

APLICACIÓN WEB PARA LA PREDICCIÓN Y RECOMENDACIÓN DE TRATAMIENTO PARA EL CÁNCER DE PRÓSTATA

WEB APPLICATION FOR THE PREDICTION AND
RECOMMENDATION OF TREATMENT FOR PROSTATE
CANCER.

Gonzalo Hernández Hernández

GRADO EN INGENIERÍA DEL SOFTWARE FACULTAD DE INFORMÁTICA.

Universidad Complutense de Madrid



TRABAJO DE FIN DE GRADO

Curso 2021 – 2022

Directores

Antonio Sarasa Cabezuelo

Víctor Manuel Carrero López

Agradecimientos.

A mi familia y amigos por acompañarme hasta el final.

Resumen

Se ha creado una aplicación web que permite a los usuarios obtener una predicción sobre la probabilidad que un paciente tenga cáncer de próstata, y de que este sea más o menos agresivo con vistas a indicar uno u otro de los tratamientos posibles, a partir de parámetros clínicos, analíticos, radiológicos, etc., reflejados en su historia clínica, además de recomendar posibles acciones a seguir para su vigilancia.

La aplicación permite a los usuarios crear historiales de visitas médicas, añadirles nuevas visitas a esos historiales y modificarlos si fuera necesario.

La aplicación se ha diseñado para que sea lo más intuitiva y fácil de usar posible.

Palabras Clave

- Historial
- Expediente
- Calculadora
- Probabilidad
- Biopsia
- Tratamiento

Abstract

We created a web application that allows users to obtain a prediction about the probability of a patient, and the level of aggression of the cancer to be able to indicate a possible treatment, based on clinical, analytical, radiological, etc. parameter, reflected in his clinical history, in addition to recommending possible treatments or actions to follow for surveillance.

The application allows users to create medical records of patient's visits, add new visits to these records and modify them if necessary.

The app has been designed to be as intuitive and easy to use as possible.

Palabras Clave

- History
- Record
- Calculator
- Probability
- Biopsy
- Treatment

Índice

Capítulo 1 Introducción.....	10
1.1 Motivación	10
1.2 Objetivos	10
1.3 Estructura de la memoria.....	11
1.4 Diagrama de Gantt	11
Chapter 1 Introduction.....	13
1.1 Motivation.....	13
1.2 Objetives	13
1.3 Memory structure	13
1.4 Gantt's Chart	14
Capítulo 2 Estado del Arte.....	16
Capítulo 3 Tecnologías empleadas.....	18
3.1 React.....	18
3.2 Django	18
3.3 XAMPP	18
3.4 Heroku.....	18
3.5 GIT	18
3.6 Python	19
Capítulo 4 Especificación de requisitos	20
4.1 Actores	20
4.2 Casos de uso	20
4.2.1 Modulo Usuario.....	20
4.2.2 Modulo Historiales Médicos	23
Capítulo 5 Arquitectura y modelo de datos.....	30
5.1 Arquitectura	30
5.2 Modelo de datos	30
Capítulo 6 Diseño	34
6.1 Colores.....	34
6.2 Funcionalidades de la aplicación.....	34
6.2.1 Crear historial de una visita.....	34
6.2.2 Lista de historiales.....	37
6.2.3 Información detallada de un historial	38
6.2.4 Obtener Predicción	40
6.2.5 Calculadora.....	43
6.2.6 Información del historial	45

Capítulo 7 Desarrollo y documentación de API REST	49
7.1 MedicalHistorys.....	49
Ahora se describirá cada endpoint.....	49
7.2 Prostatescreen_api.....	49
Ahora se describirá cada endpoint.....	50
7.3 Users.....	50
Capítulo 8 Conclusiones y trabajo a futuro	51
8.1 Conclusiones.....	51
8.2 Trabajo a futuro	51
Chapter 8 Conclusions and future work	53
8.1 Conclusions	53
8.3 Future work	53
Capítulo 9 Aportaciones individuales.....	54
9.1 Gonzalo Hernández Hernández.....	54
9.2 Roberto Torres Prensa	54
Bibliografía.....	55
ANEXO I Guía del usuario.....	57

Índice de figuras

Figura 1- Diagrama de Gantt	12
Figura 2 - Gantt's diagram.....	15
Figura 3 - Resultado de Prostate Cancer Prevention TrialRisk Calculator.....	16
Figura 4 - Resultado de la calculadora de SWOP	16
Figura 5 - Resultado de la calculadora del Memorial Sloan Kettering Center	17
Figura 6 - Resultado de la calculadora de prostaterisk.c	17
Figura 7 - Diagrama de casos de uso módulo usuario.....	20
Figura 8 - Diagrama de casos de uso módulo historiales médicos	23
Figura 9- Diagrama del patrón MVC.....	30
Figura 10 - Models.py del módulo users	31
Figura 11 - Vista general de la función crear nuevo historial.....	34
Figura 12 - TabPanel.....	35
Figura 13 - Todas las Tabs	35
Figura 14 - Función para enviar un historial al back	36
Figura 15 - Función para crear un nuevo historial	36
Figura 16 - Imagen de las tarjetas que representan todos los historiales médicos.....	37
Figura 17 - Función GetMedicalHistroy.....	37
Figura 18 - Modelo para serializar un historial médico.....	38
Figura 19 - Botón Actualizar	39
Figura 20 - Función para hacer los campos modificables	39
Figura 21 - Función para enviar el historial al back para actualizarlo	39
Figura 22 - Función para actualizar un historial	39
Figura 23 - Distintas visitas.....	40
Figura 24 - Función para mostrar otras visitas.....	40
Figura 25 - Buscar un historial para obtener predicción.....	41
Figura 26 - Gráfica resultante de la predicción	41
Figura 27 - Mostrar gráfico de probabilidad	42
Figura 28 - Función que calcula la probabilidad de biopsia positiva.....	43
Figura 29 - Probabilidad de cáncer de bajo grado con biopsia negativa	44
Figura 30 - Probabilidad de cáncer de alto grado con biopsia negativa	44
Figura 31 - Probabilidad de cáncer de bajo grado con biopsia positiva.....	45
Figura 32 - Probabilidad de cáncer de alto grado con biopsia positiva	45
Figura 33 - Información de la calculadora.....	46
Figura 34 - Información de Clínico Patológicas	47
Figura 35 - Información de la biopsia.....	47
Figura 36 - Cálculo del estadio	48
Figura 37 - Información de la prostatectomía.....	48
Figura 38 - Tipos de endpoints utilizados en medicalHistorys	49
Figura 39 - Descripción de endpoints utilizados en medicalHistorys.....	49
Figura 40 - Tipos de endpoints utilizados en prostascreen_api.....	49
Figura 41 - Descripción de endpoints utilizados en prostascreen_api.....	50
Figura 42 - Tipos de endpoints utilizados en users	50
Figura 43 - Descripción de endpoints utilizados en users	50
Figura 44 - Pantalla de login	57
Figura 45- Home	57
Figura 46- Información personal	58

Figura 47 - Modificar información personal.....	58
Figura 48 - Nuevo historial	59
Figura 49 - Grupos de variables.....	59
Figura 50 - Lista de historiales.....	60
Figura 51 - Información detallada del historial	60
Figura 52 - Diferentes vistas.....	61
Figura 53 - Vista general al obtener una predicción	61
Figura 54 - Botones con la información de la calculadora y el historial.....	62

Índice de tablas

Tabla 1 - Inicio de sesión	21
Tabla 2 - Cerrar sesión.....	21
Tabla 3 - Mostrar perfil de usuario.....	22
Tabla 4 - Modificar información del perfil.....	22
Tabla 5 - Ver todos los historiales médicos.....	23
Tabla 6 - Buscar historial médico	24
Tabla 7 - Ver información detallada de un historial.....	25
Tabla 8 - Modificar un historial médico	25
Tabla 9 - Archivar un historial médico.....	26
Tabla 10 - Crear nuevo historial	27
Tabla 11 - Obtener predicción de un historial	28
Tabla 12 - Obtener información detallada de un historial.....	29

Capítulo 1 Introducción

En este primer capítulo de la memoria se explicará cual es la motivación para realizar este proyecto, los objetivos que se pretenden alcanzar, cómo está estructurada la memoria y se mostrara un diagrama de Gantt con las acciones realizadas.

1.1 Motivación

En Estados Unidos el cáncer de próstata es el más común entre los hombres después del cáncer de piel. A aproximadamente uno de cada ocho hombres será diagnosticado con cáncer de próstata a lo largo de su vida.

El cáncer de próstata es la segunda causa principal de muerte en los hombres de los Estados Unidos, después del cáncer de pulmón. Aproximadamente uno de cada 41 hombres morirá por cáncer de próstata.

Pero el cáncer de próstata no cumple los parámetros evolutivos que cualquier tipo de cáncer. En este sentido, un diagnóstico de cáncer de próstata con factores favorables no tiene por qué comprometer la vida de un paciente. Pues puede cursar de forma menos agresiva. Por eso es importante no solo diagnosticar los casos, sino diagnosticar aquellos casos agresivos que puedan acabar a corto plazo con la vida de los pacientes.

Por estos motivos es muy importante detectar el cáncer de próstata clínicamente significativo, frente al cáncer indolente cuanto antes y recomendarle el mejor tratamiento posible al paciente. La aplicación ayudará a los médicos a poder recomendar soluciones en cuanto al diagnóstico y el tratamiento más adecuado a los pacientes gracias al modelo de datos que se podrá crear.

1.2 Objetivos

El objetivo principal de este trabajo es implementar una aplicación que les permita a los médicos ingresar el expediente de un paciente con sospecha de cáncer de próstata, una vez diagnosticado, tener el seguimiento de las visitas de los pacientes y sus expedientes y recibir una recomendación sobre un posible tratamiento de mayor o menor agresividad, dependiendo de aquello que el modelo sea capaz de predecir respecto de la fase en la que se encuentre el paciente.

Los objetivos en específicos son:

- Desarrollar una aplicación que permita de la manera más intuitiva posible ingresar un expediente de un paciente en la aplicación.
- Utilizar una calculadora que recomiende si un paciente se debe realizar una biopsia prostática en función de sus síntomas o, si ya se la ha realizado, decir en qué grado se encuentra el tumor y que posibles tratamientos son recomendables.
- Implementar una API REST para facilitar el acceso a los servicios que ofrece la aplicación.

