CAMPUS
DE EXCELENCIA
INTERNACIONAL

Graduado en Ingeniería Informática

Universidad Politécnica de Madrid

Escuela Técnica Superior de
Ingenieros Informáticos

TRABAJO FIN DE GRADO

Creación de una aplicación web para la edición de datos genéticos

Autor: Sanad El seid Rabah

Directores: Raul Alonso Calvo

Sergio Paraíso

MADRID, JULIO 2017

Índice

Resumen

El trabajo fin de grado que se explicara a continuación tiene como objetivo realizar una aplicación web donde se representa el historial médico de un paciente mediante la representación de su pedigrí humano.

En primer lugar, uno de los objetivos más importantes de esta aplicación es realizar la representación de forma estandarizada y siguiendo un estándar internacional y así poder intercambiar dicha información entre los profesionales de la salud que se encuentran en cualquier lugar del mundo.

La aplicación se basará en la nomenclatura "Nomenclatura Estandarizada del Pedigrí Humano" de la 'National Society of Genetic Counselors' de los Estados Unidos de América.

En segundo lugar, en el desarrollo de la aplicación se utilizará una de las librerías más potentes de hoy en día para generar el grafo que representara el pedigrí humano del paciente. Esta librería es Cytoscape.js que esta, y como su nombre indica, escrita en JavaScript.

El proyecto aparcará el desarrollo de las distintas páginas que representan el front-end de la aplicación y donde el usuario podrá ir navegando entre ellas. Dichas paginas tenían una primera versión diseñada por el grupo de Informática Biomédica donde se mejorarán para realizar las funcionalidades necesarias, para la mejoría de dichas páginas se utilizó librerías como Bootstrap para mejorar su diseño.

Finalmente, una vez realizadas todas las funcionalidades necesarias para cumplir con los objetivos de la aplicación web se han realizado una serie de pruebas para asegurar el correcto funcionamiento y el cumplimiento de la serie de requisitos identificados al principio del proyecto.

Summary

The end-of-grade work that will be explained below aims to make a web application where the medical history of a patient is represented by the representation of his human pedigree.

First of all, one of the most important objectives of this application is to perform the representation in a standardized way and following an international standard and thus be able to exchange this information among health professionals who can be anywhere in the world.

The application will be based on the nomenclature "Standardized Nomenclature of the Human Pedigree" of the National Society of Genetic Counselors of the United States of America.

Second, at the development of the application one of the most powerful libraries will be used to generate the graph that represent the patient's human pedigree. This library is Cytoscape.js that is, and as the name implies, written in JavaScript.

The project will have various pages that represent the front-end of the application and where the user will be able to navigate between them. These pages had a first version designed by the group Informática Biomédica where they Will be improved to realize the necessary functionalities, for the improvement of this pages we used libraries like Bootstrap.

Finally, once all the necessary functionalities to comply with the objectives of the web application have been carried out, a series of tests have been carried out to ensure the correct functioning and compliance of the set of requirements identified at the beginning of the project.

Capítulo 1. Introducción

1.1 Definición del problema

Hoy en día se puede ver que la informática se ha introducido en casi todos los sectores debido, entre otras cosas, a las grandes facilidades que otorga para la disponibilidad y la organización de los datos, el campo de la medicina no iba ser de menos ya que era necesario organizar y poder disponer en cualquier momento y lugar de la información clínica de los pacientes.

Dicha información de los pacientes puede estar reflejada principalmente en sus historiales clínicos por lo que hace de la tarea de mantener dichos historiales actualizados y poder intercambiar dicha información una tarea muy importante.

Para poder llevar a cabo dicha tarea es necesario el uso de los sistemas de información actuales ya que con su uso se puede ahorrar mucho tiempo debido a las facilidades que nos ofrecen para acceder a información que se encuentra en destientos sitios del mundo además del gran potencial que tienen a la hora de trabajar con grandes cantidades de datos.

Con toda esta globalización nos surge un problema, y es que las cosas no se representan de forma igual en todos los lugares del mundo por lo que la representación de los datos debería estar estandarizada para que una persona que accede a dichos datos desde otro lugar del mundo los pueda entender.

En el mundo de la medicina este punto es muy importante ya que a la hora de intercambiar la información médica de un paciente es necesario entender dichos datos. Hay que destacar que en el mundo de medicina este tipo de intercambio de información entre los profesionales de la salud se realiza mucho por lo que para el correcto entendimiento de estos datos deberían seguir un estándar internacional, un pedigrí humano.

La aplicación que se desarrollara en el presente trabajo se basará en un estándar internacional para la representación de un pedigrí humano, así como un código de enfermedad que siga una clasificación internacional para el registro de dichas enfermedades.

La representación del pedigrí humano se basará en la ‘Standardized Human Pedigree Nomenclature’, en cuanto al código de enfermedades utilizado será el ICD-10 (International Classification of Diseases), conocida en español como CIE-10 (Clasificación Internacional de Enfermedades).

Llegados a este punto resulta fundamental saber que es el pedigrí humano, el pedigrí humano es el historial clínico de un paciente y su familia representados gráficamente con todas sus relaciones genéticas.

La 'National Society of Genetic Counselors Pedigree Standardization Task Force' (NSGC PSTF) empezó a trabajar junto a las mejores sociedades de profesionales genéticos sobre el tema de desarrollar una nomenclatura estandarizada del pedigrí humano a raíz de un documento que redacto la misma PSTF en los noventa donde venia explicado que incluso los profesionales dedicados a la genética estaban utilizando un conjunto de símbolos y de términos que eran inconsistentes a la hora de registrar un pedigrí.

Esta nomenclatura tiene el objetivo de facilitar la comunicación entre los profesionales de la salud, los pacientes y los familiares respecto al tema de los diagnósticos genéticos y a las pruebas realizadas al paciente, así como ayudar a reducir los errores médicos.

Sin olvidar que el uso de dicha nomenclatura ayuda ahorrar unas cantidades grandes de dinero gracias a la documentación de las pruebas que se realiza previamente y así los test genéticos se pueden empezar más eficientemente.

1.2 Objetivos

Este trabajo fin de grado tiene como objetivo el desarrollo de la parte cliente de la aplicación (front-end), utilizando para dicho desarrollo las funcionalidades anteriormente implementadas para dar suporte a la aplicación (back-end).

Esta aplicación va a dibujar un grafo que representa el pedigrí del paciente utilizando los datos del paciente que van a llegar desde la parte back-end de la aplicación, y le va ofrecer al cliente varias funcionalidades que le van a ayudar a visualizar o modificar los datos del paciente.

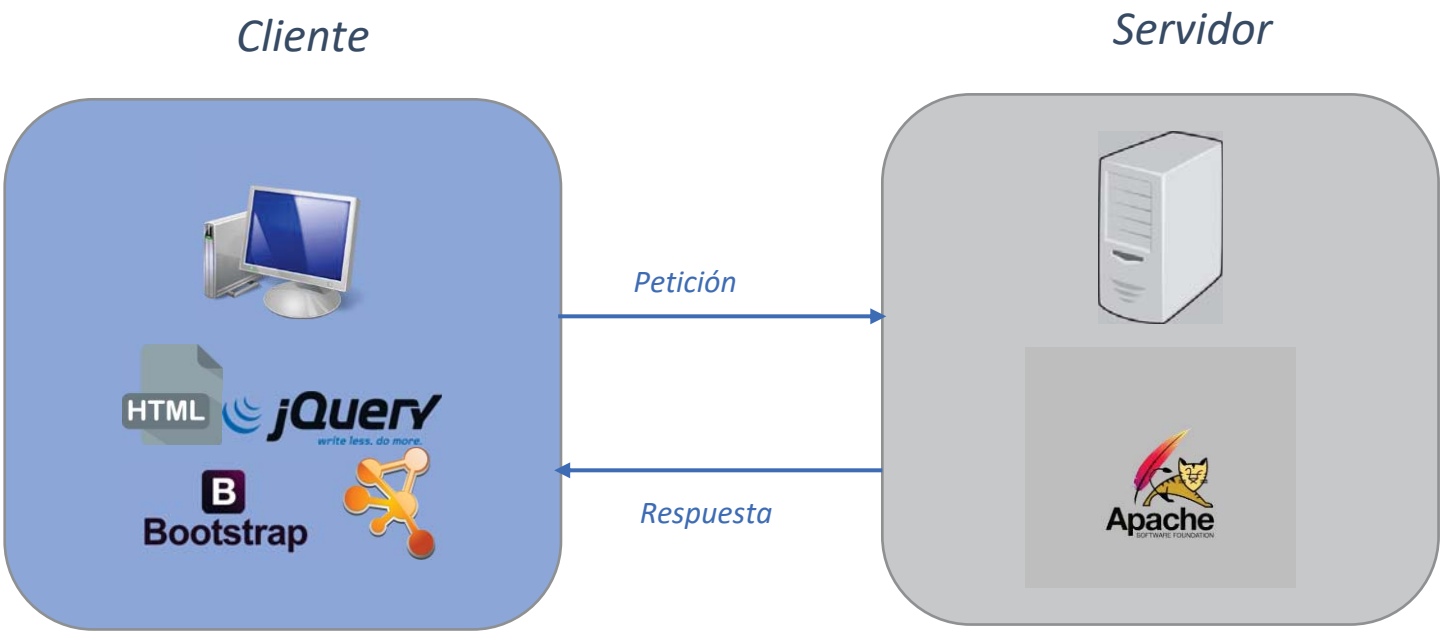
En términos generales las funcionalidades que va tener que ofrecer nuestra aplicación son los siguientes:

- Representar el pedigrí del paciente mediante un grafo utilizando los datos del paciente y sus familiares y su historial médico.
- Almacenar información sobre pacientes y de los familiares.
- Representar la información completa de cada paciente, así como poder exportar dicha información a un archivo exterior en el caso de que se desea esto.
- Poder actualizar la información del paciente en cualquier momento siempre y cuando se tengan los permisos adecuados para realizar dicha modificación.

- Dar la posibilidad de añadir nuevos diagnósticos a los pacientes, para dichos diagnósticos se van utilizar códigos estandarizados reconocidos globalmente para representar cada enfermedad.
- Añadir las relaciones existentes entre los pacientes y poder modificar el estado de las mismas.
- Añadir las gestaciones que surjan de la unión entre dos personas, así como añadir todos los nacimientos que se producen de dicha gestación.
- Poder Representar el pedigrí de cualquier familiar que se encuentra almacenado en el sistema.
- Poder eliminar cualquier paciente y su información del pedigrí y del sistema.

1.3 Solución propuesta

En el presente trabajo fin de grado y con el objetivo de cumplir todos los objetivos listados en el apartado anterior se desarrollará una aplicación web que realice todas las funcionalidades necesarias para cumplir dichos objetivos. Dicha aplicación tiene como diagrama general el mostrado a continuación.



Este diagrama representa la parte del front-end de la aplicación web y es la parte que se va realizar durante este trabajo fin de grado.

Como podemos ver en el diagrama la mayoría de las tecnologías que se van a encargar de realizar las funcionalidades estarán empleadas en la parte del cliente, donde se lanzarán desde allí peticiones HTTP hacia el servidor web.

Para la parte servidor se utilizará Apache Tomcat como servidor local de la aplicación en cuanto a las funcionalidades se implementarán en JavaScript utilizando las funcionalidades que nos ofrecen librerías como jQuery o Cytoscape.js que es la responsable de realizar el grafo, en cuanto a las páginas que forman la aplicación se implementarán en HTML y se les aplicara los estilos de diseño utilizando CSS y la librería Bootstrab para así tener mejor diseño.

Capítulo 2. Estado del arte

2.1 Tecnología de desarrollo

2.1.1 HTML

Es un lenguaje de marcado que se utiliza para el desarrollo de páginas de Internet. Se trata de la sigla que corresponde a HyperText Markup Language, es decir, Lenguaje de Marcas de Hipertexto, que podría ser traducido como Lenguaje de Formato de Documentos para Hipertexto. El estándar HTML lo define la W3C (World Wide Web Consortium) y actualmente HTML se encuentra en su versión HTML5.

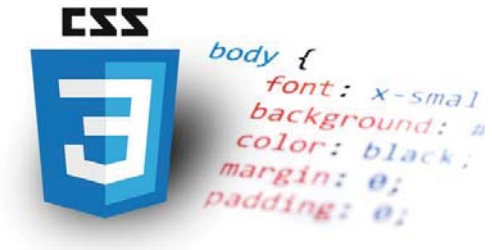
Básicamente el lenguaje HTML sirve para describir la estructura básica de una página y organizar la forma en que se mostrará su contenido, además de que HTML permite incluir enlaces (links) hacia otras páginas o documentos.

Es un lenguaje muy simple y general que sirve para definir otros lenguajes que tienen que ver con el formato de los documentos. El texto en él se crea a partir de etiquetas, también llamadas tags, que permiten interconectar diversos conceptos y formatos.

Para la escritura de este lenguaje, se crean etiquetas que aparecen especificadas a través de corchetes o paréntesis angulares: < y >. Entre sus componentes, los elementos dan forma a la estructura esencial del lenguaje, ya que tienen dos propiedades (el contenido en sí mismo y sus atributos).

Por otra parte, cabe destacar que el HTML permite ciertos códigos que se conocen como scripts, los cuales brindan instrucciones específicas a los navegadores que se encargan de procesar el lenguaje. Entre los scripts que pueden agregarse, los más conocidos y utilizados son JavaScript y PHP.

2.1.2 CSS



Las Hojas de Estilo en Cascada (Cascading Style Sheets), es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos nos permite el control total sobre estilo y formato de los documentos.

CSS se utiliza para dar estilo a documentos HTML, XML y XHTML separando el contenido de la presentación. CSS nos permitirá controlar el estilo y el formato de múltiples páginas Web al mismo tiempo. Cualquier cambio en el estilo marcado para un elemento en la CSS afectará a todas las páginas vinculadas a esa CSS en las que aparezca ese elemento.

Es recomendable separar el archivo CSS del HTML, XML o XHTML, debido a que así nos será mucho más fácil las modificaciones, actualizaciones o la variabilidad de la página según la persona que la esté usando (A este concepto se le llama Usabilidad).

Es posible usar el mismo lenguaje HTML para añadir formato a las aplicaciones web. Sin embargo, CSS ofrece más opciones y es más preciso y sofisticado además está soportado por todos los navegadores de hoy en día.

Entre los beneficios que nos puede ofrecer el uso de CSS encontramos los siguientes puntos:

- control de la presentación de muchos documentos desde una única hoja de estilo.
- control más preciso de la presentación.
- numerosas técnicas avanzadas y sofisticadas.
- aplicación de diferentes presentaciones a diferentes tipos de medios.

