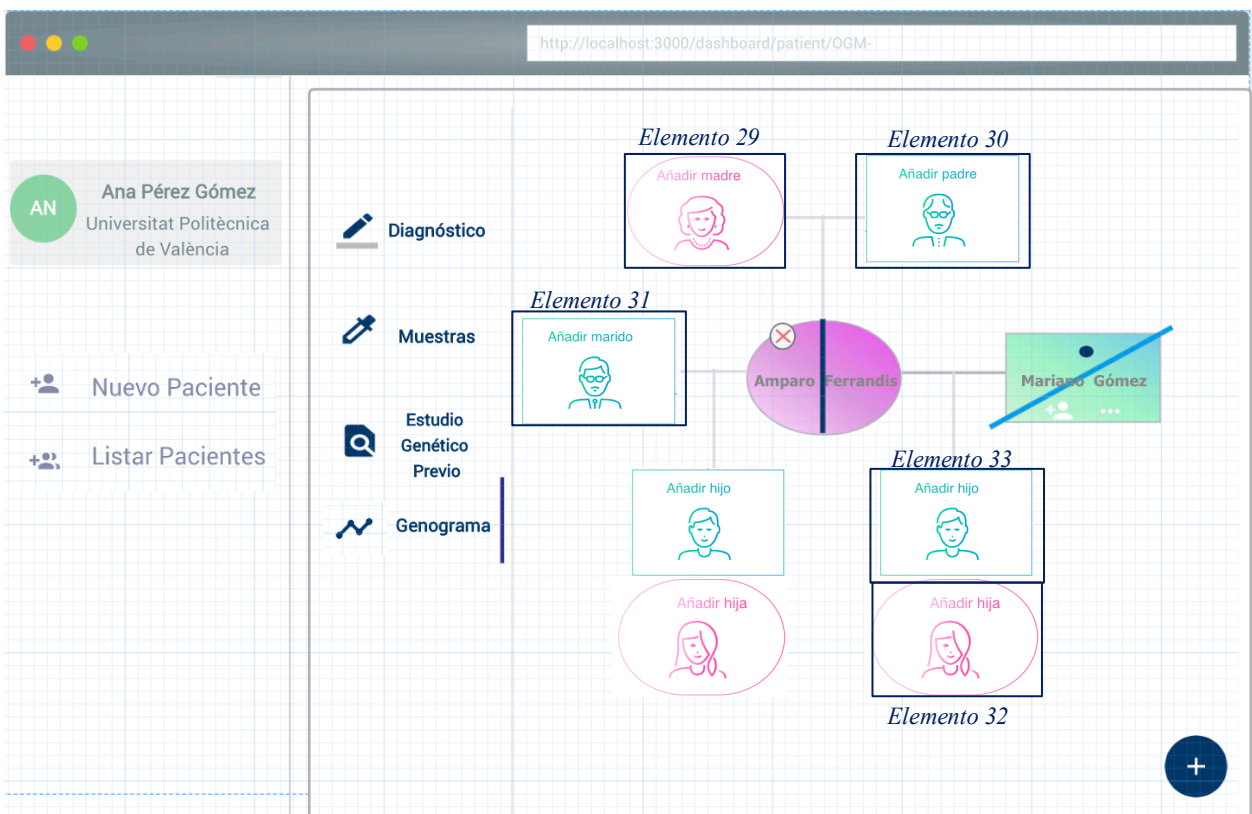


Figura 23: mockup vista adición de familiares

Asimismo, para la adición de familiares en función a un nodo se hace uso del botón correspondiente al elemento 7 de la Figura 20, dicho botón expande un menú que corresponde con el mockup de la Figura 23, facilitando así la adición de nodos.

Tras el diseño de todas las ventanas necesarias explicaremos la relación de cada elemento de los bocetos con la ampliación del modelo conceptual de OGMIOS, mostrado y explicado en la Figura 18 del apartado 3.1:

Clase Modelo Conceptual	Atributo MC	Tipo MC	Atributo Interfaz Usuario	Tipo IU	Opciones select IU	Figura mockup IU	Elemento mockup IU
Person	Name	String	Nombre Completo	String	-	Fig. 22	El. 20
	Gender	String	Sexo	Select	Mujer, Hombre	Fig. 22	El.16
	Age	Integer	Edad	Integer	-	Fig. 22	El. 21
	Birth_date	Date	Fecha de Nacimiento	Date	-	Fig. 22	El. 22
	Proband	Boolean	Estado de enfermedad	Select	Probando	Fig. 22	El.28
Affected	Boolean	Afectado			Fig. 22	El. 25	
Carrier	Boolean	Portador			Fig. 22	El. 26	
Asymptomatic	Boolean	Asintomático			Fig. 22	El. 27	
Relative	Life_status	String	Estado de Vida	Select	Vivo, Fallecido	Fig. 22	El. 24
	Death_date	Date	Fecha de defunción	Date	-	Fig. 22	El. 23
	Adopted	Boolean	Es adoptado	Check box	-	Fig. 22	El.18
	Inclusion_date	Date	Fecha de inclusión	Date	-	Fig. 22	El. 17
Relationship	Type	String	-	Nodo	Madre	Fig. 23	El. 29
					Padre	Fig. 23	El.30
					Mujer o Marido	Fig. 23	El.31
					Hija	Fig. 23	El. 32
					Hijo	Fig. 23	El. 33

Tabla 4: equivalencia Modelo conceptual e Interfaz de Usuario

Además, también vamos a relacionar cada requisito especificado en la Tabla 2 del apartado 3.2 y su solución propuesta con los elementos de los mockups.

Teniendo en cuenta que la columna Figura Mockup IU corresponde a la figura en la que se puede visualizar la solución propuesta en la interfaz de usuario y la columna Elementos Mockup IU corresponde con los componentes específicos de la figura que resuelven dicho requisito, se ha creado la siguiente tabla:

Id	Nombre	Solución	Figura Mockup IU	Elemento mockup IU
RIU-1	Usabilidad	Usar colores predictivos para las plantillas y acciones correctamente especificadas	Fig. 20, 21 y 22	---
RIU-2	Ajuste en pantalla	Scroll del genograma	Fig. 20	---
RIU-3	Navegación en pantalla	Barras laterales	Fig. 20	Elementos 1.1 y 1.2
RIU-4	Información del sistema	Implementación snackbar para indicar si se ha guardado correctamente en la BD	---	---
RIU-5	Uso del lenguaje del Usuario	Español como idioma utilizado y uso de tecnicismos conocidos por los médicos	Fig. 20, 21 y 22	---
RIU-6	Color	Aplicar a botones y cabeceras el color azul establecido como principal en la app definido en la carpeta Theme del CRD	Fig. 20, 21 y 22	Elementos 2, 8, 9.1, 10, 11, 12.1 y 13
RF-1	Edición y visualización de datos	Menú en cada nodo que enlace con la edición y visualización de los datos de dicho individuo	Fig. 20	Elemento 3
RF-2	Obtención de datos individuo	Formulario rellenado con los datos del individuo	Fig. 22	---
RF-3	Edición de datos individuo	Formulario con los campos definidos para rellenar los datos	Fig. 22	---
RF-4	Libertad para eliminar o cancelar acciones	Botón para eliminar los nodos y cancelar la introducción de datos en cada nodo	Fig. 20	Elementos 12.1 y 13
RF-5	Actualización de datos	Botón de guardar en la ficha de cada individuo	Fig. 21	Elemento 13