1.3 Estructura de la memoria

En este apartado se explicará la estructura de la memoria

- **Capítulo 1 Introducción:** En este capítulo se describe la motivación, los objetivos, la estructura de la memoria y se muestra el diagrama de Gantt del proyecto.
- **Capítulo 2 Estado del arte:** En este capítulo se muestran ejemplos de webs donde se pueden calcular el riesgo de cáncer de próstata.
- **Capítulo 3 Tecnologías empleadas:** En este capítulo se describen las tecnologías empleadas para implementar la aplicación.
- **Capítulo 4 Especificación de requisitos:** En este capítulo se describen los actores y los casos de uso.
- **Capítulo 5 Arquitectura y modelo de datos:** En este capítulo se describen la arquitectura de la aplicación y el modelo de datos.
- **Capítulo 6 Diseño:** En este capítulo se explicará el diseño y la implementación de las principales funcionalidades de la aplicación.
- **Capítulo 7 Desarrollo y documentación de API REST:** En este capítulo se explican los diferentes end points utilizados en la API REST.
- **Capítulo 8 Conclusiones y trabajo a futuro:** En este capítulo se exponen las conclusiones del proyecto y se explican las ampliaciones de la aplicación en el futuro.
- **Capítulo 9 Aportaciones individuales:** En este capítulo se explican las aportaciones de cada individuo en el proyecto.
- **Anexo I:** Manual de usuario.

1.4 Diagrama de Gantt

En este apartado se mostrará un diagrama de Gantt (Figura 1) con las acciones realizadas durante este proyecto.

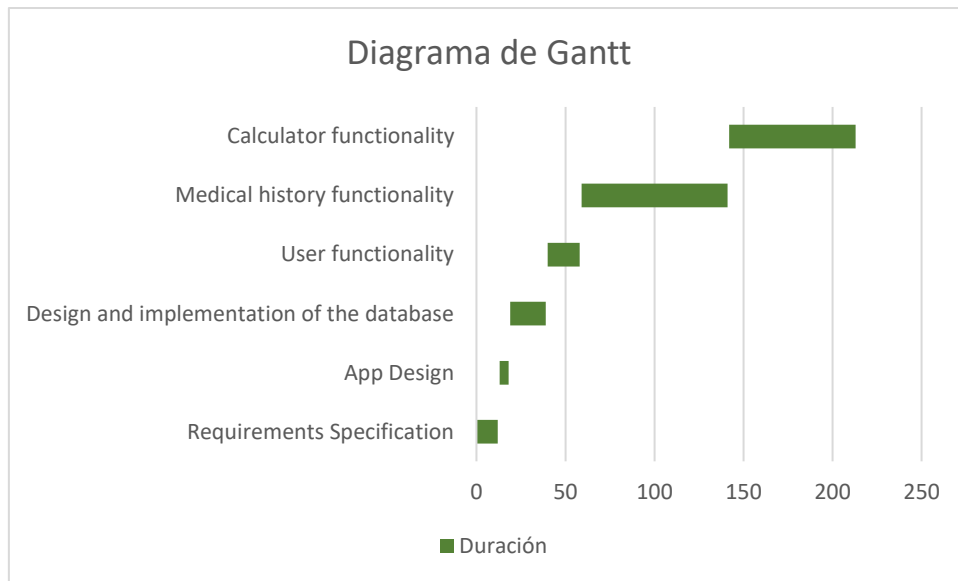


Figura 1- Diagrama de Gantt

Chapter 1 Introduction

In this first chapter of the report, the motivation for carrying out this project will be explained, the objectives to be achieved, how the report is structured and a Gantt's chart will be shown with the actions carried out.

1.1 Motivation

In the United States, prostate cancer is the most common cancer in men after skin cancer. About one in eight men will be diagnosed with prostate cancer in his lifetime.

Prostate cancer is the second leading cause of death in men in the United States, after lung cancer. About one in 41 men will die from prostate cancer.

But prostate cancer does not follow the evolutionary parameters of any other type of cancer. In this sense, a diagnosis of prostate cancer with favorable factors does not have to compromise the life of a patient. The cancer can affect the patient less aggressively. That is why it is important not only to diagnose the cases, but also to diagnose those aggressive cases that can end the lives of patients in the short term.

For these reasons it is very important to detect clinically significant prostate cancer, as opposed to indolent cancer, as soon as possible and to recommend the best possible treatment for the patient. The application will help doctors to be able to recommend solutions in terms of diagnosis and the most appropriate treatment for patients thanks to the data model that can be created.

1.2 Objectives

The main objective of this work is to implement an application that allows physicians to enter the file of a patient suspected prostate cancer, once diagnosed, have the follow-up of patient visits and their files and receive a recommendation of greater or lesser aggressiveness a possible depending on what the model is capable of predicting regarding the phase in which the patient is.

The specific objectives are:

- Develop an application that allows in the most intuitive way possible to enter a patient file in the application.
- Use a calculator that recommends whether a patient should have a prostate biopsy based on their symptoms or, if it has already been done, tell what grade the tumor is in and what possible treatments are recommended.
- Implement a REST API to facilitate access to the services offered by the application.

1.3 Memory structure

In this paragraph it will be explained the structure of the memory.

- **Chapter 1 Introduction:** This chapter describes the motivation, objectives, structure of the report and show a Gantt's chart of the project.

- **Chapter 2 State of art:** This chapter shows examples of websites where the risk of prostate cancer can be calculated.
- **Chapter 3 Technologies used:** This chapter describes the technology used to implement the application.
- **Chapter 4 Requirements Specification:** This chapter describes the actors and use cases.
- **Chapter 5 Architecture and data model:** This chapter describes the application architecture and data model.
- **Chapter 6 Design:** This chapter will explain the design and implementation of the main functionalities of the application.
- **Chapter 7 REST API development and documentation:** This chapter explains the different endpoints used in the REST API.
- **Chapter 8 Conclusions and future work:** This chapter presents the conclusions of the project and explains the extensions of the application in the future.
- **Chapter 9 Individual Contributions:** This chapter explains the contributions of each person in the Project.
- **Annex I:** User's manual.

1.4 Gantt's Chart

In this paragraph it will be shown a Gantt's chart (Figura 2) with the actions made during this project.

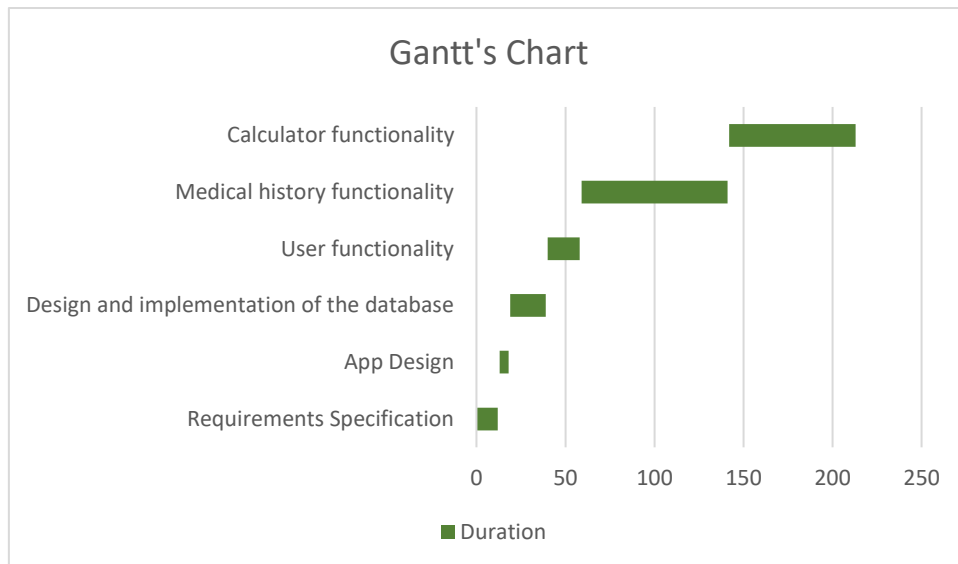


Figura 2 - Gantt`s diagram

Capítulo 2 Estado del Arte

En el siguiente capítulo se describen algunas webs en las cuales también se usan calculadoras para calcular el riesgo de cáncer de próstata utilizando el valor PSA. La mayoría de estas aplicaciones simplemente sirven como calculadoras y no tienen la opción de guardar y gestionar los historiales de los pacientes:

- **Prostate Cancer Prevention TrialRisk Calculator Version 2.0 [1]:** Esta web utiliza varios datos para calcular la probabilidad de cáncer de próstata, lo más importantes son: el nivel del PSA, el resultado del tacto rectal y si se ha realizado una biopsia y su resultado. En la Figura 3 podemos ver como se nos presenta la información.

El riesgo de cáncer de la próstata si se procediera con una biopsia

En base a los factores de riesgo proveídos, proceder con una biopsia de la próstata conllevaría:

-  un 12% de probabilidad de cáncer de la próstata de alto grado,
 -  un 22% de probabilidad de cáncer de la próstata de bajo grado,
 -  un 66% de probabilidad de que la biopsia no demuestre cáncer
-  Aproximadamente de un 2 a 4% de hombres que se someten a una biopsia sufrirán de una infección que puede que requiera hospitalización.

Por favor, consulte con su médico acerca de estos resultados.



Figura 3 - Resultado de Prostate Cancer Prevention TrialRisk Calculator

- **Prostate Cancer Research Foundation, Rotterdam (SWOP) [2]:** Esta web es una iniciativa del Centro Médico y Universitario de Rotterdam, con el objetivo de apoyar financieramente proyectos en el campo del cáncer de próstata. Esta web cuenta con muchas calculadoras diferentes dependiendo de las necesidades del usuario. La que más nos interesa es la “Risk Calculator 2” (Figura 4) que usa el nivel del PSA.