2.1.3 Bootstrap

Bootstrap es un framework CSS desarrollado inicialmente (en el año 2011) por Twitter que permite dar forma a un sitio web mediante librerías CSS que incluyen tipografías, botones, cuadros, menús y otros elementos que pueden ser utilizados en cualquier sitio web.

Bootstrap nos ayuda a maquetar un sitio web con rapidez y, sobre todo, ayudándonos a que el diseño sea correcto y usable tanto en dispositivos convencionales con en los táctiles (responsive web design). Para hacerlo, nos ofrece una serie de estilos CSS y librerías JavaScript que nos ayudarán de una manera rápi-da a desarrollar nuestro sitio web y sobre todo es recomendable para el desarrollo de prototipos y tener un tiempo de respuesta realmente bueno.

Actualmente se encuentra con la versión 3 que desde su aparición ha aumentado la compatibilidad con el desarrollo de web responsive ya que se ha añadido un sistema GRID que permite diseñar usando un GRID de 12 columnas donde se plasma el contenido, como que también nos permite insertar imagines responsive (mediante la clase “img-responsive”).

Para usar Boostrap solamente hay que incorporar dos elementos a nuestros documentos HTML: la hoja de estilos base y el fichero base de JavaScript. Una vez tenemos estos elementos enlazados solamente debemos acordarnos de utilizar las clases CSS que define Boostrap.

Además, nos ofrece gran cantidad de elementos como menús desplegables, ayudas para los formularios, etc. dentro de lo que se denomina componentes.

Boostrap es compatible con la mayoría de navegadores web del mercado, y más desde la versión 3, actualmente es totalmente compatible con Google Chrome (en todas las plataformas), Safari (tanto en iOS como en Mac), Mozilla Firefox (en Mac y en Windows), Internet Explorer (en Windows y Windows Phone) y Opera (en Windows y Mac).

2.1.4 JavaScript

JavaScript es un lenguaje de programación creado por Netscape con el objetivo de integrarse en HTML y facilitar la creación de páginas dinámicas, sin necesidad de utilizar scripts de CGI, siendo implementado en todo tipo de navegadores web.

No hay que confundir Java con JavaScript. Java es un lenguaje completo que permite crear aplicaciones independientes, mientras que JavaScript es un lenguaje que funciona como extensión del HTML.

Es un lenguaje de programación orientado a objetos, diseñado para el desarrollo de aplicaciones cliente-servidor a través de Internet. El código de programa de JavaScript, llamado script, se introduce directamente en el documento HTML y no necesita ser compilado, es el propio navegador el que se encarga de traducir dicho código.

Este lenguaje posee varias características, entre ellas podemos mencionar que es un lenguaje basado en acciones que posee menos restricciones. Además, es un lenguaje que utiliza Windows y sistemas X-Windows, gran parte de la programación en este lenguaje está centrada en describir objetos, escribir funciones que respondan a movimientos del mouse, aperturas, utilización de teclas, cargas de páginas entre otros.

Es necesario resaltar que hay dos tipos de JavaScript: por un lado, está el que se ejecuta en el cliente, este es el JavaScript propiamente dicho, aunque técnicamente se denomina Navigator JavaScript. Pero también existe un JavaScript que se ejecuta en el servidor, es más reciente y se denomina LiveWire Javascript.

Tenemos dos formas para usar JavaScript, la primera y como hemos mencionado anteriormente es embebiendo directamente el código JavaScript dentro del código HTML, la segunda es definiéndolo en un archivo JavaScript externo, estos archivos son archivos de texto guardados con la extensión .js.

Dentro de JavaScript tenemos varios frameworks o plataformas que cada una de ellas tiene sus características, pero al fin y al cabo usan el lenguaje de JavaScript, la que se ha implementado en este proyecto y es de la que vamos a hablar a continuación es jQuery.

JQuery

jQuery es una librería JavaScript open-source, que funciona en múltiples navegadores, y que es compatible con CSS3. Su objetivo principal es hacer la programación "scripting" mucho más fácil y rápida del lado del cliente. Con jQuery se pueden producir páginas dinámicas, así como animaciones parecidas a Flash en relativamente corto tiempo.

Existen muchas otras librerías JavaScript como MooTools o AngularJs pero jQuery se ha convertido en la más popular debido a su facilidad de uso y su gran potencia.

Mucha gente confunde jQuery y JavaScript como dos lenguajes de programación distintas, pero no es así ya que ambos son JavaScript. La diferencia es que jQuery ha sido optimizado para realizar muchas funciones de script frecuentes y lo hace a la vez que utiliza menos líneas de código.

Podemos resumir las ventajas de usar jQuery en los siguientes puntos

- jQuery es flexible y rápido para el desarrollo web
- Viene con licencia MIT y es Open Source
- Tiene una excelente comunidad de soporte
- Tiene Plugins
- Bugs son resueltos rápidamente
- Excelente integración con AJAX

2.1.5 Cytoscape

Cytoscape es un programa de código abierto que permite visualizar redes de interacción molecular y pathways biológicos, además integra anotaciones, perfiles de expresión de genes y otros datos.

Cytoscape fue diseñado inicialmente para investigación biológica, hoy en día es una plataforma general para el análisis y visualización de redes. Cytoscape viene con una serie de características básicas para la integración de datos, análisis y visualización. Se puede añadir más funcionalidad mediante plug-ins que permiten incluir nuevos análisis, más tipos de ficheros, scripting y conexión con bases de datos usando una API abierta.

Para el desarrollo de este trabajo se va usar la librería Cytoscape.js que está escrita totalmente en JavaScript y donde va tener la funcionalidad de realizar gráficos interactivos en el navegador web.

Cytoscape.js nos permite visualizar y manipular fácilmente gráficos ricos e interactivos. Debido a que Cytoscape.js permite al usuario interactuar con el gráfico y la biblioteca permite al cliente conectarse a eventos de usuario, Cytoscape.js se integra fácilmente en la aplicación, especialmente porque Cytoscape.js admite tanto los navegadores de escritorio como Chrome y los navegadores móviles, Como en el iPad.

Cytoscape.js se encuentra actualmente en la versión 3.0 que es la que se usó en el desarrollo de este proyecto.

También cabe destacar que al ser un proyecto de código abierto cualquiera puede contribuir libremente en mejorarla y en añadir nuevas funcionalidades a esta librería.

Una de las ventajas que nos ofrece cytoscape.js es la cantidad de plugins que tiene para facilitar el trabajo y hacerlo más bonito y más moderno y esto para debido a que es un código abierto y como hemos apuntado anteriormente cualquiera pueda añadir nuevas funcionalidades o plugins para que los demás usuarios las usen. Una de estas plugins es la cytoscape-cxtmenu.js que nos permite crear un menú circular en el nodo de Cytoscape.js. Este menú mantiene opciones o funcionalidades que se ejecutan sobre este mismo nodo de Cytoscape.js.

2.1.6 Apache Tomcat

Tomcat es un contenedor de servlets que se utiliza en la Referencia oficial de la implementación para Java Servlet y JavaServer Pages (JSP). Las especificaciones Java Servlet y JavaServer Pages son desarrolladas por Sun Microsystems cuyas especificaciones vienen dadas por la JCP (Java Community Process). Apache Tomcat es desarrollado en un entorno abierto y participatorio, bajo la licencia de Apache Software License.

Para simplificar, podríamos decir que Apache Tomcat (o Jakarta Tomcat) es un software desarrollado con Java (con lo cual puede funcionar en cualquier sistema operativo, con su máquina virtual java correspondiente) que sirve como servidor web con soporte de servlets y JSPs.

Los desarrolladores de aplicaciones web cada vez utilizan más Apache Tomcat por todas las funcionalidades que les proporciona eso añadido a que es muy estable y contiene todas las características de un contenedor de aplicaciones web comercial.

La aplicación de administración de Tomcat y las implementaciones especializadas en el rearmado del servidor son algunas de las características adicionales que ofrece.

Tomcat es mantenido y desarrollado por miembros de la Apache Software Foundation y voluntarios independientes. Los usuarios disponen de libre acceso a su código fuente y a su forma binaria en los términos establecidos en la Apache Software License. Las primeras distribuciones de Tomcat fueron las versiones 3.0.x. Las versiones más recientes son las 9.0.0.

Tanto las versiones de las especificaciones de los Java Servlet como la compatibilidad con la máquina virtual Java dependen de la versión escogida del Tomcat como podemos ver a continuación

- **Tomcat 3.x** (distribución inicial): Implementación a partir de Servlet 2.2, JSP 1.1 y JDK 1.1
- **Tomcat 4.x:** implementado a partir de las especificaciones Servlet 2.3, JSP 1.2 y JDK 1.3
- **Tomcat 5.x:** Implementado a partir de las especificaciones Servlet 2.4, JSP 2.0 y JDK 1.4
- **Tomcat 6.x:** Implementado de Servlet 2.5, JSP 2.1 y JDK 1.5
- **Tomcat 7.x:** Implementado de Servlet 3.0, JSP 2.2 y JDK 1.6

En cuanto a los componentes del Apache Tomcat podemos decir que a partir de la versión 4.x fue lanzado con tres componentes básicos que son

- **Catalina:** Este componente implementa las especificaciones de servlets y JSP
- **Coyote:** Es conector que admite el protocolo HTTP 1.1 para el servidor web y que escucha en un puerto TCP especificado por el servidor y envía la solicitud al motor Tomcat para que éste procese la solicitud y envie una respuesta al cliente.
- **Jasper:** Es un motor para JSP que analiza archivos JSP para compilar el código Java y, si se producen cambios, éste los vuelve a compilar.

2.1.7 Java Server Pages

Java Server Pages más conocido por su acrónimo JSP es una tecnología orientada a crear páginas web con programación en Java. Con JSP podemos crear aplicaciones web que se ejecuten en variados servidores web, de múltiples plataformas, ya que Java es en esencia un lenguaje multiplataforma.

La tecnología de JSP permite a los desarrolladores y a los diseñadores de web desarrollar rápidamente y mantener fácilmente páginas dinámicas, ricas en información como son las que soportan a sistemas de negociación. La tecnología de los JSP separa la interfaz del usuario de la parte lógica del contenido permitiendo a los diseñadores cambiar a su disposición las plantillas de la interfaz sin alterar el contenido dinámico subyacente.

JSP también permite introducir código para la generación dinámica de HTML dentro de una página web. Esta surge por la necesidad de crear aplicaciones dinámicas para web de forma fácil, ya que la mejor parte del resultado de un programa CGI es estático. Se podría pensar entonces en JavaScript, pero este genera HTML dinámicamente en el cliente y no puede acceder a los cursos del servidor.

Capítulo 3. Especificación de Requisitos Software

3.1 Introducción

En este apartado se va a mostrar la Especificación de Requisitos Software (ERS) de la aplicación web que se está desarrollando. Estas especificaciones siguen los puntos explicados en el estándar IEEE Practica Recomendad para Especificaciones de Requisitos Software (IEEE Std. 830-1998).

3.1.1 Propósito

El presente documento tiene como propósito definir y mostrar los requisitos y las especificaciones funcionales que va tener la aplicación web centrándonos sobre todo en la parte del "front-end". Dicha aplicación web va estar dirigida al personal médico para que puedan visualizar de forma gráfica todo el historial médico del paciente por lo que se tiene que seguir unas especificaciones y restricciones para la forma de representar dicho historial médico.

3.1.2 Alcance

Esta especificación de requisitos va dirigida al personal responsable del desarrollo de la aplicación web ya que van a necesitar consultar los requisitos para poder desarrollar las funcionalidades y las posibles mejoras de la aplicación.

También en cierta medida puede servir a los clientes finales para sabes los objetivos da la aplicación.

3.1.3 Ámbito del sistema

El propósito general de este trabajo fin de grado es construir un sistema para ayudar al personal médico en visualizar el historial clínico familiar del paciente y su pedigrí humano, dándole también la oportunidad de poder modificar dicho historial y poder añadir tanto relaciones nuevas entre los pacientes como diagnósticos descubiertos nuevamente y poder eliminar toda información innecesaria.

Este historial clínico va ser representado según el estándar 'Standardized Human Pedigree Nomenclature' de la NSGC (National Society of Genetic Counselors') de los Estados Unidos de América.

En cuanto a los diagnósticos también, como era recomendado, se utilizó para cada enfermedad que pertenezca a dicho diagnóstico un código de enfermedad que siga una nomenclatura internacional. En nuestro caso hemos utilizado la más recomendad y es la 'clasificación internacional de enfermedades' conocida también como ICD.

Lo que se busca también es que este sistema sea reconocido por cualquier personal sanitario en todo el mundo y por eso se intentó usar todas estas estandarizaciones.

3.1.4 Definiciones, Acrónimos y Abreviaturas

En la tabla que vamos a tener a continuación se listara los diferentes acrónimos y abreviaturas con sus significados

Nombre	Descripción
HTML	HyperText Markup Language
ICD	International Classification of Diseases
CIE	Clasificación Internacional de Enfermedades
NSGC	National Society of Genetic Counselors

3.1.5 Referencia

Referencia	Titulo
IEEE	Recommended Practice for Software Requirements Specifications

3.1.6 Visión general Del documento

Este documento se divide en tres secciones principales. La primera sección es la introducción al mismo documento y donde se muestra una visión general de la Especificación de Requisitos Software (ERS).

La siguiente sección es la descripción general del documento donde se van a explicar las funcionalidades que debe realizar el sistema, los factores y restricciones que afectan al sistema explicándolo de forma general donde no se va entrar en todos los detalles.

La tercera y última sección va ser los requisitos específicos donde se muestran todos los detalles que van a necesitar los desarrolladores para poder cumplir con todos los requisitos de una manera satisfactoria y correcta.

3.2 Descripción general

En esta sección del documento se van a explicar todos los factores generales que van afectar al sistema y a sus requisitos, esta explicación va ser de una forma general para dar una visión general de los requisitos y así poder entenderlos de forma global sin entrar en detalle.

3.2.1 Prospectivas del producto

Este sistema es un sistema que pretende poder representar el pedigrí humano y el historial clínico del paciente para poder facilitar la posibilidad de diagnosticar posibles enfermedades. Es un sistema totalmente nuevo donde no toma como referencia cualquier otro ya que no se ha encontrado otro sistema que realiza las funcionalidades que este sistema quiere alcanzar.

3.2.2 Funcionalidad del producto

Nuestro sistema tiene que realizar las siguientes funcionalidades

- **Generar el grafo que representa el pedigrí del paciente**. La función principal de nuestro sistema será generar el grafo del historial clínico del paciente donde se muestra todas las relaciones familiares y los diagnósticos del paciente.

- **Añadir nuevas relaciones entre los pacientes**. Nuestro sistema también da la oportunidad de tanto añadir nuevas relaciones entre los pacientes como modificar las que ya existen sobre el mismo grafo.