RF-6	Plantilla del nodo en función a características	Se aplicarán a los nodos las plantillas definidas en función a las características que se seleccionen en las fichas de cada individuo	Fig. 20	---
RF-7	Buscador individuo por nombre	Barra de búsqueda por nombre del nodo	Fig. 20	Elemento 4
RF-8	Avatar de individuo	Icono asociado a cada ficha de paciente en función al sexo y edad	Fig. 22	Elemento 14
RF-9	Selección de estado de enfermedad excluyente	Mostrar un <i>select</i> con las cinco opciones en la ficha de cada individuo (Sano, Portador, Afectado, Asintomático y Probando)	Fig. 22	Elemento 15
RF-10	Género excluyente	Mostrar un <i>select</i> con las 2 opciones en la ficha de cada individuo	Fig. 22	Elemento 16
RF-11	Fechas con calendario	Mostrar un calendario para la selección de fechas	Fig. 22	Elemento 17
RF-12	Adopción tick	Mostrar un <i>checkbox</i> para indicar adopción	Fig. 22	Elemento 18
RF-13	Identificar nodo por nombre	Mostrar el nombre del individuo asociado a cada nodo	Fig. 20	Elemento 5
RF-14	Minimización de árbol	Icono en los nodos que tengan más ramas familiares, pero no se muestren por cuestión de espacio	Fig. 20	Elemento 6
RF-15	Adición de nuevos individuos	Botón en cada nodo que permita crear nuevos nodos con parentesco	Fig. 20 y 23	Elemento 7
RF-16	Registro con nombre obligado	El sistema solo guardará y actualizará la información del nodo si ha registrado el nombre del individuo	Fig. 22	Elemento 19
RF-17	Leyenda disponible	Botón que muestre la leyenda del genograma e imagen con la leyenda disponible	Fig. 21	Elementos 9.1 y 9.2
RF-18	Exportación a base de datos	Botón de guardar que salve toda la información de los nodos	Fig. 21	Elemento 10

RNF-1	Persistencia	Cuando el usuario seleccione el botón de guardar el sistema exportará la información a la base de datos asociada	Fig. 21	Elemento 10
RNF-2	Acceso restringido	La opción del genograma solo estará disponible cuando el usuario se haya identificado	---	---

Tabla 5: equivalencia de requisitos con elementos mockup

Aquellos requisitos que se resuelven por implementación del código u otras formas que no incluyen elementos que se pueden solucionar mediante componentes de los *mockups*, tienen en la celda correspondiente un guion.

3.3.2 Implementación

La implementación de esta plataforma web se ha llevado a cabo en numerosos ficheros aislados para su posterior integración en la aplicación principal del CRD.

Los archivos implementados son: `defaultGenogram.js`, `templates.js`, `Genogram.js` y `SpeedDial.js`. Tanto `defaultGenogram.js` como `templates.js` se basa en la lógica de la librería utilizada, `BalkaFamilyTree`, la cual ha servido para desarrollar las plantillas y el tema estético del genograma.

Comenzando por **`templates.js`**, se han creado las plantillas para cada caso de estado de los individuos contemplando: el estado de vida, de enfermedad, la adopción y el sexo. Para la implementación de cada *template* nos hemos basado en las plantillas especificadas en la Tabla 3 del apartado 3.3.1. La aplicación de dichas plantillas a los nodos en función a las características se ha gestionado mediante *tags*, un método utilizado por la librería de `Balkan` que permite establecer plantillas específicas para nodos etiquetados, de tal manera que, dependiendo de los *tags* de un nodo se aplica una plantilla u otra.

Por otra parte, en **`defaultGenogram.js`** se ha definido cada elemento que forma el genograma: menús, barras laterales, botones de edición y eliminado, botones de adición de nodos, el formulario con sus campos y el tipo de estos, la barra de búsqueda, los títulos para cada nodo, la separación entre los nodos, la orientación y escala inicial del genograma, entre otras características que lo componen. Finalmente, dicha configuración del genograma se exporta para ser utilizada en el archivo `Genogram.js`.

Una vez habiendo implementado las plantillas y la configuración del genograma hemos creado **`Genogram.js`**, donde se desarrolla la interacción y función de este.

En el Fragmento de código [1](#) se crea el árbol en función a los nodos, la referencia y la función de actualización. Para la creación del árbol se etiqueta cada nodo proporcionado asociándole las *tags* definidas en *templates.js* además de añadir a dicho árbol las características definidas en *defaultGenogram.js*.

```
const createTree = (nodes, ref, onUpdateFunction) => {
  const taggedNodes = setNodeTags(nodes);
  const tree = defaultGenogram(taggedNodes, ref);
  tree.onUpdateNode(onUpdateFunction);
  return tree;
};
```

Fragmento de código 1: creación del árbol inicial

En el Fragmento de código [2](#) se hace uso de los *hooks* de *React* mencionados y explicados en el apartado del CRD [2.4](#). Como se puede observar el componente Genograma cuenta con un estado formado por tres variables: *genogram*, *needsUpdate* y *open*. **Genogram** utilizado para establecer y gestionar el estado del genograma, inicialmente con estado nulo, **needsUpdate** para controlar si el genograma necesita actualizarse o no, inicialmente con estado a *false* y **open** para saber cuando un componente esta abierto o no, inicializado a *false*. También usamos el *hook* **divRef** para obtener la referencia del genograma que se utiliza en los *useEffect*.

```
const Genogram = ({ nodes }) => {
  const [genogram, setGenogram] = useState(null);
  const [needsUpdate, setNeedsUpdate] = useState(false);
  const [open, setOpen] = useState(false);
  const divRef = useRef();
};
```

Fragmento de código 2: creación del genograma

Además, como se muestra en el Fragmento de código [3](#), se han definido las funciones: **onUpdate**, que gestiona el *hook* de *needsUpdate* poniéndolo a *true* gracias a su setter, **handleClose** que pone *open* a *false* mediante *setOpen*, **handleOpen** que funciona a la inversa que *handleClose*, estableciendo *open* a *true* y por último **handleSave** que guarda los nodos e imprime por pantalla la información de estos en formato JSON.

```
const onUpdate = () => {...};

const handleClose = () => {...};

const handleOpen = () => {...};

const handleSave = () => {...};
```

Fragmento de código 3: funciones manejadoras del genograma

En la implementación de este genograma se ha decidido definir dos **useEffect** distintos, ya que la lógica interna es similar a lo que nosotros buscamos, pero no actúa completamente en base a lo esperado. Nosotros queremos actualizar los nodos teniendo en cuenta la adición y eliminado de *tags* en función a la interacción del usuario, es por ello por lo que hemos decidido separarlo en dos *useEffect* dando simplicidad e incrementando la mantenibilidad del código.