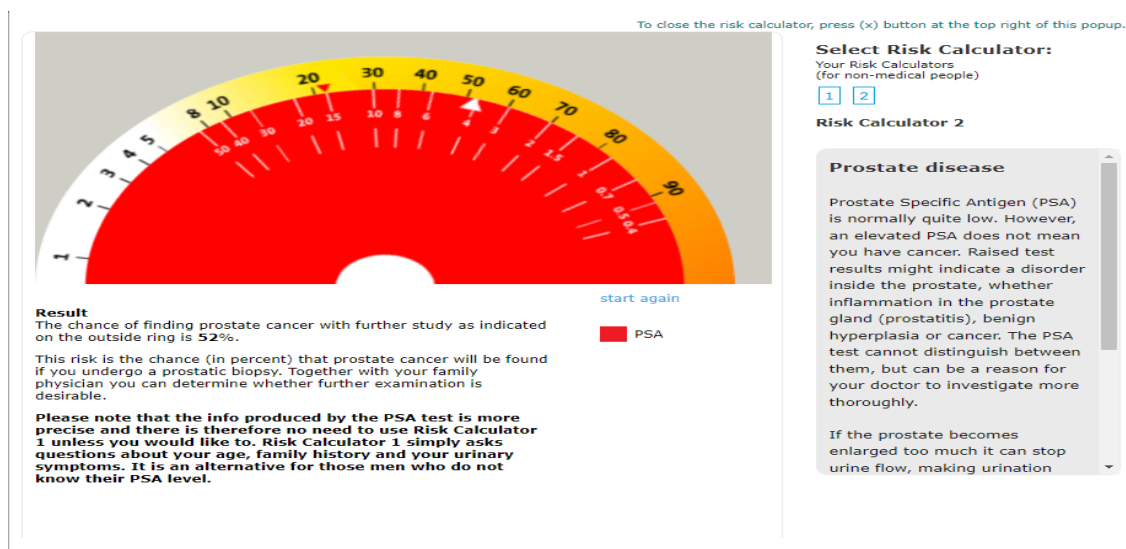


Figura 4 - Resultado de la calculadora de SWOP

- **Memorial Sloan Kettering Cancer Center** [3]: Memorial Sloan Kettering es líder en el cuidado de pacientes e investigación y su web cuenta con calculadoras muy avanzadas en la predicción de cáncer de próstata, Una de ellas es su calculadora de riesgo de cáncer de próstata de alto grado con biopsia prostática (Figura 5), que usa el nivel del PSA, el tacto rectal y el resultado de la biopsia para dar un resultado.

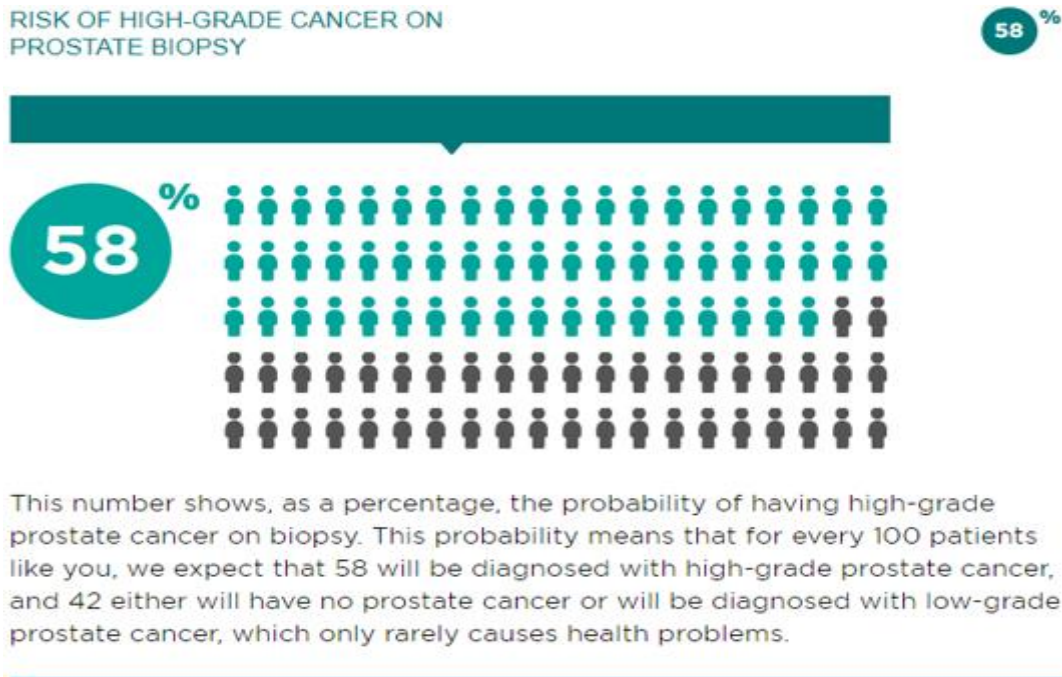


Figura 5 - Resultado de la calculadora del Memorial Sloan Kettering Center

- **ProstateRisk.ca** [4]: prostaterisk.ca fue creado por Robert Nam, para poder mejorar la detección del cáncer de próstata, su calculadora utiliza, aparte del PSA y el tacto rectal, la edad y la etnia del paciente. En la Figura 6 podemos ver el resultado de la calculadora.

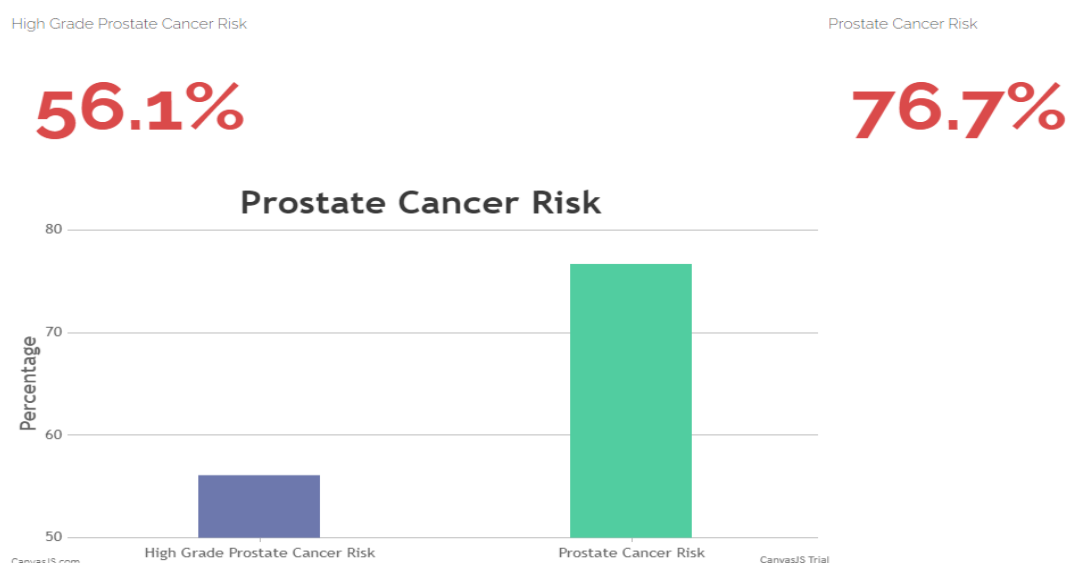


Figura 6 - Resultado de la calculadora de prostaterisk.c

Capítulo 3 Tecnologías empleadas

En este capítulo se van a comentar las tecnologías empleadas en el desarrollo del proyecto.

3.1 React

React [1] es una librería front-end de JavaScript [2] para desarrollar aplicaciones web. Tiene como finalidad la creación de componentes interactivos para interfaces de usuarios.

Su principal característica es el componente. Las aplicaciones en React están compuestas por componentes que funcionan como piezas con las que el usuario puede interactuar, los cuales se pueden combinar hasta crear una web completa.

3.2 Django

Django [4] es un framework para Python diseñado para poder realizar cualquier tipo de aplicación, permitiendo que el desarrollo sea escalable y de gran calidad.

La principal característica de Django es su ORM [8] el cual permite escribir consultas de SQL para crear tablas y gestionar los datos sin la necesidad de tener que aprender el lenguaje. Es muy fácil de usar e incluye las principales consultas SQL.

3.3 XAMPP

XAMPP [5] es un paquete de software libre, que consiste principalmente en el sistema de gestión de bases de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script PHP y Perl.

XAMPP simplemente es un paquete que contiene el gestor de bases datos que utiliza como cliente de MySQL a phpMyAdmin, pero te permite gestionar complementemente bases de datos MySQL, ya sea crear, editar y consultar de una amena rápida y sencilla.

3.4 Heroku

Heroku [6] es una plataforma de servicios en la nube que permite manejar los servidores y sus configuraciones, escalamiento y la administración. Una ventaja de heroku es que puede soportar diferentes lenguajes de programación.

Heroku utiliza Git como conexión entre el desarrollo local y la nube por lo que resulta muy fácil mantener actualizada la aplicación ya que simplemente se necesitan los comandos típicos de GIT para ello.

3.5 GIT

Git [7] es un software que sirve para el control de versiones, permitiendo que varias personas puedan trabajar sobre el mismo código en paralelo.

Algunas de sus principales ventajas son permitir que se puedan comparar, fusionar o recuperar diferentes versiones de un código y su estructura en árbol permitiendo que se pueda trabajar en diferentes ramas de un proyecto sin cambiar el código original.

3.6 Python

Python [12] es un lenguaje de programación de tipado dinámico y programación multiparadigmática, lo que significa que puedes usar Python para la programación orientada a objetos, programación imperativa y la programación funcional. El principal objetivo de Python es priorizar el desarrollo de un código fácilmente legible.

Python se ha convertido en un lenguaje muy popular a la hora de trabajar en proyectos con IA y machine learning debido a la gran variedad de librerías que existen y lo fácilmente legible que resulta programar en este lenguaje.

Capítulo 4 Especificación de requisitos

En este capítulo se va a explicar la especificación de requisitos para el trabajo realizado.

4.1 Actores

Debido al tipo de aplicación que hemos desarrollado, la cual tiene como objetivo ser utilizada en la fase inicial de su existencia por el doctor Víctor Manuel Carrero López, por lo que solo existe un único actor:

Usuario registrado: el usuario será el Urólogo que tiene acceso a la aplicación y a todas sus funcionalidades.

4.2 Casos de uso

A continuación, se van a presentar los casos de uso definidos en la aplicación, para ello se han dividido en dos módulos funcionales:

- **Módulo usuario.**
- **Módulo historiales médicos.**

4.2.1 Modulo Usuario

En este módulo se describen las diferentes funciones que tienen los usuarios para tratar su información personal. En la Figura 5 se observa el diagrama de casos de uso del módulo usuario.