- **Añadir las gestaciones que salen de la unión de dos personas**. El sistema tiene que ser capaz de añadir nuevas gestaciones entre dos personas y que salgan representados en el grafo.

- **Añadir los nacimientos producidos por estas gestaciones.** El sistema puede añadir el nuevo paciente producido de esta gestación, así como toda la información que se quiera asociar a este nuevo paciente.

- **Eliminar los elementos no deseados del grafo.** En el mismo grafo que representa nuestro sistema se puede eliminar alguna de las relaciones que ya están representadas o alguno de los familiares del paciente.

- **Exportar la información del paciente.** La aplicación tiene que poder exportar tanto la información de un paciente en concreto como también exportar todo el grafo representado en caso si lo necesita el personal médico.

- **Añadir nuevos diagnósticos del paciente.** El sistema tiene que ser capaz de añadir nuevos diagnósticos de los pacientes sobre el grafo y que se cambien el grafo en el momento depende de estos nuevos diagnósticos añadidos.

3.2.3 Características del usuario

Como ya se ha explicado anteriormente nuestra aplicación web está destinada al personal médico o relacionado con el sector ya que es una aplicación que va representar el historial clínico de los pacientes por lo tanto quien la va usar tiene que tener la formación necesaria para poder entender tanto la nomenclatura estandarizada para la codificación y almacenamiento de un pedigrí humano como el estándar internacional de nombrado de enfermedades. Dado que es una aplicación web el usuario tiene que tener los conceptos básicos para poder manejar un navegador web.

3.2.4 Restricciones

El usuario final necesita tener un navegador web para poder acceder al sistema ya que es una aplicación web, el usuario no va estar limitado al uso de algún navegador web en concreto ya que el sistema funciona en cualquiera y en cualquier sistema operativo. También necesita tener una conexión a internet.

3.2.5 Suposiciones y dependencias

Este sistema sigue una nomenclatura para la representación del historial clínico del paciente y debido a eso es de suponer que la aplicación web sufra algún cambio en el caso de que esta nomenclatura tenga alguna modificación.

También cabe destacar que el personal que está manejando el sistema tiene que conocer que está tratando con información personal de los pacientes por lo que tiene que pedir permiso a los pacientes en el caso de querer publicar dicha información.

Por último, en cuanto a los requisitos de los equipos donde se va ejecutar la aplicación como ya se explicó en el apartado anterior no es necesario que tengan más que el navegador y la conexión a internet para una correcta ejecución.

3.3 Requisitos específicos

En esta sección se especifican los requisitos funcionales que debe satisfacer la aplicación web para poder construir el pedigrí humano de los pacientes y que permitan que al hacer las pruebas poder asegurar que el sistema satisface estos requisitos.

Los Requisitos específicos se van a clasificar en tres tipos diferentes que son

- Requisitos Hardware
- Requisitos Software
- Requisitos funcionales

3.3.1 Requisitos Hardware

Identificación del requisito	RE_01
Nombre	Funcionamiento en distintas plataformas Hardware
Descripción	El sistema tiene que funcionar en todas las plataformas Hardware
Tipo	Requisito Hardware

Identificación del requisito	RE_02
Nombre	Funcionamiento en distintas arquitecturas Hardware
Descripción	El sistema tiene que funcionar en todas las arquitecturas Hardware tanto de 32 bits como de 64 bits
Tipo	Requisito Hardware

3.3.2 Requisitos Software

Identificación del requisito	RE_03
Nombre	Funcionamiento en distintas plataformas Software
Descripción	El sistema tiene que funcionar en todas las plataformas Software
Tipo	Requisito Software

Identificación del requisito	RE_04
Nombre	Instalación dentro del servidor Apache Tomcat
Descripción	El sistema va estar instalado dentro del servidor de Apache Tomcat
Tipo	Requisito Software

Identificación del requisito	RE_05
Nombre	Tipo de Información
Descripción	El sistema utilizara la información en formato JSON Ya guarda la información después de los cambios en este mismo formato
Tipo	Requisito Software

3.3.3 Requisitos funcionales

Identificación del requisito	RE_06
Nombre	Generación del grafo
Descripción	El sistema tiene que generar el grafo que representa el pedigrí humano del paciente usando los datos de entrada
Tipo	Requisito funcional

Identificación del requisito	RE_07
Nombre	Añadir relaciones
Descripción	El sistema tiene que ser capaz de añadir nuevas relaciones en el grafo entre dos personas que no estaban unidas desde antes
Tipo	Requisito funcional

Identificación del requisito	RE_08
Nombre	Añadir nueva persona
Descripción	El sistema tiene que añadir nuevos pacientes que tengan relación con el paciente, aunque no estén en el grafo anteriormente. (añadir hermano, pareja,..)
Tipo	Requisito funcional

Identificación del requisito	RE_09
Nombre	Eliminar paciente
Descripción	El sistema permite eliminar algún paciente del grafo
Tipo	Requisito funcional

Identificación del requisito	RE_10
Nombre	Nueva gestación
Descripción	El sistema permite añadir una nueva gestación a una unión entre dos personas en el grafo
Tipo	Requisito funcional

Identificación del requisito	RE_11
Nombre	Nuevo nacimiento
Descripción	El sistema añade una nueva persona de forma de nacimiento en una gestación en el grafo
Tipo	Requisito funcional

Identificación del requisito	RE_12
Nombre	Añadir diagnostico
Descripción	El sistema permite añadir un nuevo diagnostico a un paciente añadiéndolo el código de la enfermedad diagnosticada
Tipo	Requisito funcional

Identificación del requisito	RE_13
Nombre	Exportar el grafo
Descripción	El sistema tiene la posibilidad de exportar una imagen del grafo entero a un archivo externo
Tipo	Requisito funcional

Identificación del requisito	RE_14
Nombre	Exportar información
Descripción	El sistema permite exportar la información del paciente a un archivo externo
Tipo	Requisito funcional

Identificación del requisito	RE_15
Nombre	Editar la información
Descripción	El sistema permite editar la información personal del paciente
Tipo	Requisito funcional

Identificación del requisito	RE_16
Nombre	Mostrar la información
Descripción	El sistema permite mostrar la información entera de cada uno de los pacientes que estén representados en el grafo
Tipo	Requisito funcional

3.3.4 Atributos del sistema

Para poder medir la calidad del producto hay que fijarse en cuatro puntos principales que miden esta calidad y son

- **Rendimiento:** Es la probabilidad de que el sistema funcione sin problemas ni fallos
- **Seguridad:** Es asegurar la información de los pacientes y los recursos del sistema
- **Disponibilidad:** El sistema tiene que estar disponible de forma continua y así poder utilizarlo en cualquier momento
- **Mantenibilidad:** El sistema tiene que disponer de la documentación necesaria para que el sistema sea actualizado de forma fácil y sencilla
- **Portabilidad:** La aplicación tiene que funcionar en otras plataformas sin problemas ni fallos.

Capítulo 4. Desarrollo

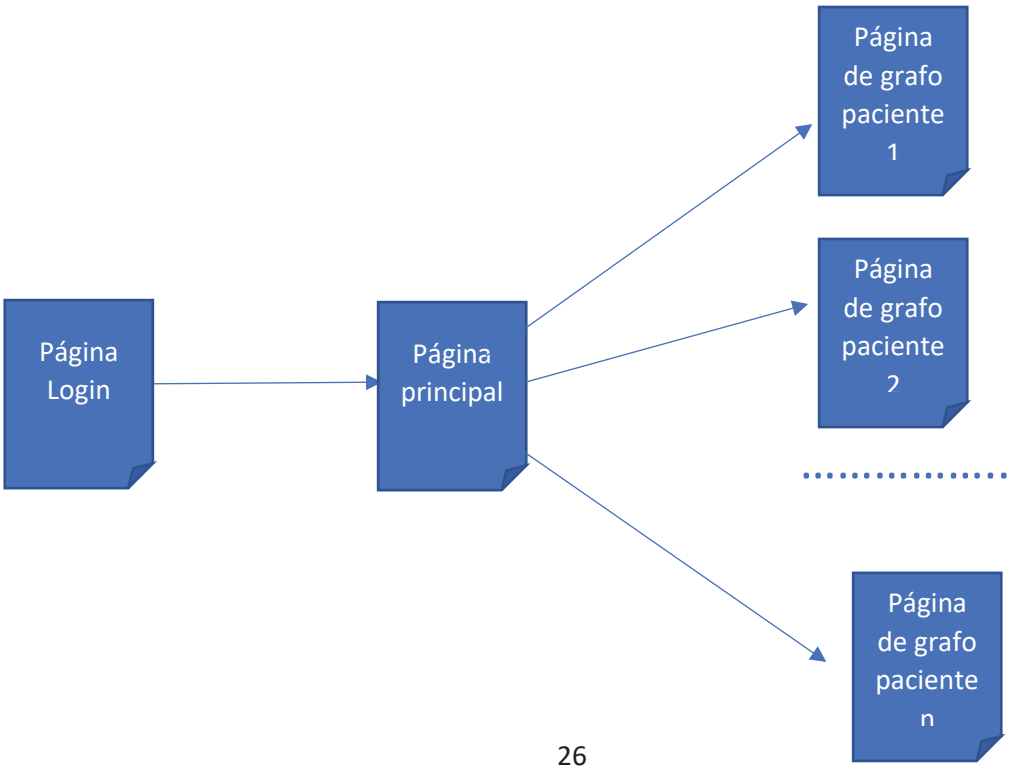
4.1 Introducción

En este capítulo se detallan todos los pasos seguidos para el desarrollo de nuestra aplicación web desde la instalación del servidor pasando por la implementación del diseño de la página web y acabando en la implementación de las funcionalidades requeridas.

4.2 El esquema de la aplicación

En este apartado se muestra un esquema general de la aplicación del flujo de su funcionamiento empezando desde la página de autenticación y llegando hasta la página final donde se muestra el grafo del historial médico del paciente.

En primer lugar, la aplicación dispone de una página de autenticación (Login) donde el usuario tiene que meter su nombre de usuario y su contraseña para poder conectarse a la página principal. Una vez realizado el proceso de autenticación correctamente ingresamos en la página principal donde nos aparecen los distintos modelos que podemos abrir para visualizar el grafo del historial médico del paciente que será allí donde se podrán realizar todas las funcionalidades anteriormente descritas.



4.3 Instalación y configuración del Apache Tomcat

Para la implementación de la aplicación web era necesario para empezar la instalación de un servidor web donde poder desarrollar la aplicación y sus funcionalidades. El servidor elegido era el Apache Tomcat como ya se ha indicado en los apartados anteriores. Este servidor una vez instalado y para su correcto funcionamiento era necesario realizar alguna configuración de las variables de entorno que necesita.

Estas variables de entrono son:

- **CATALINA_HOME.** Es la variable de entrono que tiene que apuntar a la ruta donde se encuentra instalado el servidor Apache Tomcat.

- **JAVA_HOME.** Esta variable es importante para muchas aplicaciones java y tiene que apuntar a donde se encuentra instalado el Java Development Kit (jdk).

Una vez configuradas estas variables del entorno el servidor ya está listo para arrancar y esta tarea se hace fácilmente con el script de arranque que nos ofrece y que solo hace falta ejecutarlo para que arranque el servidor. Este script es el "startup.bat" en caso de Windows o en Linux se usaría "startup.sh", estos scripts se encuentran en la carpeta "bin" dentro del directorio donde se encuentra instalado el servidor.

La imagen 1 se muestra lo que se obtiene si el servidor arranco correctamente.

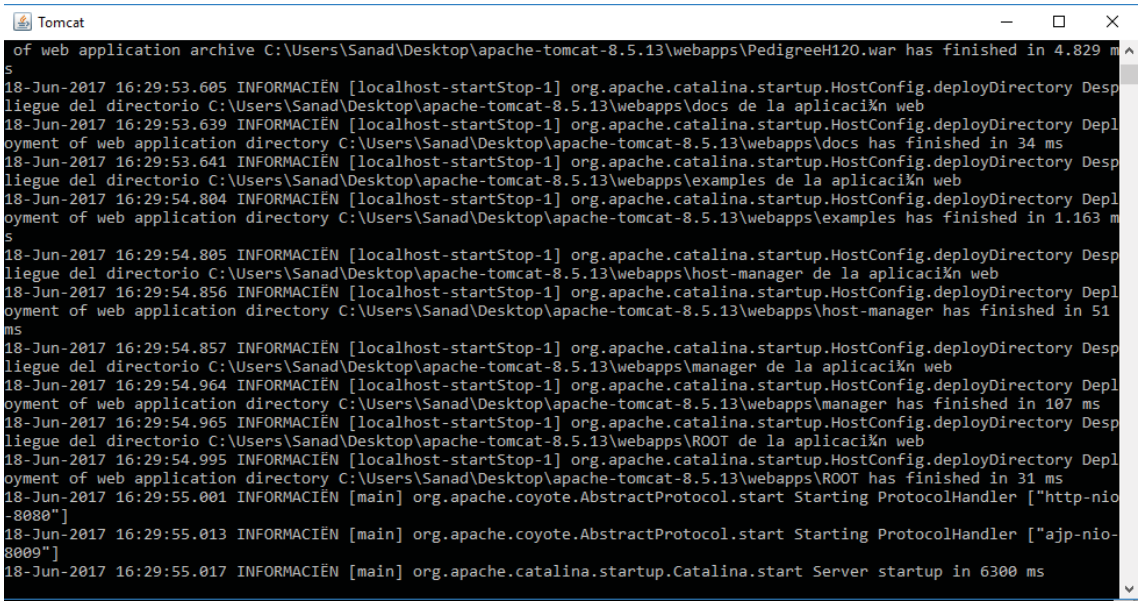


Imagen 1. Arranque del servidor

4.4 Representación de los elementos del grafo

Para la representación del grafo que va formar el pedigrí humano de los pacientes se utilizó la 'Nomenclatura Estandarizada del Pedigrí Humano' de la 'National Society of Genetic Counselors'. En este apartado se van a mostrar las formas que representan cada elemento del grafo según esta nomenclatura.

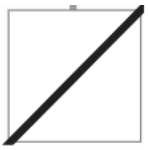
Los pacientes se representan en forma de nodos de la forma que los nodos representados en forma de cuadrado son pacientes masculinos y los que tienen forma de circulo son femeninos y en caso de que todavía no se le ha definido un género al paciente se representa en forma de rumbo.

Paciente masculino

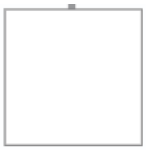
Paciente femenino

Paciente con genero indefinido todavía

En cuanto al estado del paciente si está vivo o muerto se representan de la siguiente manera.