El primer **useEffect** definido rerenderiza el genograma cuando los nodos se actualizan, es decir, se usa cuando se obtienen los nodos por primera vez creando el árbol inicial llamando a *createTree*, explicado anteriormente.

El segundo **useEffect** se ejecuta cuando la función *needsUpdate*, es decir, tras obtener los nodos de caso de haber modificado sus *tags* se debe de actualizar el árbol de nuevo. Para ello se comprueban las etiquetas que no han sido aplicadas de cada nodo para posteriormente llamar a *createTree* y volver a renderizar el genograma, pero esta vez con las etiquetas nuevas de los nodos.

```
useEffect(() => {...}, [nodes]);  
useEffect(() => {...}, [needsUpdate]);
```

Fragmento de código 4: hooks *useEffect* del genograma

Por último, el Fragmento de código 5 muestra el genograma comprendido en un *div* y el *SpeedAddDial* que implementa un botón expandible, desarrollado en el fichero **SpeedDial.js**, el cual puede comprender numerosos botones cuando se despliega, en nuestro caso, como se ha mostrado en el diseño, despliega dos botones: guardar y leyenda. La funcionalidad del *SpeedDial*, en concreto en la acción de mostrar la leyenda, se gestiona con el *open*, *handleOpen* y *handleClose*, ya que al seleccionar el botón correspondiente se abre un diálogo que muestra la imagen de la leyenda y que también cuenta con la opción de cerrar el diálogo. En igual forma, *handleSave* se ejecuta cuando el usuario selecciona el botón de guardar mostrando su contenido por consola y salvando los nodos.

```
return (  
  <>  
    <div id={"genogram"} ref={divRef} />  
    <SpeedAddDial  
      ...  
    ></SpeedAddDial>  
  </>  
);
```

Fragmento de código 5: devolución de la pestaña genograma

En la implementación se ha trabajado con **mockdata** ya que, para probar la correcta funcionalidad de la plataforma, así como la asignación de las *tags* y por tanto la esperada aplicación de las plantillas se requieren nodos con información que equivalen a la familia del paciente. Al no poder comprobar con estos datos, ya que como se ha mencionado, la implementación se ha llevado a cabo en archivos independientes, se ha hecho uso de los llamados datos simulados, creando manualmente en el código los nodos, con su nombre y el resto de las características.

3.3.3 Integración

El proceso de integración consta de dos partes: la adición del código implementado en el apartado anterior en la parte del *frontend* del CRD siguiendo la estructura previamente definida y la conexión del *backend*.

Atendiendo a la estructura de la aplicación principal, explicada en la sección [2.4](#) del CRD, se han movido los ficheros implementados de forma aislada a las carpetas correspondientes, dando lugar a la estructura mostrada en la Figura [24](#).

Comenzando por los ficheros *templates.js* y *defaultGenogram.js*, hemos creado dentro de la carpeta *lib*, que recoge todo el código e implementación de las distintas librerías, una carpeta llamada *genogram* que encapsula toda la lógica de *BalkanFamilyTree JS*.

Por la parte del fichero *Genogram.js*, se ha incluido en la carpeta *Pages* que contiene las diferentes vistas de la aplicación. Concretamente, dentro de *components* de la vista de *PatientDetails*, aquí nos encontramos con la carpeta *tabs* que recoge todas las pestañas de la aplicación, en esta ubicación se ha añadido una carpeta llamada *genogramTab* donde hemos añadido el fichero *Genogram.js*, haciendo posible así la vista del genograma que hemos implementado.

En cuanto al fichero *SpeedDial.js*, como implementa un componente que va a compartirse con otros componentes se ha decidido incluirlo en la carpeta *Components*, dentro de *shared* en la carpeta *buttons*, ya que implementa un botón con acción.

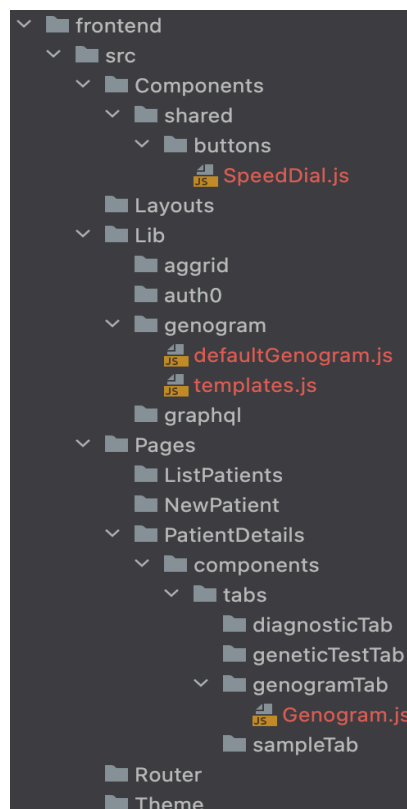


Figura 24: estructura de la aplicación con los archivos del genograma

Como se ha nombrado en el apartado anterior, en la implementación se ha hecho uso de los *mockdata*, pero en el punto de integrar el código los datos de los nodos del genograma se deben obtener en función de cada paciente. Por ello, es importante la correcta conexión a la base de datos de cada enfermo. Para esto, nos hemos introducido en la parte de consultas de *graphql* en el *backend* ubicadas en *schemas* de *graphql*, donde para obtener el genograma se ha: i) definido la *query* donde a partir de id del paciente se obtiene su genograma, ii) implementado el resolver donde se ha definido la función, **getGenogram**, que recibe: un objeto principal, argumentos, contexto de ejecución e información, y es responsable de devolver un resultado para el campo, en este caso *patient_id*, y iii) desarrollado la función en el driver utilizando *knex*, en el cual se filtra por el identificador del paciente y se obtiene el genograma por su nombre de columna.

```
exports.typeDefs = gql(`
  type Query {
    ...
    getGenogram(patient_id: String!): [GenogramPerson]!
  }
`);
```

Fragmento de código 6: backend - Consulta obtención genograma

```
getGenogram(parent, args, context, info) {
  return context.dataSources.mysqlDataSource.getGenogram(
    args.patient_id
  );
},
```