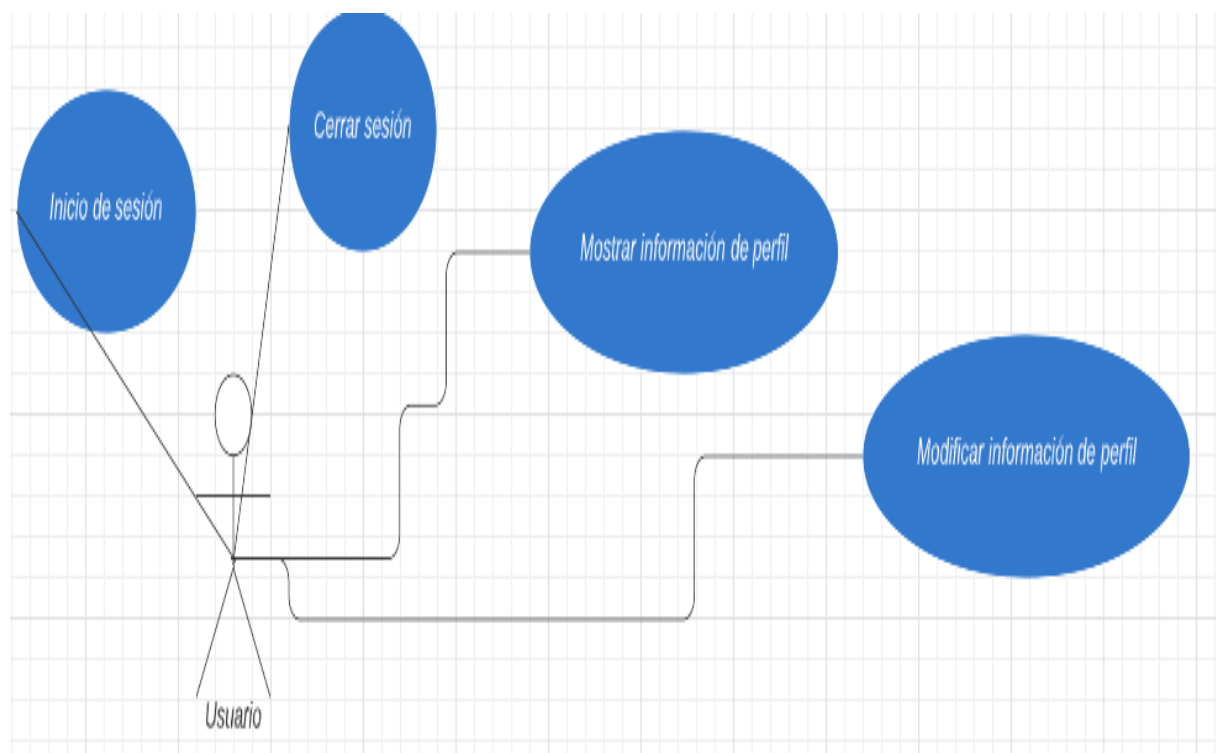


Figura 7 - Diagrama de casos de uso módulo usuario

	Inicio de sesión	
Versión	1.0.0	
Prioridad	Alta	
Estabilidad	Alta	
Precondición	El usuario ha de estar registrado	
Descripción	Los usuario que ya estén registrados en la aplicación podrán acceder a ella.	
Entrada	Correo y contraseña	
Salida	Nombre, primer apellido, segundo apellido y email.	
Secuencia normal	Paso	Acción
	1	El usuario accede a la aplicación
	2	El usuario introduce sus credenciales y pulsa el botón de inicio
	3	El sistema comprueba las credenciales
	4	El sistema genera un token para el usuario
Postcondición	NA	
Excepciones	Paso	Acción
	3	El inicio de sesión no se ha podido completar, las credenciales no eran correctas
Comentarios	NA	

Tabla 1 - Inicio de sesión

	Cerrar sesión	
Versión	1.0.0	
Prioridad	Alta	
Estabilidad	Alta	
Precondición	El usuario ha de estar registrado y con la sesión iniciada	
Descripción	Los usuario podrán cerrar la sesión de su perfil	
Entrada	Token de autenticación	
Salida	NA	
Secuencia normal	Paso	Acción
	1	El usuario accede la aplicación
	2	El usuario pulsará el botón de cerrar sesión en el header
	3	El sistema elimina los token asociados a esa cuenta
	4	El sistema devuelve al usuario a la pantalla de registro
Postcondición	NA	
Excepciones	Paso	Acción
	3	El usuario no tiene un token de sesión, se informa del error
Comentarios	NA	

Tabla 2 - Cerrar sesión

	Mostrar perfil de usuario	
Versión	1.0.0	
Prioridad	Alta	
Estabilidad	Alta	
Precondición	El usuario ha de estar registrado y con la sesión iniciada	
Descripción	Los usuario podrán acceder a la información de su perfil	
Entrada	Id del usuario	
Salida	NA	
Secuencia normal	Paso	Acción
	1	El usuario accede la aplicación
	2	El usuario pulsará so nombre en el header
	3	El sistema elimina mostrará la información del usuario
Postcondición	NA	
Excepciones	Paso	Acción
	3	El id no corresponde con el del usuario, se informará del error
Comentarios	NA	

Tabla 3 - Mostrar perfil de usuario

	Modificar información del perfil	
Versión	1.0.0	
Prioridad	Alta	
Estabilidad	Alta	
Precondición	El usuario ha de estar registrado y con la sesión iniciada	
Descripción	El usuario puede modificar los datos de su perfil.	
Entrada	Id del usuario	
Salida	NA	
Secuencia normal	Paso	Acción
	1	El usuario accede a la web de la aplicación
	2	El usuario hace pulsa sobre su nombre en el header
	3	El sistema devuelve la información de usuario a partir de su id
	4	El usuario pulsa el botón de modificar datos
	5	El usuario escribe los nuevos datos y pulsa el botón de actualizar
	6	El sistema comprueba que los datos introducidos son correctos
	7	El sistema devuelve los datos actualizados
Postcondición	NA	
Excepciones	Paso	Acción
	6	Los datos introducidos no son correctos, se informará del error
Comentarios	NA	

Tabla 4 - Modificar información del perfil

4.2.2 Modulo Historiales Médicos

En este módulo se describen todas las funciones que tienen que ver con los historiales médicos o expedientes. En la Figura 6 se observa el diagrama de casos de uso del módulo historiales médicos.

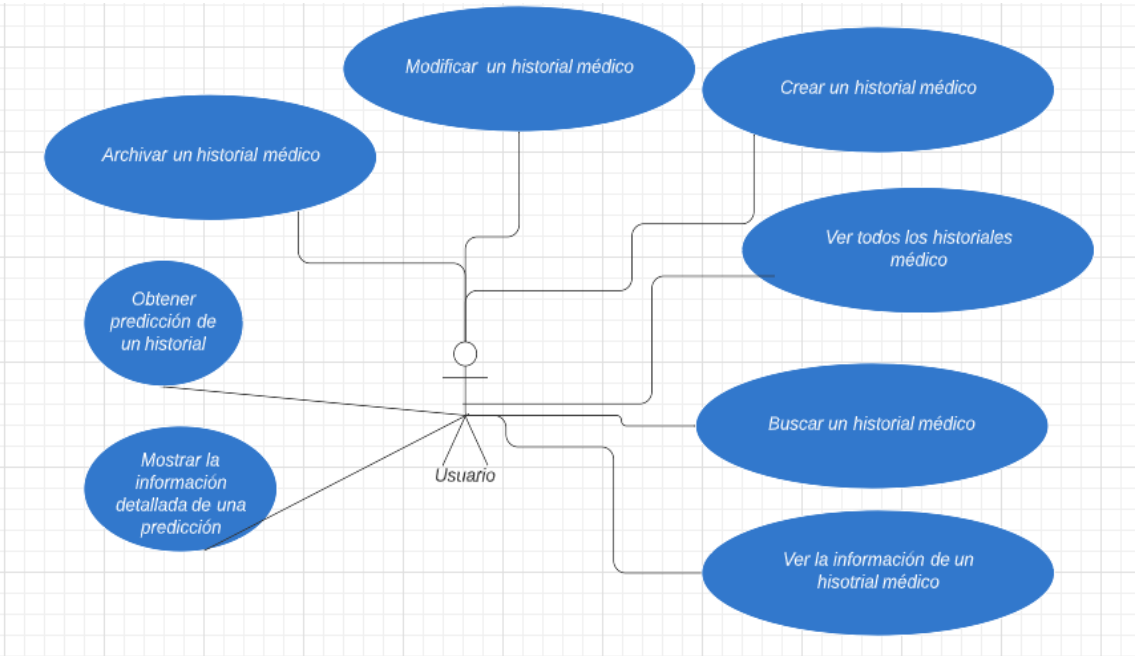


Figura 8 - Diagrama de casos de uso módulo historiales médicos

Ver todos los historiales médicos		
Versión	1.0.0	
Prioridad	Alta	
Estabilidad	Alta	
Precondición	El usuario ha de estar registrado y con la sesión iniciada	
Descripción	Los usuarios podrán ver todos los historiales en la aplicación	
Entrada	NA	
Salida	Lista de historiales médicos de la aplicación	
Secuencia normal	Paso	Acción
	1	El usuario accede la aplicación
	2	El usuario pulsará el botón de expedientes en el header
	3	El sistema muestra la lista de historiales médicos
Postcondición	NA	
Excepciones	NA	
Comentarios	NA	

Tabla 5 - Ver todos los historiales médicos

	Buscar un historial médico	
Versión	1.0.0	
Prioridad	Media	
Estabilidad	Alta	
Precondición	El usuario ha de estar registrado y con la sesión iniciada	
Descripción	Los usuarios podrán buscar un historial por su número de historial	
Entrada	El número del historial	
Salida	Historial médico buscado	
Secuencia normal	Paso	Acción
	1	El usuario accede la aplicación
	2	El usuario pulsará el botón de expedientes en el header
	3	El sistema muestra la lista de historiales médicos
	4	El usuario escribe el número de historial en la barra de búsqueda y pulsa el botón buscar
	5	El sistema comprueba que el número de historial es válido
	6	Se muestra el historial buscado
Postcondición	NA	
Excepciones	Paso	Acción
	5	El número de historial no es válido
Comentarios	NA	

Tabla 6 - Buscar historial médico

	Ver la información detallada de un historial	
Versión	1.0.0	
Prioridad	Alta	
Estabilidad	Alta	
Precondición	El usuario ha de estar registrado y con la sesión iniciada	
Descripción	El usuario puede pulsar la tarjeta de un historial para ver la información detallada de un historial	
Entrada	El número del historial	
Salida	La información detallada del historial médico	
Secuencia normal	Paso	Acción
	1	El usuario accede la aplicación
	2	El usuario pulsará el botón de expedientes en el header
	3	El sistema muestra la lista de historiales médicos
	4	El usuario pulsará sobre el historial que quiera para ver la información detallada
	5	El sistema muestra la información detallada del historial
Postcondición	NA	
Excepciones	NA	
Comentarios	El sistema mostrará la información de la última visita por defecto.	