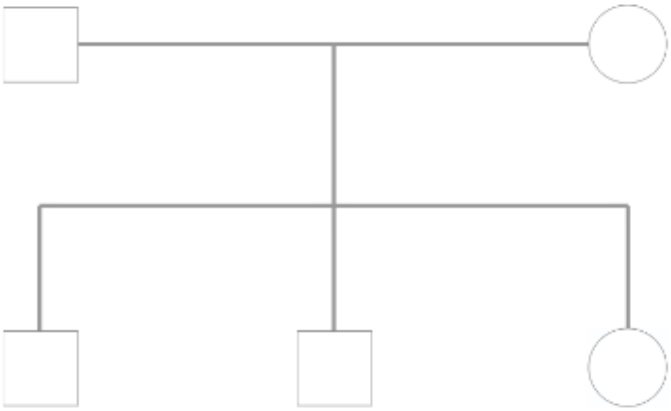
Paciente muerto



Paciente vivo

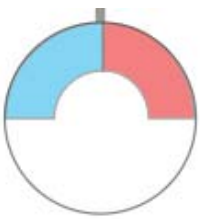
Las relaciones de matrimonio o de pareja que van a unir a dos personas se van a representar con una línea que va conectar los dos pacientes.

En cuanto a los hijos se van a representar en un nivel inferior a los padres y van a estar conectados con ellos. En la siguiente imagen tenemos un ejemplo de cómo se representará un matrimonio y tres hijos producidos a partir de este matrimonio.



En esta misma imagen podemos observar se representan las relaciones de hermano/hermana en nuestro grafo. Tienen que estar en el mismo nivel y conectado por la parte superior entre ellos como se puede ver en la imagen.

Adicionalmente en nuestra aplicación para los diagnósticos añadidos a los pacientes a cada nodo se le añade un color en una esquina para que así a simple vista el personal médico sepa que este paciente (nodo) tiene tal enfermedad. Este color lo puede elegir el personal médico en el momento de añadir este diagnóstico a la información del paciente. Un ejemplo de un nodo con diagnósticos añadidos es el que se muestra a continuación.



4.5 Diseño de las páginas de la aplicación

En este apartado se va a mostrar el diseño y la implementación de las distintas páginas que tiene la aplicación sin entrar en detalle en las funcionalidades que vendrán explicadas en el siguiente apartado.

4.5.1 Login

La primera página que encontramos es la página de la autenticación del usuario o dicho de otra manera "Login" donde el usuario tiene que meter su nombre de usuario y la contraseña para poder acceder a la página principal de la aplicación.

El diseño de esta página es un diseño básico para una página de Login que contiene dos cuadrados de texto para meter el nombre del usuario y la contraseña y un botón para realizar la acción de autenticación como podemos observar en la imagen 2.

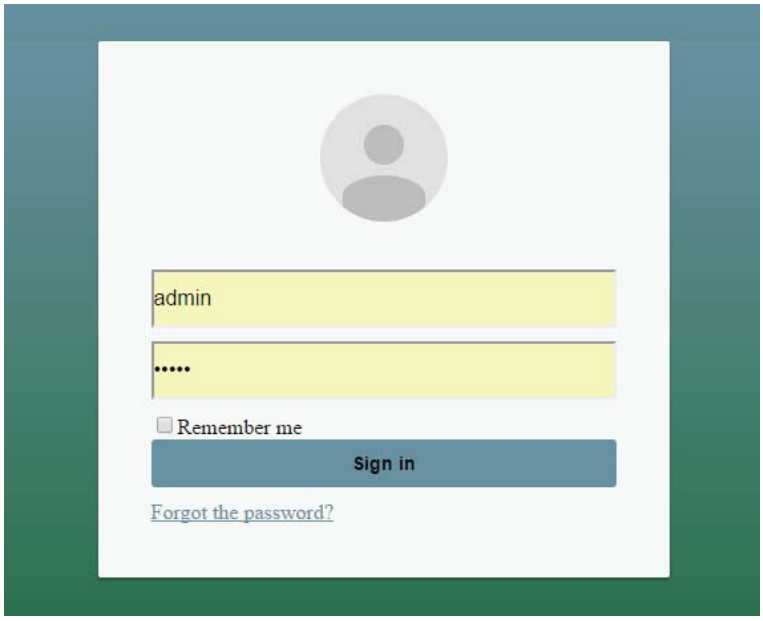


Imagen 2. Página de Login

Esta página se implementó en un archivo JSP (Java Server Pages). Las páginas JSP están compuestas de código HTML mezclado con etiquetas especiales para programar scripts de servidor en sintaxis Java.

En cuanto al código HTML implementado dentro de esta es un código bastante simple que empieza cargando el archivo de las probidades CSS que se van a utilizar dentro de esta página y acto seguido vienen las etiquetas básicas que forman esta página donde se crean los cuadrados de texto el botón y su tipo los colores y todo el diseño que se muestra en la foto anterior.

4.5.2 Página principal

La página principal (index) de la aplicación web es la primera página con la que se encuentra el personal clínico después de autenticarse. Es una página básica donde el personal clínico pude elegir entre los distintos modelos o templates disponibles para representar el historial clínico del paciente. Además de poder elegir el paciente al que se quiere visualizar su historial clínico.

En cuanto al diseño de esta página es un diseño sencillo y fácil de entender y ver ya que el propósito principal de esta página no es realizar funcionalidades si no redirigir el usuario a las siguientes páginas que son las responsables de realizar el grafo que representa el historial clínico del paciente elegido.

En la imagen 3 se muestra el diseño de esta pagina

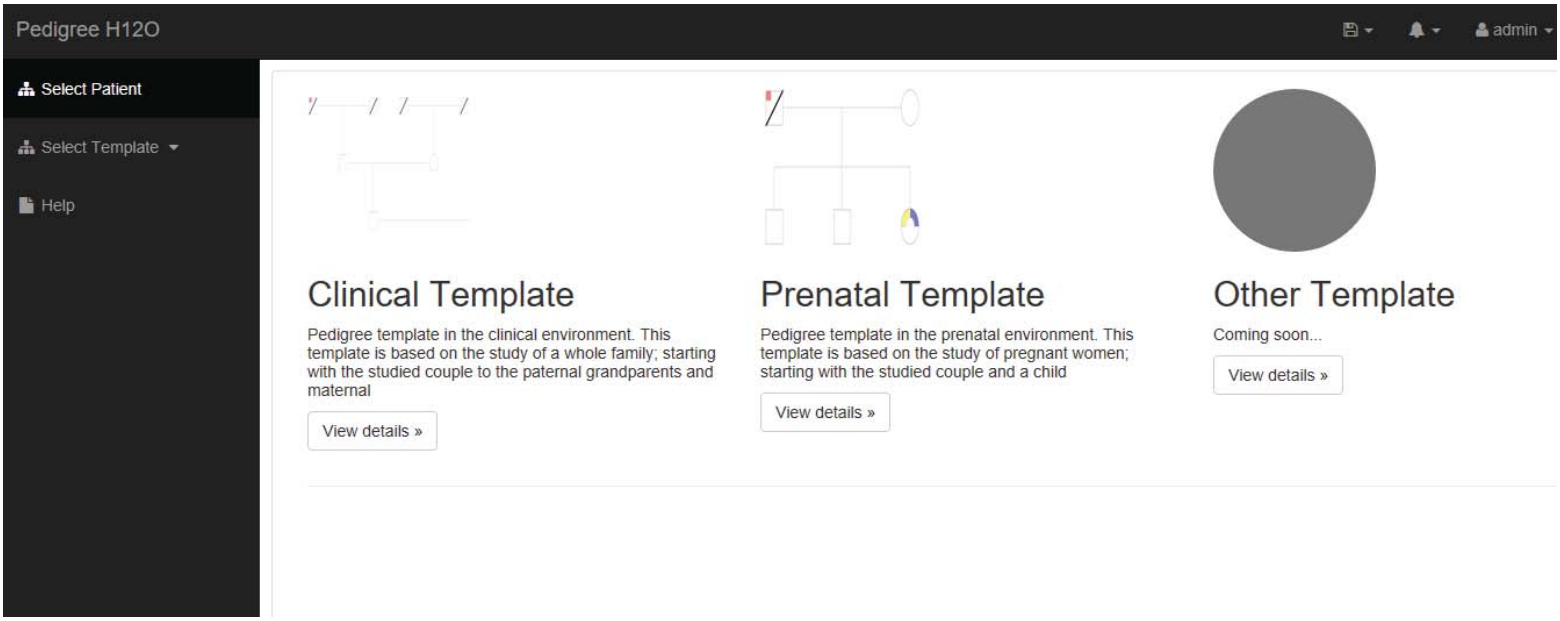


Imagen 3. Página principal

El código que implementa esta página e igual que en el caso de la página de Login está escrito en un archivo JSP donde se pueden mezclar el código HTML que realiza el diseño de esta página con etiquetas especiales para programar scripts de servidor en sintaxis java.

Lo primero que tiene el código de esta página y antes de empezar con el cuerpo de la página se cargan tanto los archivos CSS como los de la librería de Bootstrap que van a ser los responsables a darle las características del diseño como los colores, el tipo de botones o el tipo de menús que se van a desplegar. Estos archivos se cargan dentro de un código HTML usando la etiqueta <*link*> y haciendo referencia de donde se encuentra dicho archivo entre los directorios de la aplicación.

Algunos ejemplos de uso de esta etiqueta son

```
<link href="css/sb-admin.css" rel="stylesheet">

<link href="css/bootstrap.min.css" rel="stylesheet">
```

Una vez terminada esta tarea de cargar los archivos y las librerías necesarias se empieza con el desarrollo y la implementación del cuerpo de la página utilizando el código HTML y sus etiquetas y cargando en cada momento la clase necesaria para el tipo de diseño deseado para cada parte de la página. Estas clases son los que vienen definidos en los archivos anteriormente cargados.

Un ejemplo de la forma que se cargan estas clases en una etiqueta HTML es

```
<nav class="navbar navbar-inverse navbar-fixed-top" role="navigation">
```

El usuario en esta página además de las funcionalidades de redirigirle a las paginas donde se representa el grafo del historial clínico del paciente también puede abandonar la aplicación web mediante un Log out que le permite cerrar la sesión y salir de la aplicación.

Este Log out se encuentra en la parte superior de la página pinchando en el nombre de usuario como podemos observar en la imagen 4.

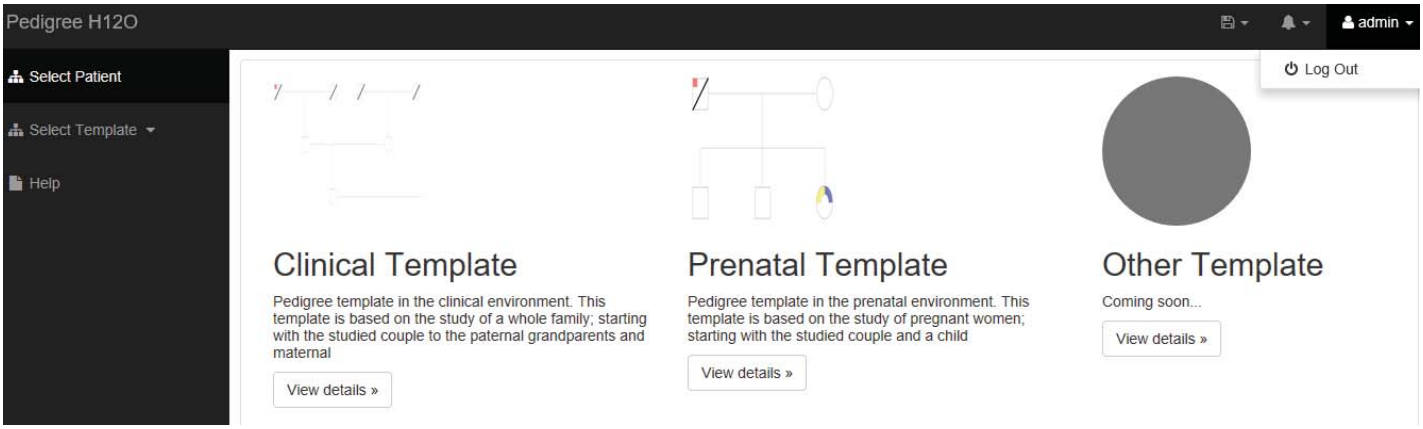


Imagen 4. Página principal con Log out

4.5.3 Página de representación del pedigrí

Esta página es donde se representa el grafo del historial clínico del paciente y es donde se van a realizar la mayoría de las funcionalidades de la aplicación ya que dichas funcionalidades se realizan sobre el mismo grafo. Una ver elegido el paciente en la página principal esto nos lleva a esta página donde se va representar el grafo de este paciente.

En cuanto al diseño mantiene el mismo estilo que tiene la página principal con la diferencia de que va tener el grafo en el centro de la página como podemos observar en la imagen 5.

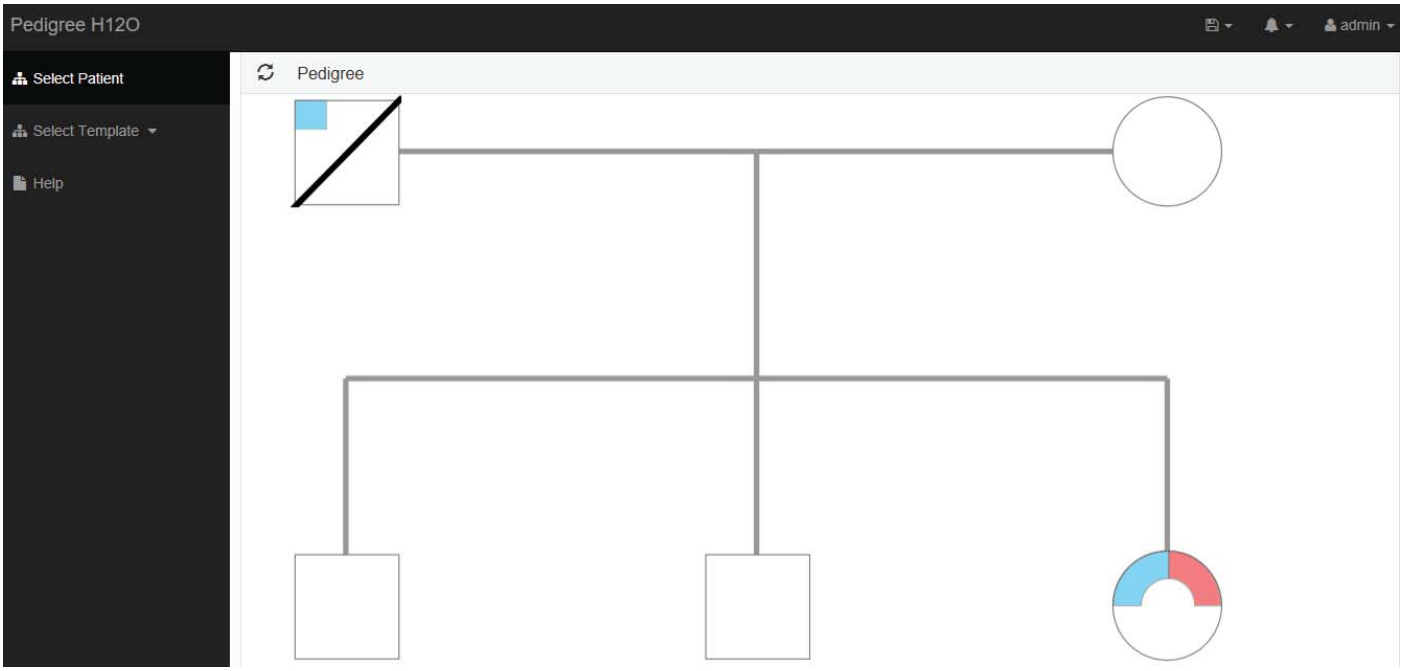


Imagen 5. Página de representación del pedigrí

En este caso este grafo es una representación de un matrimonio con tres hijos que están representados según la nomenclatura que se explicó anteriormente con sus diagnósticos.