Fragmento de código 7: backend - Resolver de getGenogram

```
class MySQLDriver extends SQLDataSource {
  ...
  getGenogram = (patient_id) => {
    return this.knex("cardioDiagnosis")
      .where({ patient_id: patient_id })
      .select("genogram")
  };
};
```

Fragmento de código 8: backend - Driver getGenogram

Mientras que, para la actualización del genograma, como se ha explicado en el punto [3.3.2](#), se guarda mediante la opción de guardado del botón *SpeedDial*. Para su correcta integración, en la base de datos se han modificado en el backend tanto i) las *mutations*, donde a partir del paciente y el genograma se obtiene un boolean que indica si se ha actualizado correctamente el genograma en la base de datos o no, ii) los *resolvers* de *graphql* donde se ha implementado la función de **updateGenogram** que devuelve el id del paciente y el genograma correspondiente, iii) los drivers de *SQLDataSource* haciendo uso de la librería *knex* filtrando por el id del paciente y actualizando mediante *update()* el genograma.

```
exports.typeDefs = gql(`
  type Mutation {
    ...
    updateGenogram(patient_id: String, genogram: String!): Boolean!
  }
`);
```

Fragmento de código 9: backend - Mutation de actualización genograma

```
updateGenogram(parent, args, context, info) {
  return context.dataSources.mysqlDataSource.updateGenogram(args);
},
```

Fragmento de código 10: backend - Resolver de updateGenogram

```
class MySQLDriver extends SQLDataSource {
  ...
  updateGenogram = async ({patient_id, genogram}) => {
    try {
      await this.knex("cardioDiagnosis")
        .where({ patient_id: patient_id })
        .update ({ genogram: genogram})

      return true;
    } catch (e) {...}
  };
}
```

Fragmento de código 11: backend - Driver updateGenogram

Para finalizar con la integración y correcta conexión con la base de datos en el frontend se ha modificado el código. Para ello en cuanto a la obtención del genograma inicial del paciente se llama a la *query* definida en el *schema* de *graphql* del *backend* **getGenogram** con el parámetro *patient_id*. Con ello, además se actualiza la consulta de los pacientes de cardiología (*GET_CARDIO_PATIENT_DATA_BY_ID*) definida en las *queries* del frontend, donde se obtenía toda su información del paciente, haciendo que, de igual forma, se obtenga el genograma con todos sus atributos definidos.

```
export const GET_CARDIO_PATIENT_DATA_BY_ID = gql`
  query getPatientDataByID($patient_id: String!) {
    ...
    getGenogram(patient_id: $patient_id) {...}
  }
`;
```

Fragmento de código 12: frontend - Actualización consulta genograma

Por otro lado, para que se actualice correctamente el genograma se ha modificado ligeramente en el frontend el código de la función *handleSave()* en *Genogram.js*, que es la encargada de manejar el evento de guardado. Para ello en dicha función se llama a la *mutation* definida en el *schema* de *graphql* del *backend*, *updateGenogram*, con los parámetros: *patient_id* y *genogram*. Además, tras la llamada a la *mutation* se ha tenido en cuenta los dos errores que se pueden generar para la actualización, diferenciando entre el error de persistencia en la base de datos y el error de comunicación entre el *frontend* y *backend*.

```
export const UPDATE_GENOGRAM = gql`
  mutation UpdateGenogram($patientId: String, $genogram: String) {
    ...
    updateGenogram(patient_id: $patientId, genogram: $genogram)
  }
`;
```

Fragmento de código 13: frontend - Mutation de exportación genograma

Para la completa implementación de la persistencia se ha modificado también el script de la base de datos, añadiendo el atributo del genograma.

```
CREATE TABLE IF NOT EXISTS `ogmios_crd`.`cardioDiagnosis` (  
  ...  
  `genogram` LONGTEXT,  
  ...  
)
```

Fragmento de código 14: actualización script Base de datos tabla CardioDiagnosis

3.4 Workflow

En este apartado únicamente se va a mencionar la estrategia seguida en el sistema de control de versiones. Para ello, en GitLab se creó al comienzo del proyecto OGMIOS un grupo del CRD compuesto por: la base de datos, el *frontend* y el *backend*. A partir de este punto, para el desarrollo de este TFG se abrió una nueva carpeta llamada *genogram*, además de las tres ya mencionadas. Tras subir todo el código a *genogram*, se creó a partir de las *merge request*, una rama nueva tanto en el *frontend* como en el *backend*. De esta forma se ha ido introduciendo y actualizando el código correctamente. Para finalizar el proceso se han cerrado ambas ramas. El esquema de *workflow* final se puede observar en las figuras.