Tabla 7 - Ver información detallada de un historial

Modificar un historial médico		
Versión	1.0.0	
Prioridad	Media	
Estabilidad	Alta	
Precondición	El usuario ha de estar registrado y con la sesión iniciada	
Descripción	El usuario puede modificar la información de un historial médico	
Entrada	El número del historial, la información actualizada del historial	
Salida	La información detallada del historial médico actualizada	
Secuencia normal	Paso	Acción
	1	El usuario accede la aplicación
	2	El usuario pulsará el botón de expedientes en el header
	3	El sistema muestra la lista de historiales médicos
	4	El usuario pulsará sobre el historial que quiera para ver la información detallada
	5	El sistema muestra la información detallada del historial
	6	El usuario pulsará el botón de modificar la información del historial
	7	El usuario modificará la información del historial y pulsará el botón de enviar formulario
	8	El sistema comprobará que la información modificada es correcta y actualizará el historial
	9	El sistema mostrará la información del historial actualizada
Postcondición	NA	
Excepciones	Paso	Acción
	8	La información modificada no es correcta, el historial no se actualizará
Comentarios	NA	

Tabla 8 - Modificar un historial médico

Archivar un historial		
Versión	1.0.0	
Prioridad	Media	
Estabilidad	Media	
Precondición	El usuario ha de estar registrado y con la sesión iniciada	
Descripción	El usuario puede archivar un historial para que no se vuelva a mostrar en la aplicación	
Entrada	El número del historial	
Salida	NA	
Secuencia normal	Paso	Acción
	1	El usuario accede la aplicación
	2	El usuario pulsará el botón de expedientes en el header
	3	El sistema muestra la lista de historiales médicos
	4	El usuario pulsará sobre el historial que quiera archivar
	5	El sistema muestra la información detallada del historial
	6	El usuario pulsará el botón de archivar historial
	7	El sistema archivará el archivo
Postcondición	NA	
Excepciones	NA	
Comentarios	NA	

Tabla 9 - Archivar un historial médico

Crear nuevo historial		
Versión	1.0.0	
Prioridad	Alta	
Estabilidad	Alta	
Precondición	El usuario ha de estar registrado y con la sesión iniciada	
Descripción	El usuario puede crear un nuevo historial médico con los datos del paciente	
Entrada	Los datos del historial médico, el id del usuario	
Salida	NA	
Secuencia normal	Paso	Acción
	1	El usuario accede la aplicación
	2	El usuario pulsará el botón de nuevo expediente en el header
	3	El sistema muestra la vista de creación de nuevo expediente
	4	El usuario rellenará la información del expediente y pulsará en el botón de enviar
	5	El sistema muestra comprueba la que información del expediente es correcta
	6	El sistema muestra un mensaje indicando que el expediente se ha creado
Postcondición	NA	
Excepciones	Paso	Acción
	5	La información del historial es incorrecta, se informará del error
Comentarios	NA	

Tabla 10 - Crear nuevo historial

Obtener predicción de un historial		
Versión	1.0.0	
Prioridad	Alta	
Estabilidad	Alta	
Precondición	El usuario ha de estar registrado y con la sesión iniciada	
Descripción	El usuario obtener la predicción en base a la información de un historial	
Entrada	El historial del que se quiere obtener la predicción	
Salida	Información de la predicción	
Secuencia normal	Paso	Acción
	1	El usuario accede la aplicación
	2	El usuario pulsará el botón de obtener predicción en el header
	3	El usuario escribirá un número de historial en la barra de búsqueda y pulsará buscar
	4	El sistema comprobará que el número de historial existe y devolverá la tarjeta del historial
	5	El usuario pulsará el botón de obtener predicción
	6	El sistema calculará la predicción en base la información del historial médico
Postcondición	NA	
Excepciones	Paso	Acción
	4	El número del historial no existe, no se devolverá nada
	6	Puede que algún dato sea incorrecto en el historial y no se realice la predicción
Comentarios	NA	

Tabla 11 - Obtener predicción de un historial

Obtener información detallada de un historial		
Versión	1.0.0	
Prioridad	Alta	
Estabilidad	Alta	
Precondición	El usuario ha de estar registrado, con la sesión iniciada y se ha tenido que realizar la predicción	
Descripción	El usuario puede acceder a la información detallada de una predicción	
Entrada	NA	
Salida	Información detallada de la predicción	
Secuencia normal	Paso	Acción
	1	El usuario accede la aplicación
	2	El usuario pulsará el botón de obtener predicción en el header
	3	El usuario escribirá un número de historial en la barra de búsqueda y pulsará buscar
	4	El sistema comprobará que el número de historial existe y devolverá la tarjeta del historial
	5	El usuario pulsará el botón de obtener predicción
	6	El sistema calculará la predicción en base la información del historial médico
	7	El usuario pulsará los botones debajo de la predicción para obtener información detallada
Postcondición	NA	
Excepciones	Paso	Acción
	4	El número del historial no existe, no se devolverá nada
	6	Puede que algún dato sea incorrecto en el historial y no se realice la predicción
Comentarios	NA	

Tabla 12 - Obtener información detallada de un historial

Capítulo 5 Arquitectura y modelo de datos

En este capítulo se explicará la estructura y el modelo de datos de este proyecto.

5.1 Arquitectura

Para el desarrollo del proyecto, se aplica el patrón MVC. La aplicación se estructura en tres partes: el modelo, las vistas y los controladores.

- **Las vistas:** Son las clases .js del proyecto y en ellas se muestra la información de la aplicación al usuario. Reacciona sin necesidad de recargar la página si hay algún cambio.
- **Controlador:** Son los ficheros views.py dentro de los módulos de la aplicación los que se encarga de conectar la información enviada desde las vistas con el modelo de datos.
- **Modelo:** Es el encargado de obtener la información de la base de datos. Como Django funciona por ORM no es necesario tener peticiones específicas a la base de datos, sino que puedes usar instrucciones que simulen las peticiones MySQL. Estas instrucciones se realizan en los ficheros views.py.



Figura 9- Diagrama del patrón MVC

En la Figura 10 podemos ver un esquema de la estructura del patrón MVC.

5.2 Modelo de datos

En este capítulo se va a describir el modelo de datos utilizado para el trabajo realizado.

Para esta aplicación únicamente es necesario almacenar información de dos entidades principales: el usuario y el historial médico de la visita del paciente. La toda la información relacionada con el historial de un paciente estará almacenada en una sola entidad.

El sistema además contendrá usuarios registrados. Un usuario tiene un nombre, un primer apellido, un segundo apellido, un email y una contraseña. Cada usuario podrá crear varios historiales y cada historial será creado por un único usuario.

La base de datos de este proyecto la hemos creado usando el ORM de Django. Este ORM te permite crear las tablas de las bases de datos mediante instrucciones sin la necesidad de usar

lenguaje MySQL, eso significa que hemos podido definir las variables de la tabla, las restricciones, el tipo de dato, longitud, primary key... y las relaciones con otras tablas de la base de datos. En la figura 10 podemos ver la definición de la entidad usuario.

```
class UserProfile(AbstractBaseUser):
    """ Modelo Base de Datos para Usuarios en el Sistema """
    email = models.EmailField(max_length=255, unique=True)
    name = models.CharField(max_length=255)
    first_name = models.CharField(max_length=255)
    last_name = models.CharField(max_length=255)
    is_active = models.BooleanField(default=True)

    objects = UserProfileManager()

    USERNAME_FIELD = 'email'
    REQUIRED_FIELDS = ['name']

    def get_name(self):
        return self.name

    def get_first_name(self):
        return self.first_name

    def get_last_name(self):
        return self.last_name

    def __str__(self):
        return self.email

class Profile(models.Model):
    user_profile = models.ForeignKey(
        settings.AUTH_USER_MODEL,
        on_delete=models.CASCADE
    )
    status_text = models.CharField(max_length=255)
    created_on = models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return self.status_text
```

Figura 10 - Models.py del módulo users

La base de datos de la aplicación es muy simple, ya que solo está formada por dos tablas: La tabla “medicalhistorys_medicalhistory” que contiene toda la información relacionada con los historiales médicos y la tabla de “users_userprofile” que contienen toda la información relacionada con los usuarios. A continuación, se describen la información que almacenan ambas tablas.

Ambas tablas están relacionadas con una relación 1:N, un usuario puede estar relacionado con varios historiales médicos y un historial médico solamente puede estar relacionado con un usuario.

Tabla users_userprofile:

Lista de parámetros de la tabla users_profile:

- Id: Es el identificador del usuario.

- password: Representa la contraseña del usuario.
- last_login: Representa la fecha de la última conexión.
- Email: Representa el email del usuario.
- Name: Representa el nombre completo del usuario.
- First_name: Representa el nombre del usuario.
- Last_name: Representa el apellido del usuario.
- Is_active: Comprueba si el usuario está conectado en este momento.

Tabla medicalhistorys_medicalhistory:

Lista de parámetros de la tabla medicalhistorys_medicalhistory:

- Id: Representa el identificador del historial.
- Nhist: Representa el número de historial.
- isActive: Representa si el historial está activo.
- fechaVisita: Representa la fecha de la visita del paciente.
- fechaCir: Representa la fecha en la que se realizará la cirugía.
- Edad: representa la edad del paciente.
- Etnia: Representa la etnia del paciente.
- Obeso: representa el IMC del paciente.
- Hta: Representa la hipertensión arterial.
- Dm: Representa la diabetes mellitus.
- Tabaco: Representa el número de cigarrillos que el paciente consume al día
- Hereda: representa si el paciente ha heredado el cáncer
- VasectomíaPrevia: representa si el paciente se ha realizado una vasectomía.
- usoDeFarmacosInhibidoresDe5AlfaReductasa: representa el uso de fármacos del paciente.
- densidadDelPSA: representa la densidad del psa.
- VelocidadDelPSA: representa la velocidad del psa.
- tratamiento = representa el posible tratamiento del paciente.
- gleasonPiezaPRL = representa el Gleason obtenido tras la prostatectomía.
- bilat2 =representa la bilateralidad tras prostatectomía.
- tHistologico = representa le tipos histológico del cáncer.
- gleasonEnBiopsia = representa el valor de Gleason tras biopsia.
- localiz = representa la localización del cáncer.
- multifoc = representa la multifocalidad del cáncer.
- volumen = representa el volumen del cáncer.
- extracap = representa la extensión extracapsular del cáncer.
- vvss = representa la invasión de las vesículas seminales.
- ilinf2 = infiltración de la vesícula linfática tras prostatectomía.
- ivascu2 = infiltración vascular tras prostatectomía.
- pinag = representa el pin de alto grado.
- margen = representa los márgenes quirúrgicos positivos.
- tnm2 = representa el valor TNM tras prostatectomía.