Las funcionalidades que se pueden realizar en esta página se van a explicar en el siguiente apartado ya que en este aparato se va mostrar solo el diseño y como se realizó dicho diseño.

Al igual que las dos páginas anteriores la página de la representación fue desarrollada en un archivo JSP donde se tiene tanto el código HTML como algún script de programación que fue necesario para su desarrollo.

A diferencia de las otras dos páginas en esta página además de cargar los archivos CSS y los de la librería de Bootstrap que son los responsables del diseño había que cargar las librerías de jQuery y de Cytoscape.js así como todos los plugins de estas dos librerías que van a ser necesarias para realizar el grafo y realizar todas las funcionalidades que tiene la aplicación.

Estas librerías se van a cargar utilizando la etiqueta *<script>* ya que los archivos que componen estas librerías son archivos escritos en JavaScript. Poniendo dentro de la etiqueta la ruta donde se encuentran estos archivos en los directorios de la aplicación.

Algunos de los ejemplos del uso de esta etiqueta son

```
<script src="js/jquery.auto-complete.js"></script>

<script type="text/javascript" src="js/cytoscape.js"></script>

<script type="text/javascript"  src="js/cytoscape.min.js"></script>
```

Una vez cargados todos los archivos y todas las librerías necesarias se empieza a desarrollar el cuerpo de la página. Dentro de este cuerpo se encuentran desarrollados algunos modelos que se van a utilizar dentro de las funcionalidades y no se ven así en el grafo sin llamar a alguna de estas funcionalidades.

Estos modales son los que vienen a continuación

- **Modal de creación de un nodo:** consiste en una tabla donde se meten los datos del nuevo nodo añadido. Estos datos son el nombre, la fecha de nacimiento, el sexo, el estado y si se quiere añadirle alguna diagnosticación.

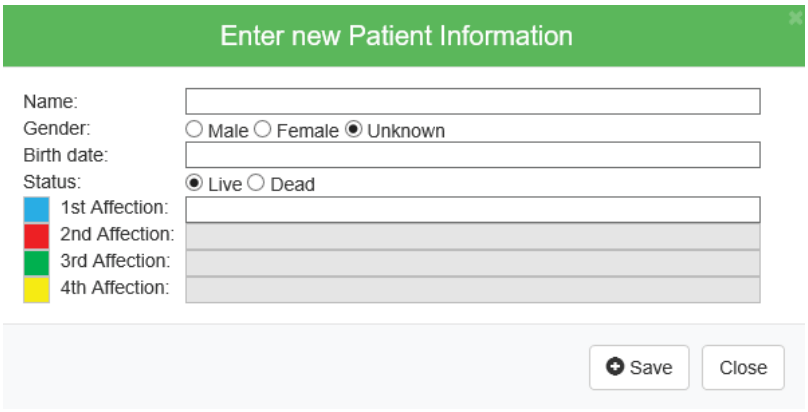


Imagen 6. Modal de creación de un nodo

- **Modal de modificación de un nodo:** Este modal se usa cuando se llama a la función de editar la información de algún nodo. Consiste en una tabla con la información que ya tiene el nodo para poder cambiarla

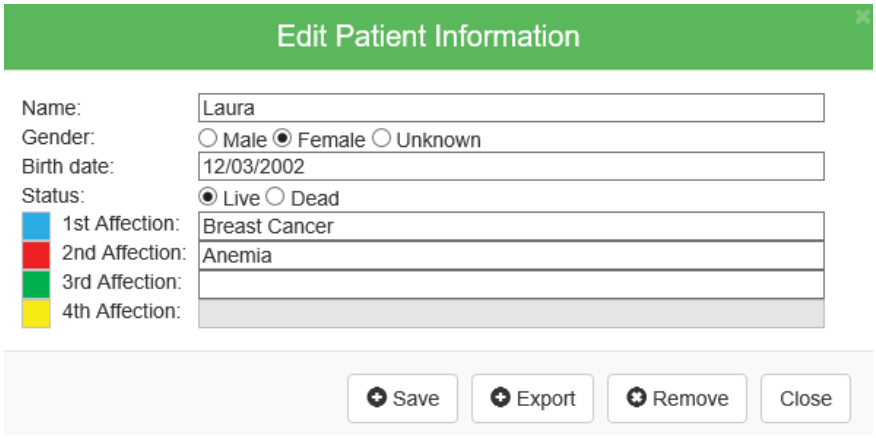


Imagen 7. Modal de modificación de un nodo

- **Modal de mostración de la información de un nodo:** Este modal es el que se muestra al llamar a la función de mostrar la información de un paciente. Consiste en una tabla con toda la información del nodo y con botones que realizan distintas funcionalidades sobre este nodo.

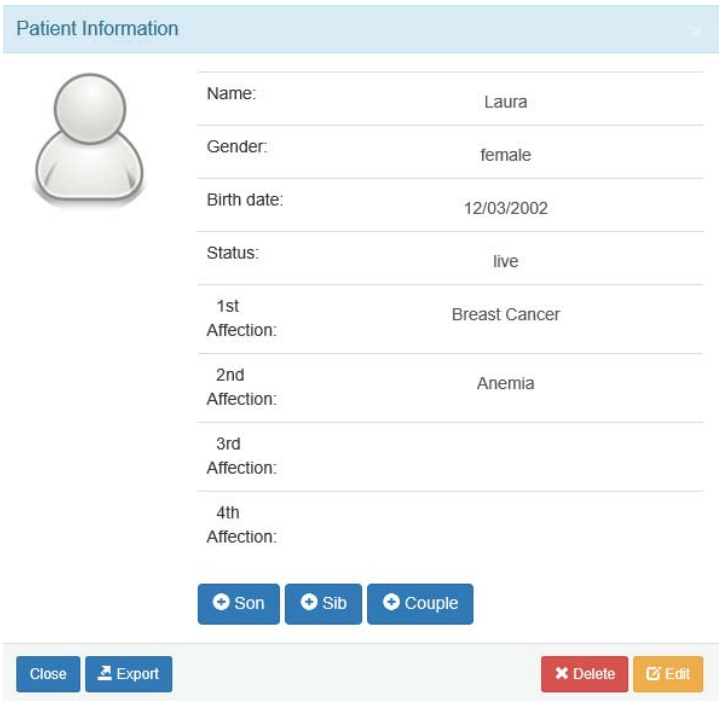


Imagen 8. Modal de mostración de la información de un nodo

4.6 Desarrollo de las funcionalidades

En este apartado se explica y se muestra la implementación de las funcionalidades que realiza la aplicación web y la forma y las tecnologías que se utilizaron para llevar a cabo esta implementación de forma que se satisfacen los requisitos y los objetivos de la aplicación web.

Todas estas funcionalidades han sido desarrolladas utilizando el lenguaje de programación de JavaScript haciendo uso de las muchas tecnologías y librerías que están escritas en este idioma como jQuery y Cytoscape.js y todas las librerías y plugins de estas dos que ayudan a llevar a cabo diversas funciones que van a ser necesarias para hacer las funcionalidades principales.

4.6.1 Datos de entrada

La realización de esta aplicación web se divide, como es lo normal en las aplicaciones web, en dos partes la parte del back-end donde se define el modelo de los datos que va tener y la forma de cómo van a estar clasificados estos datos y la parte de front-end que es la parte que ve el cliente y que utiliza estos datos proporcionados de la parte del back-end para realizar las funcionalidades. Este trabajo se centra en la parte del front-end asique para llevarlo a cabo se han utilizado archivos que contienen datos que están representados de la misma forma que va a proporcionar el back-end estos datos para que así a la hora de integrar los dos partes juntos no haya muchos problemas de integración.

Los datos están escritos en archivos de JavaScript donde se encuentran representados en objetos JSON. JSON es un formato para el intercambio de datos donde se describen los datos con una sintaxis dedicada que se usa para identificar y gestionar los datos.

Básicamente los datos se encuentran en dos objetos JSON uno que tiene toda la información del nodo y otro objeto que tiene las aristas que unen estos nodos, está clasificado de esta forma ya que la librería Cytoscape.js, que es la responsable de realizar el grafo, necesita que los datos de entrada para realizar el grafo estén clasificados de esta forma.

En el siguiente apartado se mostrará como la librería Cytoscape.js llama a estos archivos donde se encuentran los datos para usar los datos que contienen para realizar le grafo.

Los dos objetos JSON representan los datos de la siguiente forma

- **Objeto de las aristas:** tiene tres campos que son el identificador de la arista (id), el origen de donde va salir la arista (source) es el id del nodo origen y por ultima el destino de la arista (target) es el id del nodo destino. A continuación, en la imagen 9 se muestra un ejemplo de este objeto.

```
var demoEdges = [
{data: {id: 'a1',source: '1_1',target:'2_1'}},
{data: {id: 'a2',source: '2_1',target:'3_1'}},
{data: {id: 'a3',source: '2_1',target:'2_2'}},
{data: {id: 'a4',source: '2_2',target:'1_2'}},
{data: {id: 'a5',source: '2_2',target:'3_2'}},
{data: {id: 'a6',source: '1_2',target:'1_3'}},
{data: {id: 'a7',source: '2_2',target:'2_3'}},
{data: {id: 'a8',source: '3_2',target:'3_3'}}
];
```

Imagen 9. Ejemplo de objeto de datos de las aristas

- **Objeto de los nodos:** tiene todos los campos necesarios para la información del nodo como el identificador del nodo, nombre, fecha de nacimiento, sexo y los diagnósticos que tiene el nodo además tiene dos campos que representan la posición del nodo que son la columna (col) y la fina (row) ya que la librería Cytoscape.js utiliza estos dos campos para posicionar el nodo dentro del grafo. A continuación, en la imagen 10 se muestra un ejemplo de este objeto.

```
var demoNodes = [
    {data: {id:'1_1',col: 1,row: 1, weight:'20',
        name:'Pedro H.', birthDate:'19/07/1965',gender:'male', status:'dead',
        diseases:'1', content:'',
        affection1:'Breast Cancer', affection2:'', affection3:'', affection4:'',
        image:'http://localhost:8080/PedigreeH12O/svgs/draw.jsp?gender=male&status=dead&diseases=1&color1=00b0f0'},
    },
    {data: {id:'2_1',name:'',col: 2,row: 1,weight:'10',
        image:'svgs/pixel.svg'},
    },
    {data: {id:'3_1',col: 3,row: 1, weight:'20',
        name:'Cristina J.', birthDate:'03/02/1970', gender:'female', status:'live',
        diseases:'0', content:'',
        affection1:'', affection2:'', affection3:'', affection4:'',
        image:'http://localhost:8080/PedigreeH12O/svgs/draw.jsp?gender=female&diseases=0&status=live'},
    },

    {data: {id:'2_2',col: 2,row: 2, name:'',weight:'10',
        image:'svgs/pixel.svg'},
    },
    {data: {id:'1_2',col: 1,row: 2, name:'',weight:'10',
        image:'svgs/pixel.svg'},
    },
    {data: {id:'3_2',col: 3,row: 2, name:'',weight:'10',
        image:'svgs/pixel.svg'},
    },

    {data: {id:'1_3',col: 1,row: 3, weight:'20',
        name:'Luis', birthDate:'20/05/1997', gender:'male', status:'live',
        diseases:'0', content:'',
        affection1:'', affection2:'', affection3:'', affection4:'',
        image:'http://localhost:8080/PedigreeH12O/svgs/draw.jsp?gender=male&diseases=0&status=live'},
    },
```

Imagen 10. Ejemplo de objeto de datos de los nodos

4.6.2 Implementación del grafo

En este apartado se va a explicar cómo genera la librería Cytoscape.js el grafo utilizando los datos de entrada, así como mostrar todas las posibilidades que nos ofrece esta librería para el momento de generar el grafo como el estilo del grafo o la animación del momento de generarlo.

Ante de empezar hay que apuntar que Cytoscape.js, como su nombre indica, es una librería escrita totalmente en código JavaScript y su uso es compatible con la librería de jQuery que se va a utilizar también en el desarrollo de la aplicación.

Para empezar con el trabajo en Cytoscape.js se crea un objeto cytoscape que tiene destientos atributos dentro que describen los detalles del grafo que se quiere generar. Estos atributos son

- **Container:** es donde se declara la zona en la página web donde se va dibujar el grafo. Apunta hacia el identificador de la etiqueta HTML que define la zona de la página donde se va a desplegar el grafo. Un ejemplo usando jQuery puede ser el siguiente donde “cy “es el identificador de la etiqueta
  container: document.getElementById('cy')

- **Style:** en este atributo se definen las propiedades de diseño CSS de los elementos del grafo. Estos elementos son los nodos y las aristas. Estas propiedades CSS son como el tamaño del nodo, tamaño de la arista o el tipo de fondo. En la imagen 11 se muestra un ejemplo del uso de este atributo

```
style: [
                {
                    selector: 'node',
                    css: {
                        'width':'data(weight)',
                    'height':'data(weight)',
                    'background-fit': 'cover',
                    'text-valign': 'bottom',
                    'font-size': 9,
                    'shape':'rectangle',
                    'opacity':0.01,
                    'background-image': 'data(image)',
                        }
                    },
                    {
                        selector: 'edge',
                        css: {
                            'width':1

                        }
                    }
            ],
```

Imagen 11. Ejemplo de uso del atributo style

- **Elements:** en este atributo es donde se cargan los datos de entrada de los nodos y de las aristas que conforman en grafo que tiene que realizar la librería. En nuestro caso y como se ha indicado en el apartado anterior se van a cargar de un archivo externo que va contener todos estos datos. Dentro de este atributo en "nodes" es donde se tienen que cargar los datos de los nodos y en "edges" los datos de las aristas. En el ejemplo que se va a mostrar en la imagen 12 los datos de los nodos se encuentran en el objeto demoNodes y los datos de las aristas en el objeto demoEdges.

```
elements: {

  nodes: demoNodes,
  edges: demoEdges
},
```

Imagen 12. Ejemplo de uso del atributo elements

- **Layout:** en este atributo es donde se le da un nombre al grafo que le identifica y también donde se pueden añadir funciones para ordenar los nodos del grafo o funciones para posicionar los nodos dentro del grafo. Además, dentro de este atributo se pueden dar otras características especiales al grafo si se desea. A continuación, en la imagen 13 se muestra un ejemplo del uso de este atributo.

```
layout: {
  name: 'grid',
  //padding: 30, // padding used on fit
  boundingBox: undefined, // constrain layout bounds; { x1, y1, x2, y2 } or { x1, y1, w, h }
  avoidOverlap: true, // prevents node overlap, may overflow boundingBox if not enough space
  rows: 15, // force num of rows in the grid
  columns: 15, // force num of cols in the grid
  sort: function(node){ return node.data('weight')*100 ;},
  animate: false, // whether to transition the node positions
  animationDuration: 500, // duration of animation in ms if enabled
  position: function(node){ return {row:node.data('row'), col:node.data('col')};},
},
```

Imagen 13. Ejemplo de uso del atributo layout

Una vez completados todos los atributos anteriormente indicados con todas las características deseadas el grafo se genera en la misma zona de la página donde se había indicado y con estas características.