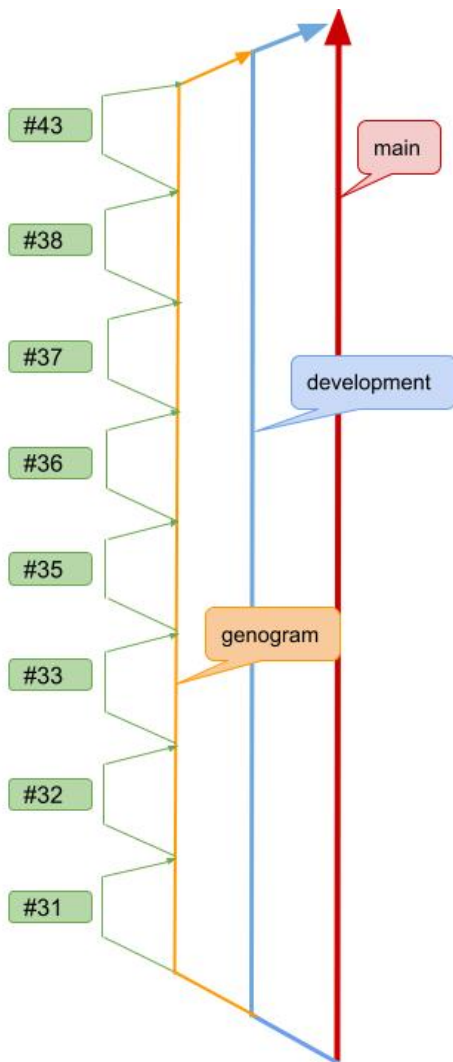


Figura 26: frontend - Ramas del workflow del proyecto

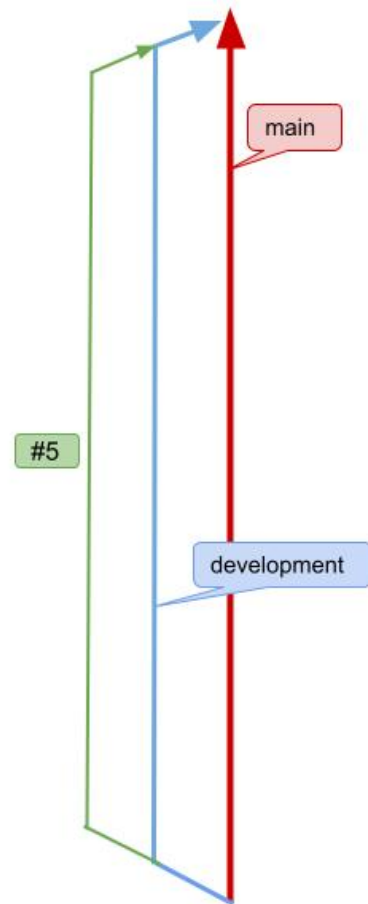


Figura 26: backend - Ramas del workflow del proyecto

En el proyecto se ha partido de la rama Development, ya creada sobre la Main. Tras estas se ha creado la rama Genogram, en el caso del frontend, sobre la que se ha ido abriendo una rama por cada issue a resolver. En el caso del backend, el trabajo no era muy extenso y se ha dado solución a toda la persistencia en la base de datos en una sola rama, por ello, se ha abierto únicamente la issue #5. Cuando ya se ha completado cada issue y por tanto cerrado todas las ramas correspondientes a estas, se ha dado por implementado completamente el genograma y por ello se ha mergeado a la rama Development y esta se ha fusionado a Main que es la rama de producción.

Los *Issues* que se han tenido que solucionar se han dado debido a errores o perfeccionamiento de los campos implementados en nuestro código. Esto ha sido una forma de poder mejorar y completar el desarrollo de la plataforma. Asimismo, estos *issues* se han integrado en un *Epic* que los agrupa permitiéndonos tener una vista más general. Estos *Issues* que han surgido y posteriormente se han resuelto han sido:

- Issues del frontend:

- **#31 Add Floating Buttons:** este issue implementa la funcionalidad del SpeedDial.
- **#32 Add healthy template:** añadir al *select* de las opciones del estado de la enfermedad la opción: sano.
- **#33 Change checkbox color:** oscurecer el color del *checkbox* de adopción para una mejora de visualización.
- **#35 Increase width of edit menu:** aumentar el tamaño del menú del formulario para que encajen los campos correctamente.
- **#36 Remove img attribute:** eliminar del formulario el campo de la url imagen.
- **#37 Proband template error:** solucionar el error con la plantilla del probando.
- **#38 There is not partner template:** solucionar el error con la plantilla de adición de marido/mujer tras seleccionar el botón de añadir nodos.
- **#43 Add mutation to send stringified genogram:** añadir la llamada a *mutation* para que al guardar el genograma este se actualice en los datos del paciente con formato de *string*.

- Issues del backend:

- **#5 Save genogram:** Crear *graphql stuff* en *schema*, en resolver imprimir genograma por pantalla e implementar la conexión con la base de datos.



CAPÍTULO 4:

Validación de la Solución

Tras la integración del módulo web en la aplicación se ha actualizado la herramienta que está desplegada para que incluya el genograma. Para validar la solución hemos tenido una serie de reuniones con los cardiólogos del Hospital La Fe, el Hospital Clínico y el Hospital Alicante en las cuales se les ha explicado el funcionamiento del módulo y se les ha invitado a probar la aplicación con la funcionalidad del genograma.

Para ello han seguido las acciones indicadas para probar así las operaciones básicas que implementa nuestra plataforma web, estas operaciones básicas que han realizado se recogen y explican en la Tabla [6](#).

Nº Paso	Acción
1	Se ha creado un paciente
2	Se ha accedido al apartado de genograma
3	A partir del nodo del paciente se ha añadido el padre
4	Se ha añadido la madre del paciente
5	Se han añadido cuatro hijos más a este padre y esta madre, que son los cuatro hermanos del paciente
6	Se ha añadido marido y mujer a los hermanos del paciente y al propio paciente
7	Se han creado hijos del paciente con su mujer
8	Se han creado más nodos en función a la madre del paciente para conseguir la minimización de esa rama del árbol
9	Se han rellenado las edades de los nodos para que se mostraran los iconos de avatar en función a su edad y sexo
10	Se ha especificado que el paciente es el probando
11	Se ha indicado que el padre del paciente estaba muerto en el estado de vida
12	Se ha indicado que un hermano del paciente es adoptado y asintomático

13	Se ha indicado que la madre del paciente es portadora
14	Se ha indicado que la hermana del paciente esta afectada por la enfermedad
15	Se ha eliminado el nodo del hermano del paciente
16	Se ha buscado en la barra de búsqueda el nombre del paciente para que la aplicación lo ubicara en el árbol
17	Se ha cambiado el sexo de uno de los hermanos creado del paciente a mujer
18	Se ha cambiado la edad de uno de los hijos del paciente
19	Se ha añadido la fecha de nacimiento del probando
20	Se ha visualizado los datos de la mujer del paciente
21	Se ha visualizado la leyenda para consultar el código de plantillas
22	Se ha guardado el genograma en la Base de Datos

Tabla 6: acciones probadas en la plataforma web

Una vez realizada todas estas acciones realizamos un *focus group* donde los médicos nos han dado su *feedback* que se resume a continuación.

La aplicación ha sido de su agrado, les parece muy atractivo visualmente y muy intuitivo gracias a la especificación de cada acción. Además, han destacado la forma tan sencilla de añadir familiares mediante un simple botón en cada nodo, la manera de rellenar los formularios también les parece muy simple gracias a los tipos de los campos. Por otro lado, agradecen que haya una leyenda que les permita saber e identificar las plantillas con las características. Por último, han destacado la forma tan fácil y accesible de guardar y de obtener los datos, ya que principalmente para guardar es seleccionar un botón y para la carga de los datos simplemente hay que acceder a la pestaña del genograma dentro de cada paciente. Por todo ello han quedado muy satisfechos.

Asimismo, tras el *feedback* recibido, la aplicación web se encuentra disponible y en uso para los médicos que forman parte del proyecto. Por tanto, ha pasado a ser una aplicación de explotación dentro de OGMIOS y un sistema de práctica diaria para los médicos, con la cual se gestiona el estudio genético familiar a partir de la creación y visualización del genograma de los pacientes, resolviendo así una de las necesidades principales del proyecto OGMIOS.

CAPÍTULO 5:

Conclusiones

Tal y como hemos podido comprobar a lo largo de esta tesis, se ha alcanzado el objetivo principal de este TFG: desarrollar un módulo web para la creación y visualización de genogramas. De este objetivo surgieron cinco preguntas de investigación, las cuales vamos a resolver a continuación gracias al desarrollo del proyecto.