- psapos = representa el valor psa postoperatorio.
- tduplit = representa el tiempo de duplicación del psa.
- numVisita = representa el número de visita del paciente.
- gleasonPrimarioBiopsiaProstática = representa el valor del Gleason primario en la biopsia.
- gleasonSecundarioBiopsiaProstática = representa el valor del Gleason secundario en la biopsia.
- gleasonPrimarioTrasProctatectomía = representa el valor del Gleason primario tras la prostatectomía.
- gleasonSecundarioTrasProctatectomía = representa el valor del Gleason secundario tras la prostatectomía.
- ctnmBiopsia = representa el valor del cTNM tras la biopsia.
- tactor = representa el resultado del tacto rectal.
- psapre = representa el valor del psa preoperatorio.
- psa = representa el valor del psa total.
- psaStratos = representa el valor del psa en estratos.
- psalt = representa el valor del psalt.
- tdupree = tiempo que tardará el psa en duplicarse.
- estadiajeRm = representa el estadio obtenido en la resonancia magnética.
- diametroMaximoLesionRm = representa el diámetro máximo de la lesión obtenido en la resonancia magnética.
- isupBiopsiaRmDirigida = representa el valor del isup en la resonancia magnética.
- hallazgosRadiologicos1 = representa el valor de la pi-rads pre-biopsia.
- hallazgosRadiologicos2 = representa el valor de la pi-rads tras una biopsia.
- nbipsia = representa el número de la biopsia
- bilat = representa la bilateralidad del cáncer tras biopsia.
- iperin = representa la infiltración peri neural tras biopsia
- iperin2 = representa la filtración peri neural tras prostatectomía.
- ilinf = representa la infiltración linfática tras biopsia.
- ivascu = representa la infiltración vascular tras biopsia.
- cPositivoBiopsiaDirigida = % de cores positivos tras realizar la biopsia.
- afectacionCores = % de afectación de cada core positivo.
- isup = representa el valor del isup.
- t1mtx = representa el tiempo en meses hasta la primera metástasis.
- fechaFin = representa la fecha de la última revisión.
- tsegu = representa el tiempo de seguimiento desde PRL en meses.
- psafin = representa el valor del último psa en seguimiento.

Capítulo 6 Diseño

En este capítulo se van a describir los principales aspectos acerca del diseño y funcionalidad realizados para el desarrollo de la aplicación:

6.1 Colores

En este apartado se explicará la selección de colores que se ha utilizado en el proyecto.

Para la aplicación se ha querido utilizar el mínimo número de colores posibles para que el resultado fuera el más sencillo y agradable posible, así pues, como base se ha utilizado el color azul claro y como contraste sea utilizado el blanco. También se ha utilizado el color morado para resaltar diferentes partes de la aplicación y el color verde para realizar las tarjetas que representan los historiales de visita de los pacientes.

6.2 Funcionalidades de la aplicación

En este apartado se describen las principales funcionalidades implementadas.

6.2.1 Crear historial de una visita

Para poder crear un historial de una visita el usuario deberá escribir el valor de las variables de cada sección del historial pudiendo hacerlo en el orden que se quiera.

Un historial está dividido en 8 secciones distintas:

- Filiación
- Sociodemográficas
- Antecedentes médicos
- Parámetros Clínico-Patológicas
- Biopsia Prostáticas
- Tras Prostatectomía
- Evolutivos
- Comentario adicional

En la Figura 11 podemos ver la vista general del formulario para crear un nuevo historial con las distintas secciones a la izquierda, por las que podemos navegar clicando en ellas.

The image shows a web application interface for creating a new medical history. On the left, there is a vertical sidebar with a list of sections: FILIACIÓN, SOCIODEMOGRÁFICAS, ANTECEDENTES, CLÍNICO-PATOLÓGICAS, BIOPSIA PROSTÁTICAS, TRAS PROSTATECTOMÍA, EVOLUTIVOS, and ENVIAR FORMULARIO. The 'FILIACIÓN' section is currently selected and highlighted in blue. To the right of the sidebar is a form titled 'Filiación' with four input fields: 'NHS', 'FECHACIR', 'Nº DE VISITA', and 'FECHA DE VISITA'. Each field has a date format placeholder (dd/mm/aaaa) and a calendar icon. The form is displayed on a white background with a blue header and footer.

Figura 11 - Vista general de la función crear nuevo historial

Para poder navegar entre las diferentes secciones de manera dinámica y que se mantengan el valor de las variables se ha optado por utilizar la librería TabPanel de React. En la Figura 13 podemos verlo.

```
<TabPanel className="tabsPanels" value={value} index={0} style={{ width: "100%", margin: "5%" }} >
  <Form_Filiacion handleChanges={handleChanges} parameter={parameters}></Form_Filiacion>
</TabPanel>
<TabPanel className="tabsPanels" value={value} index={1} style={{ width: "100%", margin: "5%" }}>
  <Form_Sociodemograficas handleChanges={handleChanges} parameter={parameters}></Form_Sociodemograficas>
</TabPanel>
<TabPanel className="tabsPanels" value={value} index={2} style={{ width: "100%", margin: "5%" }}>
  <Form_Antecedentes handleChanges={handleChanges} parameter={parameters}></Form_Antecedentes>
</TabPanel>
<TabPanel className="tabsPanels" value={value} index={3} style={{ width: "100%", margin: "5%" }}>
  <Form_Clinico_Patologias handleChanges={handleChanges} parameter={parameters}></Form_Clinico_Patologias>
</TabPanel>
<TabPanel className="tabsPanels" value={value} index={4} style={{ width: "100%", margin: "5%" }}>
  <Form_Biopsia_Prostaticas handleChanges={handleChanges} parameter={parameters}></Form_Biopsia_Prostaticas>
</TabPanel>
<TabPanel className="tabsPanels" value={value} index={5} style={{ width: "100%", margin: "5%" }}>
  <Form_Tras_Proctatectomia handleChanges={handleChanges} parameter={parameters}></Form_Tras_Proctatectomia>
</TabPanel>
<TabPanel className="tabsPanels" value={value} index={6} style={{ width: "100%", margin: "5%" }}>
  <Form_Evolutivos handleChanges={handleChanges} parameter={parameters}></Form_Evolutivos>
</TabPanel>
<TabPanel className="tabsPanels" value={value} index={7} style={{ width: "100%", margin: "5%" }}>
  <Form_Submit medicalHistory={parameters} user={props} handleChanges={handleChanges}></Form_Submit>
</TabPanel>
```

Figura 12 - TabPanel

Cada sección del historial representa un Tab diferente, el cual es llamado en la clase madre para visualizarlo. En la Figura 13 podemos ver el código que agrupa todos los Tabs para mostrarlo por pantalla.

```
<Tabs
  id="tabsVerticales"
  orientation="vertical"
  variant="scrollable"
  value={value}
  onChange={handleChange}
  className={classes.tabs}
  style={{ width: "15%" }}
>
  <Tab label="Filiación" {...a11yProps(0)} />
  <Tab label="Sociodemográficas" {...a11yProps(1)} />
  <Tab label="Antecedentes" {...a11yProps(2)} />
  <Tab label="Clínico-Patológicas" {...a11yProps(3)} />
  <Tab label="Biopsia Prostáticas" {...a11yProps(4)} />
  <Tab label="Tras Prostatectomía" {...a11yProps(5)} />
  <Tab label="Evolutivos" {...a11yProps(6)} />
  <Tab label="Enviar Formulario" style={{ backgroundColor: "#1C8EF9" }} {...a11yProps(7)} />
</Tabs>
```

Figura 13 - Todas las Tabs

Una vez se han rellenado todos los campos que nos interesaba podemos hacer que se inserte el historial en la base de datos pulsando el botón de enviar en la sección de enviar formulario.

El historial es enviado al completo junto con el id del doctor a la parte back de nuestra aplicación mediante la función en la Figura 16.

```

const onSubmit = async (e) => {
  e.preventDefault();
  //-----Setemos los valores Genericos-----//
  props.medicalHistory.idDoctor = props.user.user.id
  props.medicalHistory.isActive = true
  handleClose()
  await axios.post(`http://localhost:8000/medicalHistorys/newMedicalHistory`,
    {medicalHistory: props.medicalHistory},
  ).then(response =>{
    if(response.status === 201){
      history.push("/")
      history.push(`medicalHistorys`)
    }
  })
  .catch((error) => {
    setError(error.response.data[0])
  });
}

```

Figura 14 - Función para enviar un historial al back

Para gestionar las peticiones a la API se ha usado la librería axios, además para poder navegar a por las distintas urls se ha usado la librería history.

Para que el historial sea aceptado al menos se debe de haber rellenado los campos de número de historial, número de visita y fecha de vista.

Un paciente puede realizar varias vistas al doctor obteniendo resultados diferentes en su historial por lo que nos interesa mantener un registro de las distintas vistas de un paciente para tener la máxima cantidad de datos posibles. Para ello cada visita de un paciente estará representada por el número de historial y el número de visita que tenga. Todas las visitas de un paciente se asociarán con el mismo número de historial. Más adelante veremos cómo navegar entre las diferentes visitas de un paciente. Si hay algún error se le notificaría al usuario y la operación se cancelaría.

En la Figura 17 se muestra la función para crear un nuevo historial.

```

class create_MedicalHistoryPatient(APIView):
    def post(self,request):
        MedicalHistoryData = request.data['medicalHistory']
        if request.method == "POST":
            data = {}
            for attribute in MedicalHistoryData:
                if(MedicalHistoryData.get(attribute) != None):
                    data[attribute] = MedicalHistoryData.get(attribute)
            data['idDoctor'] = UserProfile(MedicalHistoryData.get('idDoctor'))
            if 'nhist' not in data:
                return Response({'El formulario debe tener un numero de Historia del Paciente'}, status = status.HTTP_400_BAD_REQUEST)
            if 'fechaVisita' not in data:
                return Response({'El formulario debe tener una fecha valida'}, status = status.HTTP_400_BAD_REQUEST)
            if 'numVisita' not in data:
                return Response({'El formulario debe tener un numero de visita'}, status = status.HTTP_400_BAD_REQUEST)

            newMedicalHistory = MedicalHistory(**data)
            newMedicalHistory.save()
            return Response({'Historial Medico creado correctamente'}, status = status.HTTP_201_CREATED)

```

Figura 15 - Función para crear un nuevo historial

Una vez comprobado que los datos son correctos, utilizando el ORM de Django, podemos guardar el historial en la base de datos sin la necesidad de usar código SQL,

con la función `newMedicalHistory.save()` se insertaría. Una vez insertado el historial se le notifica al usuario.