4.6.3 Funcionalidades sobre el grafo

Al finalizar la tarea de generar el grafo principal se han desarrollado una serie de funcionalidades que se van a realizar sobre este grafo principal. Estas funcionalidades son las necesarias para que la aplicación cumpla los objetivos deseados.

Estas funcionalidades son

- Añadir un hermano
- Añadir una pareja
- Añadir un hijo
- Borrar un nodo (paciente)
- Editar la información de un nodo (paciente)

Antes de entrar al detalle en el desarrollo de cada una de estas funcionalidades hay que apuntar que se ha hecho uso de un plugin de la librería Cytoscape.js que nos permite crear un menú circular en el momento de pulsar alguno de los nodos del grafo de tal manera que cada una de estas funcionalidades anteriormente descritas sea una opción de este menú y así tener un diseño interactivo y fácil para el usuario.

El plugin que se usó para este menú circular es cytoscape-cxtmenu.js. Se declara dentro del programa de la siguiente manera cy.cxtmenu siendo el cy el objeto cytoscape que contiene el grafo principal.

Esto objeto del menú circular tiene dos atributos principales que son

- **Selector:** es donde se declara sobre que objeto del grafo va salir este menú circular. En nuestro caso los nodos por lo que se pone de la siguiente manera
  selector: 'node'

- **Commands:** en este atributo es donde se declaras las funciones que va tener el menú. Cada una de las funcionalidades se declara en dos partes dentro del atributo Commands que son el content y el select. En el content se declara el nombre y la forma de cómo va aparecer en el menú y en el select es donde se realiza el código de la función a ejecutar una vez elegida tal opción en el menú. En el siguiente ejemplo mostrado en la imagen 14 podemos ver un ejemplo de cómo se declara alguna función

```
cy.cxtmenu({
      selector: 'node',
      commands: [

          {
              content: '<span class="fa fa-pencil-square-o">Edit</span>',
              select: function(){
                  document.getElementById('myNode').value = '#'+this.id();
                  cy.startBatch();
```

Imagen 14. Ejemplo de declaración de una función en Cytoscape

Una vez desarrolladas todas las funcionalidades anteriormente nombradas dentro del menú circular al presionar alguno de los nodos dentro del grafo en nuestra aplicación el menú aparece dándonos la oportunidad de ejecutar cualquiera de estas funcionalidades sobre el nodo presionado y tiene la siguiente forma

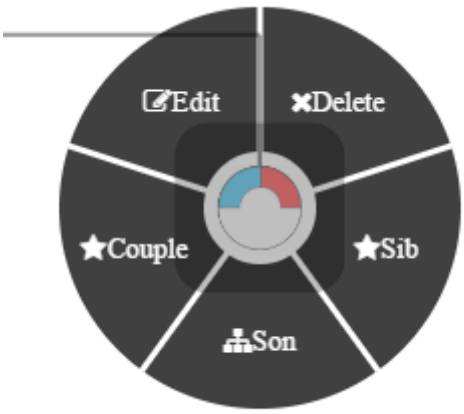


Imagen 15. Menú circular de las funcionalidades

Donde la opción Sib es la función de añadir un hermano, la opción Son es la de añadir un hijo, la opción Couple es la función de añadir pareja, la opción Edit es la función de editar la información del nodo presionado y la opción Delete es la función de borrar el nodo presionado.

Una vez explicado el funcionamiento del menú y como se va poder llamar a estas funcionalidades vamos a empezar a explicar cada una de ellas.

4.6.3.1 Añadir un hermano

Esta función nos permite añadir un nuevo nodo al grafo de forma que este nuevo nodo este en la misma fila que el nodo desde que se ejecuta la función y uniendo este nuevo nodo con el nodo de gestación que sale del matrimonio de los padres para que así siga la forma que exige la nomenclatura que se sigue para generar el pedigrí de los pacientes.

Una vez generado el nuevo nodo se ejecuta el modal de añadir nuevo nodo que tiene la página, que consiste en una tabla donde nos permite meter todos los datos del nuevo nodo (paciente) que se acaba de añadir ya que en la función cuando se genera el nuevo nodo todos estos campos que contienen los datos del nuevo nodo están vacíos y al rellenar dicha tabla se rellenan con los datos que desea el usuario desde el nombre pasando por la fecha de nacimiento , sexo estado y hasta los nuevos diagnósticos que se desean añadir en caso de quererlo.

4.6.3.2 Añadir pareja

Esta función es la función responsable a añadir un nuevo nodo para formar un matrimonio con el nodo que ejecuta la función creando así entre media un tercer nodo invisible al usuario con el fin de que a partir de este nodo se puedan crear tantas gestaciones para crear hijos como sea necesario y así mantener la forma del grafo que exige la nomenclatura. Para explicar mejor el funcionamiento de este nodo vamos a observar las siguientes imágenes 16 y 17.

Imagen 16. El nodo invisible al declarar una relación de pareja

Como se puede ver en la foto la zona subrayada en gris es donde se encuentra el nodo invisible que hemos apuntado en el apartado anterior. En el momento de crear un nuevo hijo veremos la necesidad de tener este nodo.

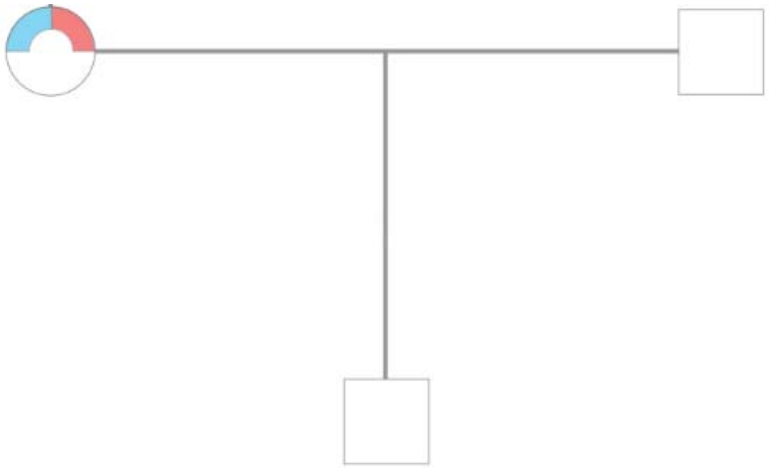


Imagen 17. Conectar el hijo a la relación de pareja

Como se puede ver la necesidad de este nodo es para que en el momento de añadir un hijo mantener la forma que exige la nomenclatura y mantener los niveles diferentes y así poder identificar claramente las diferencias entre las diferentes relaciones que existen en la representación.

Al ejecutar la función y crearse este nuevo nodo de pareja, como es el caso en la función anterior, se ejecuta el modal de añadir un nodo nuevo con la tabla para poder añadir toda la información necesaria a este nuevo nodo ya que en el momento de crearlo se crea con todos estos campos de información vacíos.

4.6.3.3 Añadir hijo

Esta función es la responsable de añadir un nuevo nodo que representa un hijo de un matrimonio por lo que este nuevo nodo debe estar en un nivel inferior al nivel del nodo que ejecuta la función para así mantener la forma de representación de la nomenclatura.

Para que un nodo pueda ejecutar esta función dicho nodo tiene que tener una pareja dicho de otra manera tiene que a ver una relación entre este nodo y otro dentro del grafo para poder generar los nodos hijo a partir de esta relación como se explicó en el apartado anterior.

En el momento de crear un nodo hijo se crea también otro nodo invisible que es el nodo de gestación para que este nodo se conecte con los nodos de gestación de los otros hijos (hermanos del nuevo nodo) y de esta forma mantener la estructura de jerarquía que tiene que tener el grafo. En las siguientes imágenes 18 y 19 se mostrará el funcionamiento de este nodo.

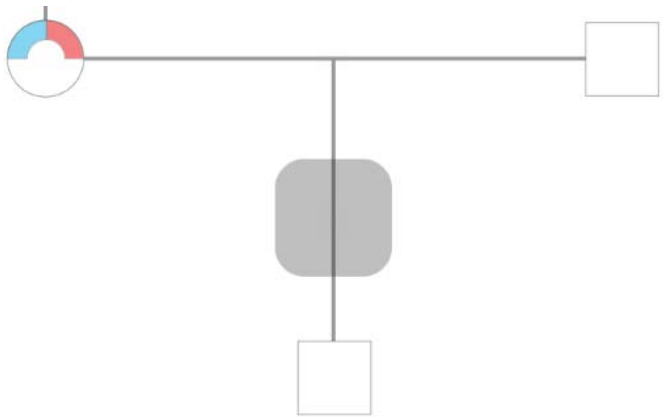


Imagen 18. Nodo invisible de gestación

Este nodo invisible se encuentra en esta zona subrayada en gris y en el momento de añadir otro hijo el resultado es el siguiente

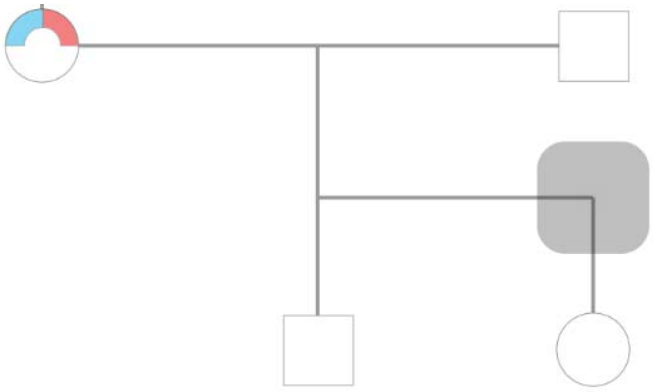


Imagen 19. Nodo invisible de gestación

Esta nueva zona subrayada es donde se encuentra el nuevo nodo de gestación del nuevo hijo que se conecta con el otro nodo de gestación invisible formando así la forma deseada por la nomenclatura para representar el pedigrí de los pacientes.

4.6.3.4 Borrar un nodo

La función de borrar un nodo es una función básica donde se borra un nodo del grafo utilizando la función de borrar del Cytoscape.js que es "*cy.remove(node)*" donde el "cy" es el objeto principal del grafo de Cytoscape y el "node" es el objeto donde se guarda el identificador del nodo a eliminar.

En esta función había que realizar una serie de comprobaciones antes de realizar el borrado del nodo ya que si un nodo tiene nodos hijo este nodo no se puede borrar para no dejar estos nodos hijo descolgados y así perder la forma y la conectividad del grafo.

Una vez realizadas las comprobaciones necesarias se borra el nodo y todos los nodos invisibles de gestación que están conectados con él y así borrar todas las aristas que le conectan y no dejar ninguna arista o nodo invisible suelto por el grafo.

4.6.3.5 Editar la información de un nodo

La última función que nos queda de las funcionalidades es la función de editar la información de un nodo donde nos da la posibilidad de cambiar cualquiera de los datos del paciente (nodo).

Dicha información que se puede modificar es el nombre del paciente, fecha de nacimiento, genero, estado y los diagnósticos que tiene el paciente o las que el usuario quiera añadir.

Al ejecutar la función se ejecuta el modal de modificar los datos del nodo que tiene la página y que consiste en la misma tabla de añadir un nuevo nodo, pero contiene los datos de este nodo en vez de que todos los campos estén vacíos.

4.6.4 Funcionalidades adicionales

La aplicación web además de las funcionalidades básicas descritas anteriormente realiza una serie de funcionalidades fuera del grafo que sirven para completar todos los objetivos esperados de la aplicación.

Estas funcionalidades están implementadas al igual que las otras funcionalidades en JavaScript y utilizando la librería de jQuery.

4.6.4.1 Exportar el pedigrí humano del paciente

Esta función permite al personal médico que está utilizando la aplicación exportar el grafo entero que representa el pedigrí humano del paciente en un archivo externo y así el personal médico pueda utilizarlo en lo que desea.

Para la realización de esta función se utilizaron una serie de librerías adicionales escritas también en JavaScript. Dichas librerías son

- **html2canvas.js:** esta librería realizaba el cambio de una etiqueta HTML a un elemento canvas para más adelante poder generarlo como imagen.

- **canvas-to-blob.min.js:** una vez realizado el cambio a un elemento canvas para poder exportar dicha imagen era necesario pasar este elemento a un objeto blob de JavaScript y allí entra el trabajo de esta librería.

- **FileSaver.min.js:** para terminar esta función y tener este archivo externo con la imagen del grafo se utiliza esta librería que es la responsable de exportar el objeto blob a un archivo externo y tenerlo en el ordenador local de donde se está ejecutando la aplicación.

Este archivo externo que va contener la imagen del grafo es un archivo de tipo PNG ya que lo que se va tener es una imagen del grafo.

Para realizar esta función hay un botón en la parte superior de la página donde se encuentra representado el grafo este botón despliega un menú donde se puede elegir la opción de exportar el grafo como se puede ver en la imagen 20 que tenemos a continuación.

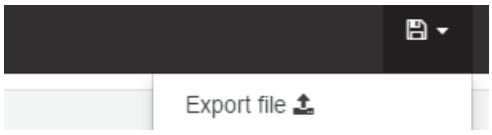


Imagen 20. Opción de exportar el grafo

4.6.4.2 Exportar la información del paciente

Esta función es la encargada en exportar el panel que contiene toda la información del paciente. Esta información del paciente es la que se encuentra en el modal de mostración de la información del paciente que se ejecuta en el momento de pinchar en algún nodo en el grafo mostrando así una tabla con toda la información y diagnósticos de este paciente.