PI 1. ¿Qué tecnologías se pueden utilizar para el desarrollo del artefacto?

En la sección [2.6](#) de esta tesis se han revisado las tecnologías actualmente disponibles, GoJS y Balkan FamilyTree JS, a partir de las cuales se ha recogido en una tabla comparativa todas sus características y se ha llevado a cabo un análisis *Fit and Gap* que nos ha permitido seleccionar la librería de Balkan FamilyTree JS como la mejor opción para el desarrollo de nuestro artefacto.

PI 2. ¿El modelo actual del proyecto OGMIOS da soporte al módulo a desarrollar?

Como se puede observar en la sección [2.3](#), en la cual se ha presentado el modelo actual de OGMIOS, este no da soporte a la plataforma web que implementa el genograma, es por ello por lo que se ha realizado una ampliación de este, la cual se ha tratado en la sección [3.1](#) de este TFG.

PI 3. ¿Cuáles son los requisitos del módulo a desarrollar?

Los requisitos de este proyecto se diferencian en requisitos de interfaz de usuario, requisitos funcionales y no funcionales, algunos de ellos son: usabilidad, libertad de acción para el usuario y persistencia, entre otros. Para más detalle de los requisitos específicos a nivel individual se han recogido en la sección [3.2](#).

PI 4. ¿Cómo implementar un genograma?

En la sección [3.3](#) se ha explicado como a partir del diseño del módulo web se ha comenzado el proceso de desarrollo e implementación del genograma en un proyecto aparte para posteriormente ser integrado en la aplicación principal. Para ello se ha seguido una metodología de uso de control de versiones en la cual a partir de la rama principal y la de desarrollo se ha ido introduciendo el código corregido mediante las diferentes ramas creadas para resolver los *Issues*. De esta forma se ha logrado la correcta implementación del artefacto correspondiente al genograma.

PI 5. ¿Resulta de utilidad para los profesionales de la salud?

Sí, resulta de utilidad, esto ha sido comprobado gracias a las reuniones realizadas con los médicos donde tras las mismas hemos recibido el *feedback* de los usuarios. Todo ello ha sido tratado en detalle en el [Capítulo 4](#) de esta tesis.

Durante todo el proceso de investigación, desarrollo e implementación necesarios para responder a las preguntas de investigación podemos observar, tras un periodo de reflexión y asentamiento de la información, que todo el conocimiento adquirido se puede agrupar en tres ejes: Tecnológico, Metodológico y Científico.

Por un lado, enfocándonos en el eje tecnológico, el desarrollo de esta plataforma web ha permitido la implementación de las tres capas, permitiendo así el aprendizaje de tecnologías nuevas como React en el *frontend*, Node en el *backend* y GraphQL como lenguaje de consulta para API. Además de poder poner en práctica las tecnologías aprendidas a lo largo de la carrera. Todo ello, ha permitido aportar una visión holística de cómo funciona un proyecto en un entorno real. De igual forma, el hecho de incorporar la plataforma web en una aplicación en desarrollo ha propiciado el trabajo en equipo y el pensamiento de grupo.

A nivel personal, a lo largo de los cuatro años del grado y el estudio de todos los ámbitos que este recoge, la parte donde menos seguridad sentía era en el desarrollo web. Es por ello por lo que, esta tesis me ha ayudado a perfeccionar dicho aspecto y de esta manera cerrar mi carrera universitaria con una mayor plenitud de conocimientos, ya que, el área dónde se ha concentrado mayor trabajo ha sido en el proceso de desarrollo de la interfaz, por lo que ha sido donde más esfuerzos se han requerido y mayor tiempo de dedicación se ha aplicado. También es importante destacar el aprendizaje global que me ha aportado el hecho de desarrollar mi TFG en un marco de un proyecto real con un grupo de profesionales que provienen de diferentes ámbitos. A nivel tanto personal como profesional me ha dotado de capacidades para adentrarme y capacitarme a la hora de formar parte de proyectos en el mundo laboral.

En cuanto al eje metodológico, ha permitido conocer *Design Science*, que es una metodología que se puede aplicar en numerosos ámbitos. Cara al futuro este conocimiento nos ha aportado una base sólida para el desarrollo de los próximos proyectos, investigaciones y estudios que se vayan a realizar. En mi caso me da ventaja en el ámbito investigador a la hora de realizar un máster y si realizo un doctorado, ya que con esta tesis me he familiarizado en la aplicación de una metodología importante y gracias a ello me ha permitido adquirir experiencia en esta área de dominio.

Por último, este trabajo se engloba en el ámbito científico de un proyecto real que reporta mejoras potenciales a la salud, ayudando a los profesionales a aplicar una medicina de precisión en función a cada paciente y dando lugar a la mejora de la calidad de vida de las personas. Por mi parte, al igual que la

mayoría de las personas, he tenido una base, adquirida etapas educativas anteriores, de: genética, proteínas, enfermedades, etc., pero esta tesis me ha permitido la ampliación de estos conocimientos aportando un gran respaldo en el ámbito biológico como, por ejemplo: el entendimiento de los archivos VCF, el tratamiento de datos genómicos o la investigación sobre las enfermedades cardiológicas. Además de permitirme conocer la gestión de hospitales y las necesidades que se generan gracias al trabajo en conjunto con los médicos. Todo ello me motiva a seguir trabajando en este contexto, es por esto por lo que mi intención actual es comenzar una etapa de investigación junto con el grupo PROS del instituto Vrain como extensión al trabajo realizado y al proyecto OGMIOS.

Como resultado, por parte del modelado conceptual, aparte de ayudar para abstraer el problema e identificar como interactúa el sistema en el cual se desenvuelve la solución. Como se ha podido observar, el modelo ha estado presente en los tres ejes gracias a tratarse de un proyecto *conceptual model based*. En el eje tecnológico para el correcto modelado de la base de datos, para la inserción de conexiones con GraphQL o la implementación del *frontend*. En el metodológico, el modelo conceptual propicia a construir explícitamente las ideas y conceptos que marca la metodología. En el científico ha servido de base para realizar el esquema y plantear el dominio. Por ello, además de permitirme poner en práctica mis conocimientos sobre el modelado conceptual, he comprobado la importancia de invertir tiempo en la realización de este y que sirva como base para el desarrollo de un proyecto.

5.1 Trabajos Futuros

Tras el desarrollo de este TFG se finaliza la primera versión del prototipo del modulo web para la visualización y edición de genogramas. A partir de él se podrán realizar mejoras y modificaciones tanto de su apariencia como de su interacción y procesos dependiendo de las nuevas necesidades y/o requisitos que surjan por parte de los *stakeholders*.