6.2.2 Lista de historiales

Esta funcionalidad se creó con la idea de que un usuario pudiera buscar un historial en concreto, además de poder ver los detalles como veremos más adelante. La lista de historiales se muestra mediante “tarjetas”. Cada tarjeta tiene como datos el número de historial, el nombre del doctor que ha creado el historial y la fecha de la última visita. En la Figura 18 podemos ver una lista de historiales.

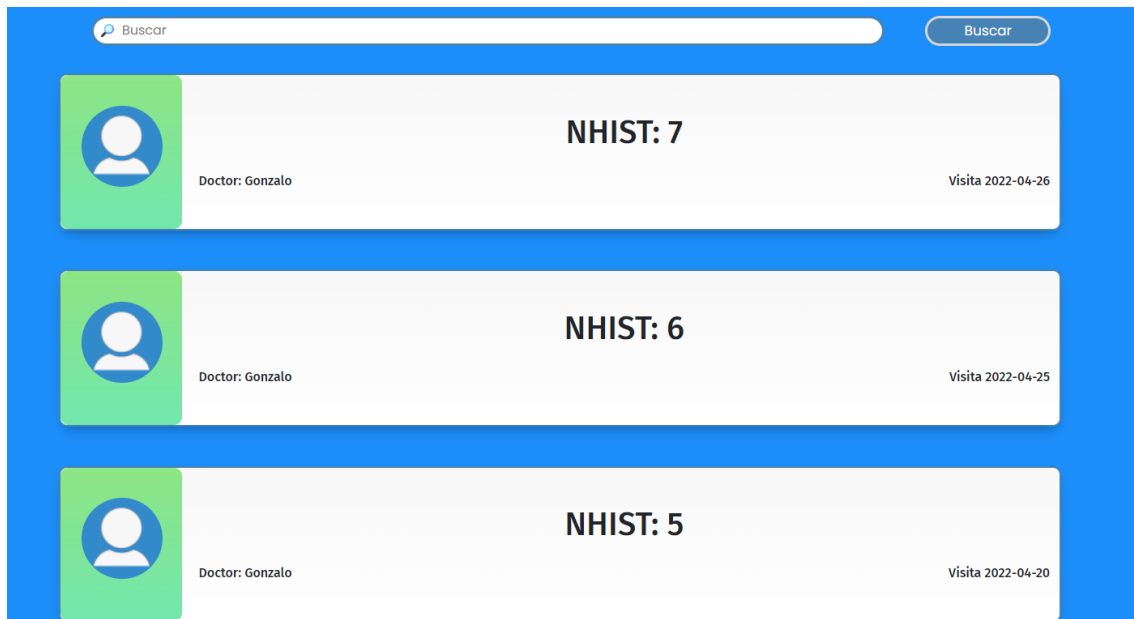


Figura 16 - Imagen de las tarjetas que representan todos los historiales médicos

Para poder buscar un historial el usuario escribe el número de historial en la barra de búsqueda y pulsa el botón de buscar. El número de historial es enviado al back de la aplicación donde se realiza el filtro en la base de datos utilizando instrucciones ORM. En la Figura 19 podemos ver la función para filtrar la base de datos por un número de historial.

```
class getMedicalHistory(APIView):
    def get(self, request, nhist):
        MedicalHistoryId = MedicalHistory.objects.filter(nhist = nhist)
        SerializerMedicalHistorys = serializers.MedicalHistorySerializer(MedicalHistoryId, many=True)
        return Response({
            'medicalHistory' : SerializerMedicalHistorys.data,
            'message': 'Obtencion exitosa del Historial Medico con id =' + str(nhist)
        })
```

Figura 17 - Función GetMedicalHistroy

Al recibir el historial de la base de datos lo recibimos en un formato del que no podemos leer la información. Para poder solucionar este problema primero debemos serializar el objeto recibido a un json, para ello usamos el objeto `serializers`. Este objeto nos permite serializar cualquier objeto, en este caso un historial médico. Para ello le pasamos el historial médico como parámetro y el objeto `MedicalHistorySerializer` se

encargará de convertirlo en un json. En la Figura 19 podemos ver cómo está formado este objeto.

```
class MedicalHistory(models.Model):
    """ Modelo Base de Datos para MedicalHistory en el Sistema """
    idDoctor = models.ForeignKey(UserProfile, on_delete=models.CASCADE)
    nhist = models.IntegerField(null=True)
    isActive = models.BooleanField(default=True)
    fechaVisita = models.DateField(null=True)
    fechaCir = models.DateField(null=True)
    edad = models.IntegerField(null=True)
    etnia = models.CharField(max_length=255, null=True)
    obeso = models.CharField(max_length=255, null=True)
    hta = models.CharField(max_length=255, null=True)
    dm = models.CharField(max_length=255, null=True)
    tabaco = models.CharField(max_length=255, null=True)
    hereda = models.CharField(max_length=255, null=True)
    vasetomiaPrevía = models.CharField(max_length=255, null=True)
    usoDeFarmacosInhibidoresDe5AlfaReductasa = models.CharField(max_length=255, null=True)
    densidadDelPSA = models.CharField(max_length=255, null=True)
    velocidadDelPSA = models.CharField(max_length=255, null=True)
    tratamiento = models.CharField(max_length=255, null=True)
    gleasonPiezaPRL = models.CharField(max_length=255, null=True)
    bilat2 = models.CharField(max_length=255, null=True)
    tHistiológico = models.CharField(max_length=255)
    gleasonEnBiopsia = models.CharField(max_length=255, null=True)
    localiz = models.CharField(max_length=255, null=True)
    multifoc = models.CharField(max_length=255, null=True)
    volumen = models.DecimalField(max_digits=7, decimal_places=4, null=True)
    extracap = models.CharField(max_length=255, null=True)
    vvss = models.CharField(max_length=255, null=True)
    ilinf2 = models.CharField(max_length=255, null=True)
    ivascu2 = models.CharField(max_length=255, null=True)
    pinag = models.CharField(max_length=255, null=True)
    margen = models.CharField(max_length=255, null=True)
    tnm2 = models.CharField(max_length=255, null=True)
    psapos = models.DecimalField(max_digits=7, decimal_places=4, null=True)
    tduplit = models.IntegerField(null=True)
    numVisita = models.IntegerField(null=True)
    gleasonPrimarioBiopsiaProstática = models.IntegerField(null=True)
    gleasonSecundarioBiopsiaProstática = models.IntegerField(null=True)
    gleasonPrimarioTrasProctatectomia = models.IntegerField(null=True)
    gleasonSecundarioTrasProctatectomia = models.IntegerField(null=True)
```

Figura 18 - Modelo para serializar un historial médico

6.2.3 Información detallada de un historial

Si pulsamos en alguna de las tarjetas que representan los historiales médicos se nos mostrará la información completa de ese historial.

En esta vista tendremos la opción de modificar los datos del historial si se desea al pulsar en el botón Actualizar. Al hacer esto todos los campos del historial pasarán a ser modificables gracias al uso de los estados de React. En la Figura 20 podemos ver como al pulsar el botón de actualizar se llama a la función de activateInputs, que la podemos ver en la Figura 21, que vuelve todos los campos modificables.

```

nav className="navProfile">
  <button className="buttonProfile" onClick={activateInputs}>
    Actualizar

```

Figura 19 - Botón Actualizar

```

const activateInputs = () =>{
  |   setDisableInputs(false)
  | }

```

Figura 20 - Función para hacer los campos modificables

Una vez realizados los cambios podemos insertar el nuevo historial en la base de datos al pulsar el botón de Enviar que se encuentra la final de la página. Este insert tendrá las mismas restricciones que al crear un nuevo historial. En la Figura 22 podemos ver la función para enviar los datos al back y en la Figura 23 se muestra la función para actualizar un historial.

```

const onSubmit = () =>{
  axios.post(`http://localhost:8000/medicalHistorys/updateMedicalHistory/${parameters.nhist}`,
    {medicalHistory: parameters, user: props.user.user},
  ).then((result) => {
    history.push('/users')
    history.push(`medicalHistoryDetail/${parameters.nhist}`, medicalHistory)
  }).catch((error) => {
    setError(error.response.data)
    setShow(false)
    console.log(error);
  });
}

```

Figura 21 - Función para enviar el historial al back para actualizarlo

```

class updateMedicalHistory(APIView):
    def post(self,request,nhist):
        MedicalHistoryData = request.data['medicalHistory']
        idDoctor = request.data['user']['id']
        myMedicalHistory = MedicalHistory.objects.get(id = request.data['medicalHistory']['id'])
        #if(idDoctor == myMedicalHistory.idDoctor_id):
        if request.method == "POST":
            data = {}
            for attribute in MedicalHistoryData:
                data[attribute] = MedicalHistoryData.get(attribute)
            data['idDoctor'] = UserProfile(MedicalHistoryData.get('idDoctor'))
            data['id'] = MedicalHistoryData['id']
            if 'nhist' not in data:
                return Response({'El formulario debe tener un numero de Historia del Paciente'}, status = status.HTTP_400_BAD_REQUEST)
            if 'fechaVisita' not in data:
                return Response({'El formulario debe tener una fecha valida'}, status = status.HTTP_400_BAD_REQUEST)
            if 'numVisita' not in data:
                return Response({'El formulario debe tener un numero de visita'}, status = status.HTTP_400_BAD_REQUEST)
            newMedicalHistory= MedicalHistory(**data)
            myMedicalHistory = newMedicalHistory
            myMedicalHistory[attribute] = data[attribute]

            myMedicalHistory.save()
            return Response({'Historial Medico actualizado correctamente'}, status = status.HTTP_200_OK)

```

Figura 22 - Función para actualizar un historial

También tendremos la opción de archivar el historial si ya no deseamos verlo más en la aplicación. Pulsando el botón de archivar debajo del de actualizar. Esto provocará que el valor de isActive del historial pase a tener un valor de falso provocando que no se muestre.

Por último, en esta vista podemos ver las diferentes visitas del paciente con este historial. Si vamos al final de la página y el paciente tiene otras vistas podremos ver los números de las distintas visitas. En la Figura 24 podemos ver como se mostraría si el paciente tiene dos visitas, por defecto al pulsar sobre la tarjeta de un historial accederemos a los detalles del historial de la última visita realizada y podremos acceder al a visita número 1 si queremos.