Esta función está escrita en JavaScript y realiza los mismos pases descritos en la función anterior de exportar la imagen del grafo con la diferencia de que esta función exporta la imagen del modal de la información del paciente.

Para poder ejecutar esta funcionalidad se incluyó un botón dentro del mismo modal de mostración de la información que nos permite realizar esta función y así el personal médico pueda utilizar esta foto de la información del paciente en lo que desea.

4.6.4.3 Centrar el grafo

Es una funcionalidad básica y sencilla pero bastante útil ya que donde se representa el grafo el usuario puede mover el grafo y aumentarlo si lo considera necesario y una vez que acabe de realizar este movimiento necesita volver a centrar este grafo y allí es donde es útil esta función ya que es la responsable de realizar dicha funcionalidad.

Para la implementación de esta función se utilizaron algunas de las funciones que nos ofrece Cytoscape.js como "*cy.center()*" donde "cy" es el objeto cytoscape que contiene el grafo entero.

Para ejecutar esta función se incluyó un botón en la parte superior de donde se representa el grafo que realiza esta centralización.

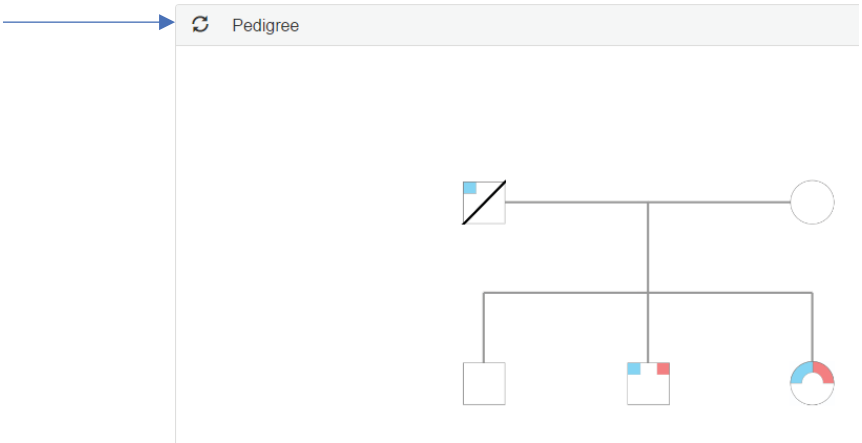


Imagen 21. Botón de centralización del grafo

Capítulo 5. Pruebas y resultados

5.1 Introducción

En este capítulo se detallan algunas de las pruebas realizadas para comprobar el correcto funcionamiento de la aplicación web y que aseguran que se cumplan todos los objetivos y requisitos de la aplicación.

Para realizar las pruebas en este capítulo se va trabajar sobre un pedigrí humano de una familia de pacientes que consta de un matrimonio de un varón y una mujer y que de dicho matrimonio nacieron tres hijos. Dicho varón se encuentra en estado fallecido y tiene un diagnostico añadido y donde uno de los hijos tiene otros dos diagnósticos añadidos también. El pedigrí de esta familia en la aplicación web se muestra de la forma que se puede ven en la imagen 22.

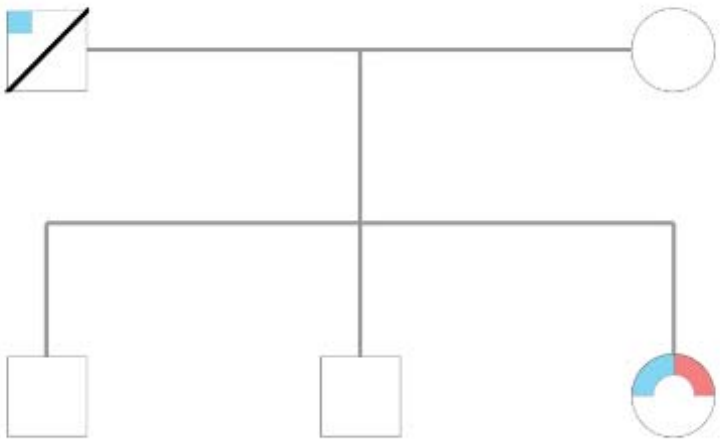


Imagen 22. Grafo inicial de las pruebas

Sobre este pedigrí se van a realizar todas las funcionalidades descritas en el capítulo anterior para poder asegurar el correcto funcionamiento de la aplicación y el complimiento de los objetivos del trabajo y se va a mostrar la evolución de este grafo después de la ejecución de cada una de las funcionalidades.

5.2 Añadir nuevo hermano

Al pulsar encima de alguno de los nodos hijo se muestra el menú circular de las funcionalidades como se puede ver en la imagen 23.

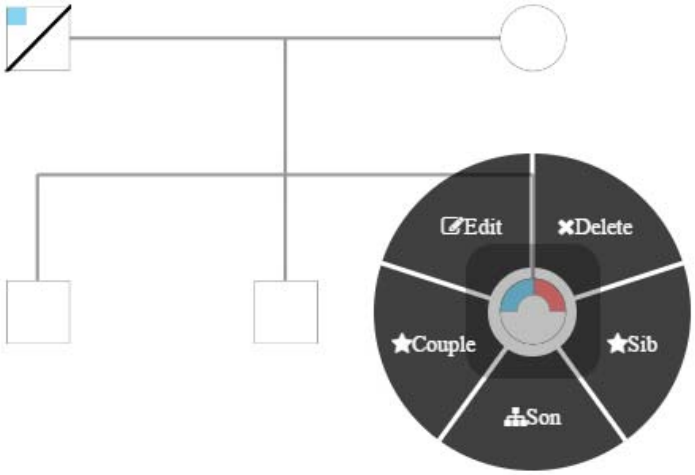


Imagen 23. Menú circular de funcionalidades

Una vez mostrado dicho menú se elige la opción “Sib” para realizar la función de crear un nuevo hermano para dicho nodo. Al ejecutar dicha función aparece el modal de creación de un nuevo nodo para poder meter la información del nuevo nodo agregado y se añade un nuevo nodo al grafo representando este nuevo hermano añadido como se puede ver en la imagen 24.

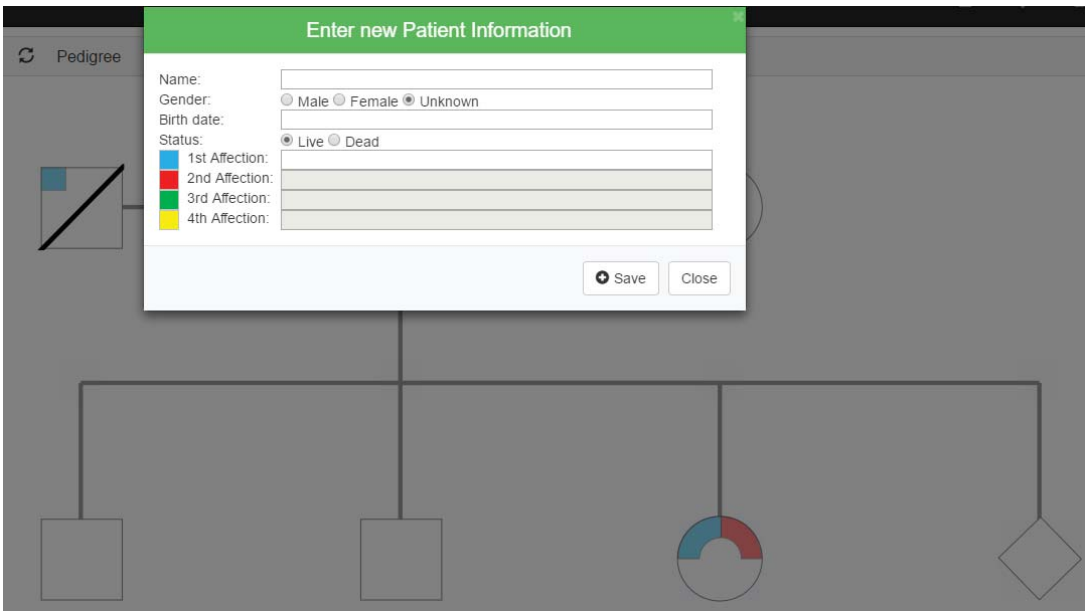


Imagen 24. Modal de creación de nuevo nodo

Al meter los datos del nuevo paciente el grafo final después de la ejecución de la función quedara de la forma que se puede ver en la imagen 25.

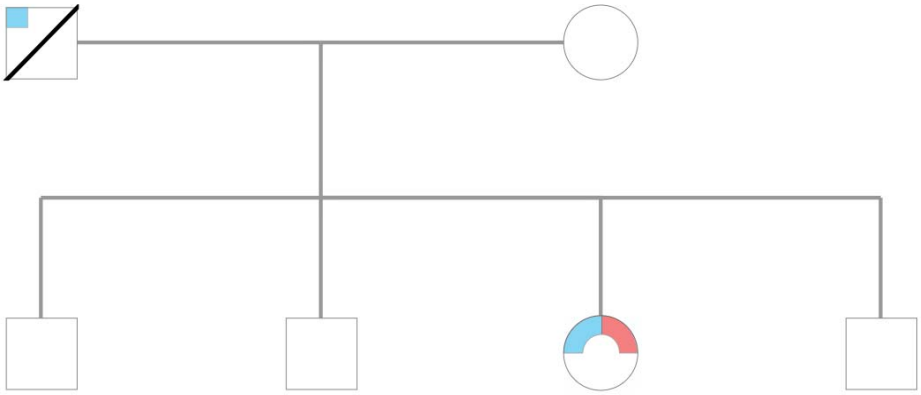


Imagen 24. Grafo después de añadir nuevo hermano

5.3 Añadir nueva pareja

Para la prueba de esta funcionalidad se le va añadir una pareja a uno de los hijos desde el grafo inicial.

Se realiza la misma acción descrita en el apartado anterior para sacar el menú circular de las funcionalidades como se mostró en la imagen 23 pero con la diferencia de que esta vez se va elegir la opción “Couple” para realizar la función de agregar una nueva pareja a este nodo.

Al ejecutar dicha función va aparecer el modal de creación de un nuevo nodo y se va agregar este nuevo nodo al grafo como se puede ver en la imagen 25.

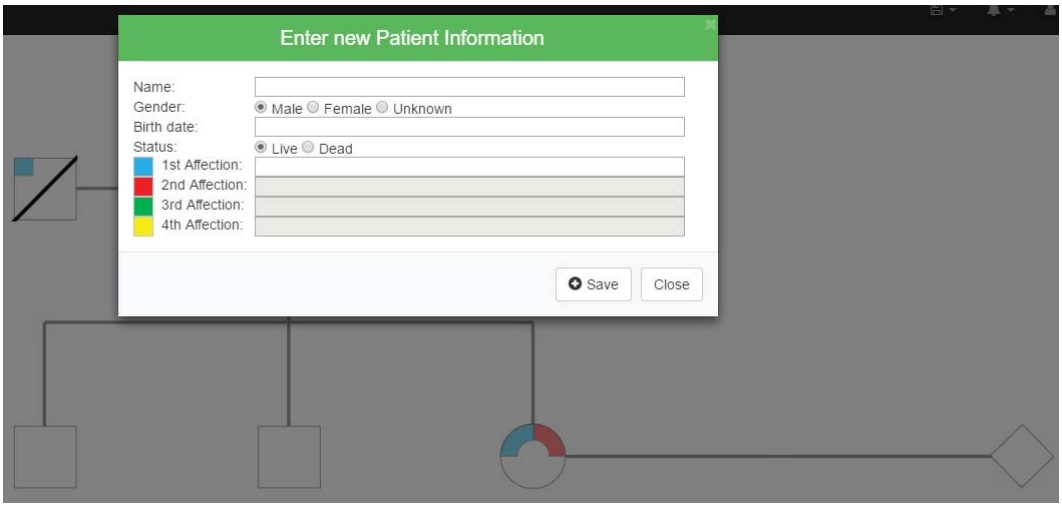


Imagen 25. Modal de creación de nuevo nodo

El grafo final después de añadir los datos de este nuevo nodo quedara finalmente de la forma que se puede ver en la imagen 26.

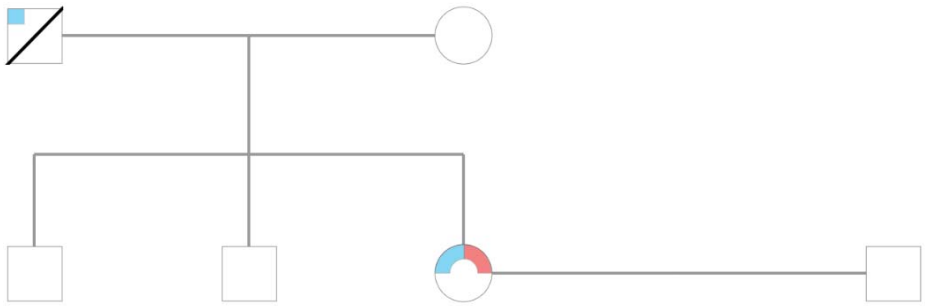


Imagen 26. Grafo después de añadir nueva pareja

5.4 Añadir un hijo

Para probar esta funcionalidad se utilizará este último matrimonio creado en el apartado anterior agregando a este matrimonio un nuevo nodo hijo.

De la misma manera descrita en los dos apartados anteriores se pulsamos sobre alguno de los dos nodos que forman el matrimonio nos saldrá el menú circular de las funcionalidades y de este menú se elige la opción "Son" para ejecutar la función de añadir un nuevo hijo a dicho matrimonio.

Una vez ejecutada esta función se generará el modal de creación de nuevo nodo para agregar los datos de este nuevo nodo hijo y se añadirá este nuevo nodo al grafo como se puede ver en la imagen 27.

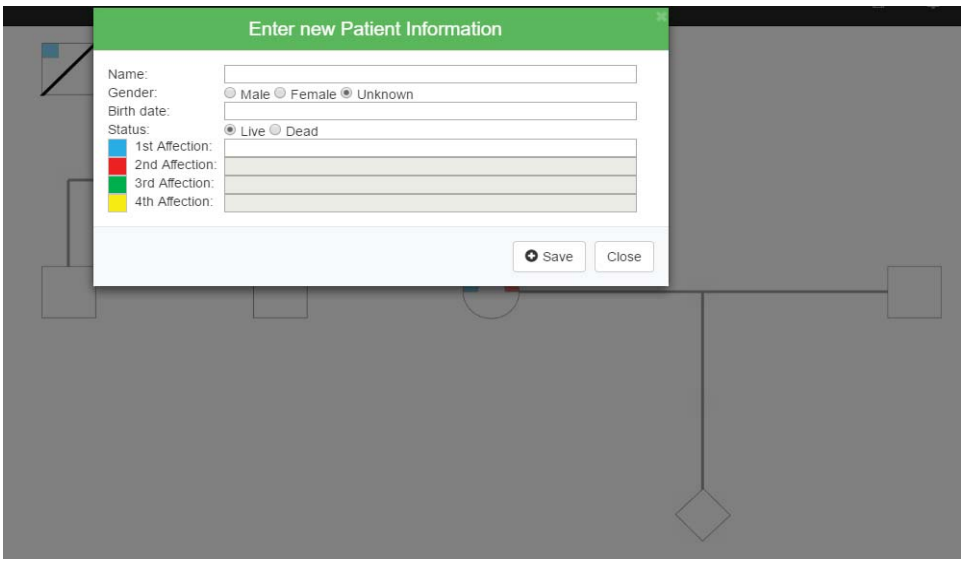


Imagen 27. Modal de creación de nuevo nodo

El grafo final después de la ejecución de esta función quedara de la forma que se puede observar en la imagen 28.