Por otra parte, al tratarse de una primera versión se ha tenido en cuenta los aspectos más imprescindibles y básicos para el estudio de las enfermedades de tipo hereditaria como: la edad, el estado de la enfermedad que ocupa el estudio, el sexo o el estado de vida. Sin embargo, se espera poder ampliar el número de plantillas del genograma añadiendo estados nuevos como enfermedades que hayan podido sufrir los familiares diferentes a la enfermedad principal y que puedan estar relacionadas con esta o como estados referentes a la reproducción: abortos, esterilidad, embarazo, etc. Además, también se pretende tener en cuenta diferentes relaciones entre los familiares y los pacientes, como por ejemplo el hecho de ser gemelos, contemplar estados emocionales como divorcios u otras opciones que puedan influir lo más mínimo en la enfermedad estudiada.

También es importante para las futuras versiones incluir la posibilidad de relacionar genogramas entre pacientes, teniendo en cuenta que se puede dar

el caso de tener enfermos de la misma familia tanto con la misma enfermedad como con distintas, de esta forma se minimizaría el trabajo ya que se trataría de árboles genealógicos iguales o similares.

Cabe destacar que a partir de este modulo web como objetivo a corto plazo se quiere incluir al resto de enfermedades que trata el proyecto OGMIOS y que están presentes en la aplicación del CRD, en este caso la oncología pediátrica. Pero además tras su aplicación y puesta en marcha se espera en el futuro, como objetivo alineado al proyecto OGMIOS, que este modelo de plataforma web sea aplicable a otras áreas clínicas.

Bibliografía

[1] «El código del genoma humano por fin está completo», *www.nationalgeographic.com.es*, 1 de abril de 2022. (accedido 22 de abril de 2022). Disponible en: https://www.nationalgeographic.com.es/ciencia/codigo-genoma-humano-por-fin-esta-completo_18097

[2] S. Goodwin, J. D. McPherson, y W. R. McCombie, «Coming of age: ten years of next-generation sequencing technologies», *Nat. Rev. Genet.*, vol. 17, n.º 6, Art. n.º 6, jun. 2016, doi: 10.1038/nrg.2016.49.

[3] J. A. Reuter, D. V. Spacek, y M. P. Snyder, «High-Throughput Sequencing Technologies», *Mol. Cell*, vol. 58, n.º 4, pp. 586-597, May 2015, doi:10.1016/j.molcel.2015.05.004. Disponible en: <https://pubmed.ncbi.nlm.nih.gov/26000844/>

[4] M. R. Kosorok y E. B. Laber, «Precision Medicine», *Annu. Rev. Stat. Its Appl.*, vol. 6, n.º 1, pp. 263-286, 2019, doi: 10.1146/annurev-statistics-030718-105251. Disponible en: <https://pubmed.ncbi.nlm.nih.gov/31073534/>

[5] I. R. König, O. Fuchs, G. Hansen, E. von Mutius, y M. V. Kopp, «What is precision medicine?», *Eur. Respir. J.*, vol. 50, n.º 4, p. 1700391, oct. 2017, doi: 10.1183/13993003.00391-2017.

[6] J. M. Brea, «Medicina y Melodía: El genograma», *Medicina y Melodía*, 4 de octubre de 2019. (accedido 22 de abril de 2022). Disponible en: <https://medymel.blogspot.com/2019/10/el-genograma.html>

[7] *Design Science Methodology for Information Systems and Software Engineering*. Accedido: 14 de junio de 2022. [En línea]. Disponible en: <https://link.springer.com/book/10.1007/978-3-662-43839-8>

[8] «Cardiomiopatías». (accedido 26 de abril de 2022). Disponible en: <https://www.bimodi.com/cardiomiopatias/>

[9] J. A. Olmos *et al.*, «CARDIOMIOPATÍAS CAPÍTULO VII», p. 61.

[10] «Autosómico dominante | NHGRI», *Genome.gov*. (accedido 29 de abril de 2022). Disponible en: <https://www.genome.gov/es/genetics-glossary/Autosomico-dominante>

[11] «Miocardiopatía Dilatada (MCD) – Swiss DNAlysis». (accedido 29 de abril de 2022). Disponible en: <https://www.swissdnalysis.ch/cardiologia/miocardiopatia-dilatada-mcd/?lang=es>

[12] «Consenso-de-Miocardiopatía-Hipertrofica-completo.pdf». Accedido: 29 de abril de 2022. [En línea]. Disponible en: <https://www.sac.org.ar/wp-content/uploads/2014/04/Consenso-de-Miocardiopatía-Hipertrofica-completo.pdf>

[13] «Cirugía Cardiovascular - Miocardiopatía Hipertrofica», *Cirugía Cardiovascular*. (accedido 29 de abril de 2022). Disponible en: <https://cirugiacardiovascular.com.mx/operacion-a-corazon-abierto/enfermedades-cardiovasculares/miocardiopatia-hipertrofica/>

[14] «Síndrome de QT Largo (SQTL) – Swiss DNAlysis». (accedido 29 de abril de 2022). Disponible en: <https://www.swissdnalysis.ch/cardiologia/sindrome-de-qt-largo-sqtl/?lang=es>

[15] «Síndrome de QT largo - Síntomas y causas - Mayo Clinic». (accedido 29 de abril de 2022). Disponible en: <https://www.mayoclinic.org/es-es/diseases-conditions/long-qt-syndrome/symptoms-causes/syc-20352518>

[16] «El cáncer infantil». (accedido 29 de abril de 2022). Disponible en: <https://www.who.int/es/news-room/fact-sheets/detail/cancer-in-children>

[17] «B1.05-Texto.pdf». Accedido: 29 de abril de 2022. [En línea]. Disponible en: <https://www.uv.es/rnti/pdfs/B1.05-Texto.pdf>

[18] Pastor Ó, León AP, Reyes JFR, García AS, Casamayor JCR. Using conceptual modeling to improve genome data management. *Brief Bioinform.* 1 de enero de 2021;22(1):45-54. Disponible en: <https://doi.org/10.1093/bib/bbaa100>

[19] García S. A, Palacio AL, Reyes Román JF, Casamayor JC, Pastor O. Towards the Understanding of the Human Genome: A Holistic Conceptual Modeling Approach. *IEEE Access*. DOI: [10.1109/ACCESS.2020.3034793](https://doi.org/10.1109/ACCESS.2020.3034793)

[20] Palacio AL, López ÓP, Ródenas JCC. A Method to Identify Relevant Genome Data: Conceptual Modeling for the Medicine of Precision. En: Trujillo JC, Davis KC, Du X, Li Z, Ling TW, Li G, et al., editores. *Conceptual Modeling*. Cham: Springer International Publishing; 2018. p. 597-609. Disponible en: https://link.springer.com/chapter/10.1007/978-3-030-00847-5_44

[21] «Cóntigo | NHGRI», *Genome.gov*. (accedido 26 de mayo de 2022). Disponible en: <https://www.genome.gov/es/genetics-glossary/C%C3%B3ntigo>

[22] «JavaScript Asíncrono | jonmircha». (accedido 26 de mayo de 2022). Disponible en: <https://jonmircha.com/javascript-asincrono>

[23] «React – Una biblioteca de JavaScript para construir interfaces de usuario». (accedido 26 de mayo de 2022). Disponible en: <https://styde.net/que-es-el-virtual-dom-en-react/>

[24] «Qué es el Virtual DOM en React», (accedido 26 de mayo de 2022). Disponible en: <https://styde.net/que-es-el-virtual-dom-en-react/>

[25] «Tipos de componentes en React Js», *DEV Community*. (accedido 26 de mayo de 2022). Disponible en: <https://dev.to/ajpalacios/tipos-de-componentes-en-react-js-4epg>

[26] «Introduction to GraphQL | GraphQL». (accedido 26 de mayo de 2022). Disponible en: <https://graphql.org/learn>

[27] «MySQL :: MySQL 8.0 Reference Manual :: 1.2.1 What is MySQL?». (accedido 26 de mayo de 2022). Disponible en: <https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>

[28] mstoimenova, «Agregar una regla SNAT o DNAT». Director/10.0/com.vmware.vcloud.tenantportal.doc/GUID-1DE460AE-DCCC-4BC8-96AC-52D06A4AFDE3.html (accedido 26 de mayo de 2022). Disponible en: <https://docs.vmware.com/es/VMware-Cloud->

[29] «Estrategias para la elaboración de genogramas por residentes de medicina familiar», p. 7, 2004. Disponible en: <https://www.medigraphic.com/pdfs/medfam/amf-2004/amf043f.pdf>

[30] M. Á. Suarez Cuba, «EL GENOGRAMA: HERRAMIENTA PARA EL ESTUDIO Y ABORDAJE DE LA FAMILIA», *Rev. Médica Paz*, vol. 16, n.º 1, pp. 53-57, 2010.

[31] «Get Started with GoJS». (accedido 26 de mayo de 2022). Disponible en: <https://gojs.net/latest/learn/>

[32] «Family Tree JavaScript Component | BALKAN FamilyTreeJS», *balkan.app/familytreejs*. (accedido 26 de mayo de 2022). Disponible en: <https://balkan.app/FamilyTreeJS>

[33] *GA4GH Pedigree Standard*. GA4GH Pedigree Standard, 2021. Accedido: 31 de mayo de 2022. [En línea]. Disponible en: <https://github.com/GA4GH-Pedigree-Standard/pedigree/blob/3c8c30231f546a9cb31d1d8f31760dd817300486/docs/source/classes.rst>

Anexo

OBJETIVOS DE DESARROLLO SOSTENIBLE

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

Objetivos de Desarrollo Sostenibles	Alto	Medio	Bajo	No Procede
ODS 1. Fin de la pobreza.				X
ODS 2. Hambre cero.				X
ODS 3. Salud y bienestar.	X			
ODS 4. Educación de calidad.				X
ODS 5. Igualdad de género.				X
ODS 6. Agua limpia y saneamiento.				X
ODS 7. Energía asequible y no contaminante.				X
ODS 8. Trabajo decente y crecimiento económico.				X
ODS 9. Industria, innovación e infraestructuras.		X		
ODS 10. Reducción de las desigualdades.				X
ODS 11. Ciudades y comunidades sostenibles.				X
ODS 12. Producción y consumo responsables.				X
ODS 13. Acción por el clima.				X
ODS 14. Vida submarina.				X
ODS 15. Vida de ecosistemas terrestres.				X
ODS 16. Paz, justicia e instituciones sólidas.				X
ODS 17. Alianzas para lograr objetivos.	X			

Reflexión sobre la relación del TFG/TFM con los ODS y con el/los ODS más relacionados.

Esta tesis se relaciona en gran medida con los ODS 3, Salud y bienestar y 17, Alianzas para lograr objetivos y además, aunque en menor medida que los anteriores, guarda relación con el ODS 9, Industria, innovación e infraestructuras.

Comenzando por el ODS 3, este TFG se alinea a la perfección con dicho objetivo de desarrollo sostenible, debido a que busca la mejora de la salud y el bienestar. Todo este proyecto científico va potencialmente a mejorar la salud y la calidad de vida de las personas. Es por ello por lo que mediante la creación y visualización de genogramas se busca profundizar y estudiar la enfermedad desde el punto de vista genético, ayudando así a los profesionales de la salud a realizar un diagnóstico más específico para cada paciente dependiendo de las características familiares y del estudio que se puede realizar a partir de la plataforma web desarrollada. Además, con este trabajo se pretende poder diagnosticar de forma precoz enfermedades o indicios a sufrir la enfermedad estudiada a los familiares de los pacientes gracias también al estudio genético previo. De esta forma se conseguiría no solo mejorar la salud de quienes sufren la enfermedad, sino también mejorar la calidad de vida permitiendo aplicar medicamentos y tratamientos más específicos según el resultado del estudio que este proyecto permite realizar.

En cuanto al ODS 17, se puede destacar que este trabajo se ha desarrollado en el marco de un proyecto científico en el que colaboran científicos líderes en el campo de la inteligencia artificial de la Universidad Politécnica de Valencia (UPV), servicios médicos de los hospitales más importantes de la Comunidad Valenciana (La Fe, Clínico y Alicante) y una empresa líder en su sector (Bionos Biotecnología). Es por esto por lo que a lo largo del desarrollo de esta tesis se ha fomentado la interoperabilidad y se ha trabajado en equipo teniendo reuniones periódicas para lograr alcanzar el objetivo común, el desarrollo de un sistema inteligente que se despliegue en los hospitales para alcanzar el objetivo de la mejora de la salud en la sociedad.

Finalmente, aunque en menor medida se puede considerar que este trabajo guarda relación con el ODS 9, industria, innovación e infraestructura. Esta tesis fomenta la innovación en el ámbito de la salud impulsando la colaboración entre dos industrias: la sanitaria y la tecnológica. Esta colaboración se realiza bajo criterios de sostenibilidad que adopta tecnologías y procesos industriales limpios y ambientalmente racionales, impulsando la tecnología, la innovación y la investigación y logrando el acceso igualitario a la información y al conocimiento, principalmente a través de internet ya que se está incluyendo un sistema de información en los hospitales que sirven como apoyo para el diagnóstico médico. En cuanto a la investigación, es cierto que este trabajo puede ser un punto de partida para nuevas investigaciones en relación con aspectos hereditarios o emocionales que se puedan visualizar a partir del genograma relacionándolo con las enfermedades tratadas.