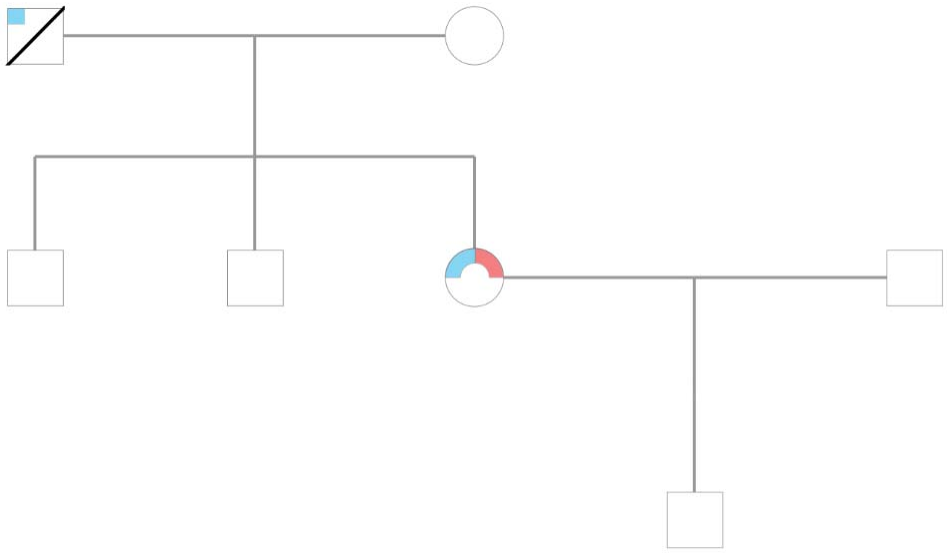


Imagen 28. Grafo final después de añadir un nuevo hijo

5.5 Borrar un nodo del grafo

Para probar la funcionalidad do borrar algún nodo del grafo se va ejecutar esta función sobre el grafo inicial representado en la imagen 22 para eliminar alguno de los nodos hijo.

Si se pulsa sobre el nodo que se desea eliminar como por ejemplo el nodo hijo que se encuentra en el medio se mostrara el menú circular de las funcionalidades y se elige desde allí la opción “Delete” como se muestra en la imagen 29.

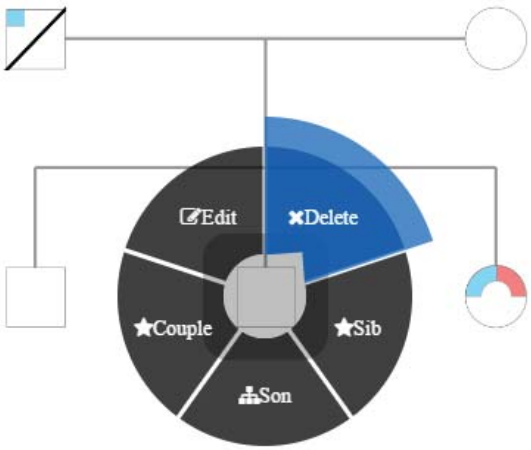


Imagen 29. Menú circular de las funcionalidades

Una vez seleccionada esta opción desde el menú se ejecutará la función de borrar el nodo y el grafo final se quedará como se puede ver en la imagen 30.

Imagen 30. Grafo final después de borrar el nodo

5.6 Editar la información de un nodo

Sobre el grafo inicial se va editar la información del primer nodo hijo añadiéndole un nuevo diagnóstico y para ello se pulsa encima del nodo y se elige la opción de “Edit” para que se ejecute esta función.

Al ejecutar esta función se va mostrar el modal de editar la información de un nodo con los datos del nodo y donde se puede cambiar cualquiera de los datos que se desea como se puede observar en la imagen 31

Imagen 31. Modal de editar la información del nodo

Una vez en este modal se pude cambiar cualquiera de los datos o añadir o eliminar diagnósticos. En este caso de prueba se va añadir un nuevo diagnostico por lo que se va al primer campo de diagnósticos vacío y se escribe el nombre de diagnóstico que se quiere añadir. La aplicación ofrece la oportunidad de auto completar el nombre del diagnóstico mientras se está escribiendo como se puede ver en la imagen 32.

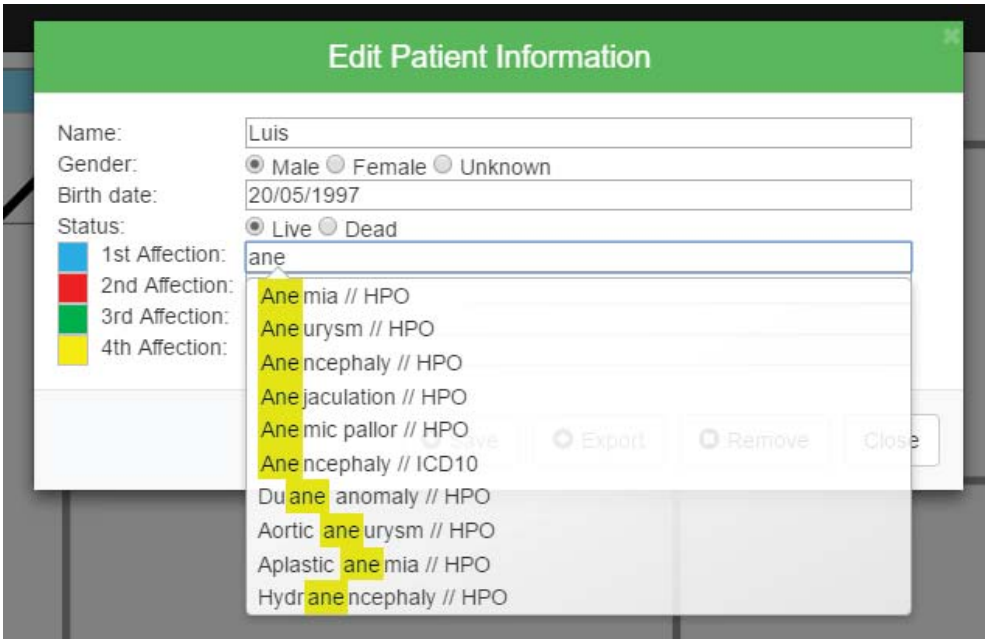


Imagen 32. Modal de editar la información del nodo

Una vez añadido este diagnóstico y terminada la modificación del nodo se guarda pulsando en el botón “save” y el grafo final se quedará de la forma que se puede ver en la imagen 33.

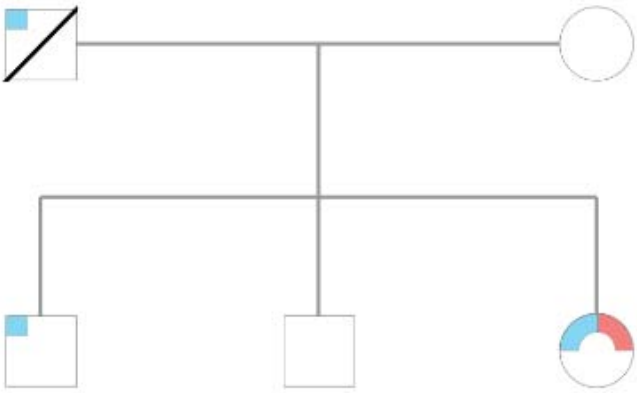


Imagen 33. Grafo final después de la modificación

Capítulo 6. Conclusiones y futuras mejoras

Una vez realizado el presente trabajo fin de grado se recogen en este capítulo las conclusiones generales obtenidas, así como las posibles mejoras que se pueden aplicar a la aplicación.

6.1 Conclusiones generales

El propósito de este Trabajo de Fin de Grado era construir una aplicación web que representa el pedigrí humano de los pacientes médicos siguiendo una nomenclatura y un estándar reconocido internacionalmente para que así todo el personal médico de diferentes sitios del mundo pueda entender dicho pedigrí y así se pueda intercambiar la información de manera más fácil y sencilla.

El primer paso para poder desarrollar dicho trabajo era estudiar la nomenclatura de "Standarized Human Pedigree Nomenclature" para saber la forma que tiene que tener el grafo, las restricciones que tiene el grafo y las forma de representar cada tipo de relación entre los pacientes.

Una vez comprendida la forma que tiene que tener el grafo se empezó a estudiar la librería de Cytoscape.js, es la librería especializada en dibujar grafos, para poder ver cómo utilizarla para generar grafos de la forma que se necesita y se llegó a la conclusión de que es una librería muy potente con muchas funcionalidades que permitirán crear lo que se necesita representando los pacientes como nodos y uniéndolos para formar las relaciones entre ellos.

A medida que se ha ido desarrollando el proyecto se encontraron muchas plugins que ayudan a mejorar el grafo y hacerlo más interactivo y fácil de usar para el personal médico. La disponibilidad de tantos plugins e información para la librería Cytoscape.js ha sido debido a que es una librería en auge y su uso está creciendo mucho por lo que demuestra que ha sido una buena elección para el desarrollo del proyecto.

Como en todo proyecto había que fijar una lista de requisitos que tiene que cumplir el sistema para asegurar que se cumplan los objetivos y para que se puedan realizar las pruebas finales para asegurar el cumplimiento de dichos requisitos y eso hace que esta fase tenga una importancia vital para finalizar la aplicación de forma correcta.

La aplicación tenía que realizar una serie de funcionalidades para poder cumplir los objetivos requeridos. Estas funcionalidades son las que se numeras a continuación

- **Añadir un hermano/hermana:** Es la funcionalidad responsable de añadir un nuevo paciente relacionado con un paciente que ya se encuentra representado en el grafo y conectarlos mediante una relación que muestra que son hermanos.

- **Añadir una Pareja:** Es la funcionalidad responsable de añadir un nuevo nodo conectado con otro nodo ya existente y añadir entre los dos un nodo invisible, es el nodo de gestación, para que en el futuro si se desea añadir hijos a la relación se conecten a dicho nodo.

- **Añadir un hijo/hija:** Esta función permite añadir un nuevo nodo hijo/hija a una relación que ya existe en el grafo.

- **Editar la información del paciente:** Es la función que permite al usuario medico editar y modificar la información de un paciente o añadirle un nuevo diagnóstico.

- **Borrar un nodo:** Dar la oportunidad al usuario de eliminar algún paciente del grafo.

- **Exportar la información:** Dicha función le ofrece al usuario la oportunidad de exportar tanto el grafo entero que representa el pedigrí y el historial médico del paciente como exportar solo la información de un paciente en concreto y así poder usar dicha información en lo que necesita como añadirla por ejemplo a informes médicos.

En conclusión, el desarrollo de este proyecto ha sido una gran oportunidad de aprender a usar muchas tecnologías nuevas, así como aprovechar muchas de las cosas aprendidas durante la carrera como por ejemplo la realización de la fase de especificar los requisitos y también los conceptos técnicos a la hora de programar. De las cosas más importantes adquiridas ha sido aprender a planificar un trabajo y como llevarlo por todas las fases y teniendo en cuenta los periodos de tiempo necesarios para realizar cada una de las tareas.

6.2 Futuras mejoras

Dicho proyecto Puede tener varias mejoras en el futuro ya que la nomenclatura que se utilizó como referencia tiene bastantes más casos especiales que se puedan dar y debido al corto periodo de tiempo no se han podido desarrollar todas. Uno de estos casos especiales que se pueden añadir es añadir un nuevo hijo adoptado que tiene una forma especial de representarlo, así como la posibilidad de añadir más de una pareja a un nodo.

Debido a que este proyecto se desarrolló utilizando una de las librerías, Cytoscape.js, que más está mejorando últimamente pues de las posibles mejoras es actualizar la versión usada ya que el crecimiento y las nuevas versiones de esta librería no paran de aumentar.

El siguiente paso teniendo ya esta parte del Front-end será integrarla con la parte Back-end del proyecto para que así tener la aplicación totalmente lista.

Bibliografía

[1] Network library for analysis and visualization Cytoscape.js. http://js.cytoscape.org/

[2] jQuery API documentation. http://api.jquery.com/

[3] Bootstrap Library. http://getbootstrap.com/

[4] Tutorials HTML. https://www.w3schools.com/html/

[5] Tutorials CSS. https://www.w3schools.com/html/html_css.asp

[6] Apache Tomcat documentation. https://tomcat.apache.org/

[7] JSON documentation. http://www.json.org/

[8] JavaScript Tutorials. https://www.w3schools.com/js/default.asp

[9] jQuery Tutorials. https://www.w3schools.com/jquery/default.asp

[10] Bootstrap Tutorials. https://www.w3schools.com/bootstrap/default.asp

[11] Cytoscape.js Examples. http://blog.js.cytoscape.org/2016/05/24/getting-started/

[12] Standardized Human Pedigree Nomenclature http://geneticcounselingtoolkit.com/cases/pedigree/Bennett%20JGC%202008%20-%20Standardized%20Human%20Pedigree%20Nomenclature%20-%20Update%20and%20Assessment%20of%20the%20Recommendations%20of%20the%20National%20Society%20of%20Genetic%20Counselors.pdf

[13] IEEE Std. 830-1998 documentation. https://www.fdi.ucm.es/profesor/gmendez/docs/is0809/ieee830.pdf

[14] cytoscape.js-cxtmenu plugin. https://github.com/cytoscape/cytoscape.js-cxtmenu

[15] html2canvas library and examples. https://html2canvas.hertzen.com/

[16] FileSaver.js library. https://github.com/eligrey/FileSaver.js/

Este documento esta firmado por

Firmante	CN=tfgm.fi.upm.es, OU=CCFI, O=Facultad de Informatica - UPM, C=ES
Fecha/Hora	Fri Jul 07 16:59:22 CEST 2017
Emisor del Certificado	EMAILADDRESS=camanager@fi.upm.es, CN=CA Facultad de Informatica, O=Facultad de Informatica - UPM, C=ES
Numero de Serie	630
Metodo	urn:adobe.com:Adobe.PPKLite:adbe.pkcs7.sha1 (Adobe Signature)