Antes de Enviar el formulario, revisa que todos los campos son correctos, y que no hay ningun campo obligatorio en blanco

[Otras Visitas: 1](#)

Figura 23 - Distintas visitas

En la Figura 25 vemos la función que obtiene las visitas con el mismo historial, si hay más de una se muestra las otras visitas.

```
const getPagesMedicalHistory = () =>{
  axios.post(`http://localhost:8000/medicalHistorys/getMedicalHistory/${parameters.nhist}`,
    {medicalHistory: parameters, user: props.user.user.id},
  ).then((result) => {
    if(result.data.medicalHistory.length > 0){
      setShowVisits(true)
      setVisitsMedicalHistory(result.data.medicalHistory)
    }
  }).catch((error) => {
    setError(error.response.data)
    console.log(error);
  });
}
```

Figura 24 - Función para mostrar otras visitas

6.2.4 Obtener Predicción

La vista obtener predicción está formada por dos partes distintas: la primera se trata de la búsqueda del historial del que queremos saber la predicción, en la barra de búsqueda puedes escribir el número del historial y pulsar el botón de “Buscar” para que aparezca la última visita de ese historial y un nuevo botón “Obtener predicción” como se muestra en la Figura 26

Numero de historial del paciente Buscar



NÚMERO DE HISTORIAL: 1

Doctor: Roberto

Visita 2022-02-07

Obtener Prediction

Figura 25 - Buscar un historial para obtener predicción

Si pulsamos en la tarjeta del historial médico se nos enviará a la información detallada del historial.

Al pulsar el botón de “Obtener predicción” se muestra en la parte derecha la información de la predicción (Figura 26).

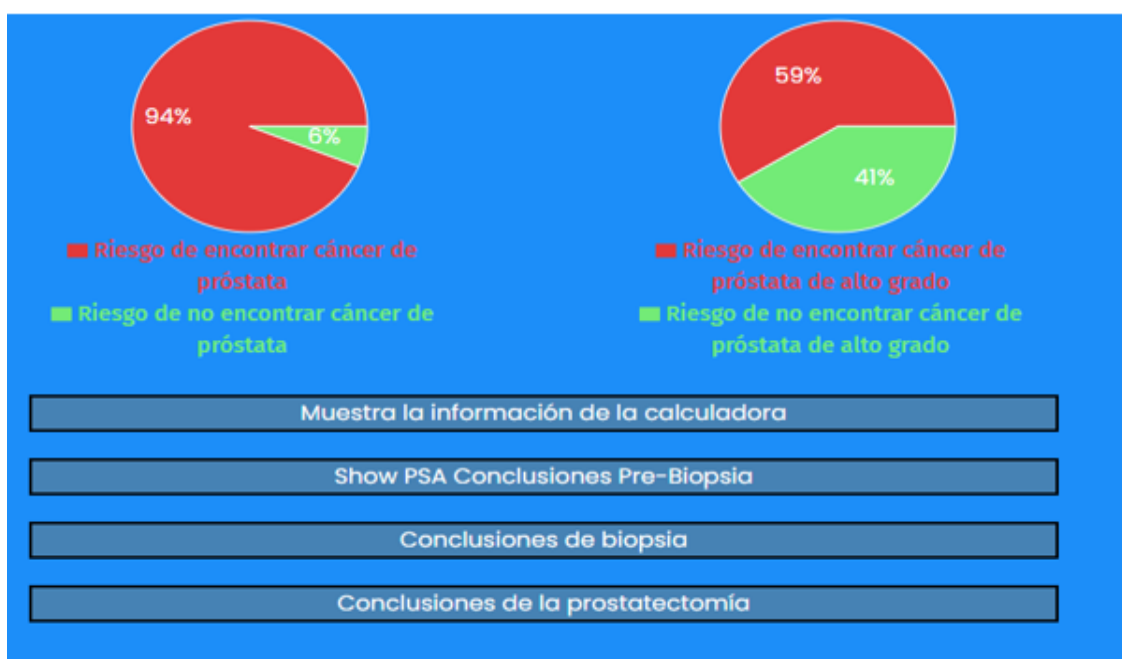


Figura 26 - Gráfica resultante de la predicción

El resultado de la calculadora se muestra mediante dos gráficos circulares, en ambos la parte roja representa la probabilidad de encontrar cáncer de próstata al realizar una biopsia y la parte verde la probabilidad de no encontrar cáncer de próstata. El primer gráfico representa la probabilidad encontrar un cáncer indolente, es decir el resultado de la biopsia mostraría una puntuación de Gleason de 6 o menos para el cáncer,

mientras que el gráfico de la derecha representa la probabilidad de encontrar cáncer de alto grado que sería una puntuación de Gleason de 8, 9 o 10. Esta es la opción que nos interesa porque como hemos comentado anteriormente un cáncer de próstata de bajo grado no significa que la vida del paciente corre peligro.

Para mostrar el resultado de la calculadora en los gráficos circulares se utiliza una librería de React llamada “PieChart” [22]. En la Figura 28 se muestra el código que crea el gráfico.

```
<PieChart width={300} height={270}>
  <Pie
    key={Math.random()}
    data={riesgoDeCancerProstata}
    style={{ backgroundColor: "white", borderRadius: "10px"}}
    cx="50%"
    cy="50%"
    labelLine={false}
    label={renderCustomizedLabel}
    outerRadius={80}
    fill="#8884d8"
    dataKey="value"
  >
    {riesgoDeCancerProstataAltoNivel.map((entry, index) => (
      <Cell key={`cell-${index}`} fill={COLORS[index % COLORS.length]} />
    ))}
  </Pie>
  <Tooltip content={<CustomTooltip />}/>
  <Legend />
</PieChart>

<PieChart width={300} height={270}>
  <Pie
    key={Math.random()}
    data={riesgoDeCancerProstataAltoNivel}
    style={{ backgroundColor: "white", borderRadius: "10px"}}
    cx="50%"
    cy="50%"
    labelLine={false}
    label={renderCustomizedLabel}
    outerRadius={80}
    fill="#8884d8"
    dataKey="value"
  >
    {riesgoDeCancerProstataAltoNivel.map((entry, index) => (
      <Cell key={`cell-${index}`} fill={COLORS[index % COLORS.length]} />
    ))}
  </Pie>
  <Tooltip content={<CustomTooltip />}/>
  <Legend />
</PieChart>
```

Figura 27 - Mostrar gráfico de probabilidad

Debajo del gráfico de la Figura X se muestran los botones con la información de la calculadora y, si es necesario, información extra sobre los datos y tratamiento recomendados si el paciente se ha realizado otras intervenciones.

6.2.5 Calculadora

En esta parte se explica que variables usa la calculadora y cuáles son las acciones recomendadas a seguir a partir de la predicción y por qué.

La calculadora estima la probabilidad de tener un PCa detectable insignificante (1 - 6) y un PCA (>6), donde se consideraría cáncer agresivo o de alto grado, usando modelos multivariantes de regresión lógica. La calculadora está basada en el modelo de datos ERSPC (The European Randomized study of Screening for Prostate Cancer) [20] de hombres de entre 55 – 75 años. La calculadora estima la probabilidad de una biopsia positiva utilizando los datos de PSA total, el resultado del tacto rectal, el PI-RAD y el volumen de la próstata [21]. En la Figura 29 se muestra la función que recibe el historial del paciente y decide qué tipo de calculo que ha de realizarse, teniendo en cuenta el resultado de tacto rectal y el resultado de la biopsia si se ha realizado anteriormente.

```
class calculateDiagnostic(APiView):
    def get(self, request, nhist):
        finalData = []
        fechas = []
        myMedicalHistorys = MedicalHistory.objects.filter(nhist=nhist)
        SerializerMedicalHistorys = serializers.MedicalHistorySerializer(myMedicalHistorys, many=True)

        for medicalHistory in SerializerMedicalHistorys.data:
            dre = 0
            biopsiaPos = 0
            nbipsia2 = medicalHistory['nbipsia']
            if nbipsia2 is not None:
                biopsiaPos = 1
            if medicalHistory['tactor'] == 'POSITIVO':
                dre = 1
            finalData.append((0 : medicalHistory['edad'], 1: float(medicalHistory['psa']), 2: float(medicalHistory['diametroMaximoLesionRm']),
            3: medicalHistory['hallazgosRadiologicos'], 4 : dre, 5 : biopsiaPos))
            fechas.append(medicalHistory['fechaVisita'].replace("-", ""))
        # Init normal/serious cancer risk; sample-by-2
        risk = np.empty((len(finalData),3))

        # Loop over data
        for idx, item in enumerate(finalData):
            if item[5] == 0:
                detectable_cancer_risk = round( pcrf_formula_prev_biopsy_neg(item[1],item[4],volume_to_volume_classes(item[2]),item[0],item[3]), 3)
                significant_cancer_risk = round( pcrf_formula_prev_biopsy_neg_serious(item[1],item[4],volume_to_volume_classes(item[2]),item[0],item[3]), 3)
            else:
                detectable_cancer_risk = round( pcrf_formula_prev_biopsy_pos(item[1],item[4],volume_to_volume_classes(item[2]),item[0],item[3]), 3)
                significant_cancer_risk = round( pcrf_formula_prev_biopsy_pos_serious(item[1],item[4],volume_to_volume_classes(item[2]),item[0],item[3]), 3)
            # Append
            risk[idx] = np.array([detectable_cancer_risk,significant_cancer_risk,float(fechas[idx])])
        return Response({
            'result' : risk,
            'message':'Calculo exitoso del Historial Medico con id ='+ str(nhist)
        })
```

Figura 28 - Función que calcula la probabilidad de biopsia positiva

La función recibe por parámetro en historial médico del paciente, se comprueba si el paciente se ha realizado una biopsia previa comprobado que el campo número de biopsia no esté vacío. Si está vacío, para hacer la predicción se usará el campo de “pi-rads prebiopsia”. Ser llamarán a las funciones de “pcrf_formula_pre_biopsia_neg”

(Figura 30) y “pcrc_formula_pre_biopsia_neg_serious” (Figura 31) si el paciente no se ha realizado una biopsia o esta ha sido negativa.

```
def pcrc_formula_prev_biopsy_neg(psa, dre_outcome, volume, age, pirads):
    """ Get chance prediction if previous biopsy was negative previously """

    priorbiopsy = 1.0
    # Volume class
    volume_class = volume
    wpirads = 0.0

    if(int(pirads) == 3):
        wpirads = 0.08258
    elif(int(pirads) == 4):
        wpirads = 1.40371
    elif(int(pirads) == 5):
        wpirads = 1.91967

    psaLog2 = math.log(psa) / math.log(2)
    volumeLog2 = math.log(volume_class) / math.log(2)

    val_prior = -2.62636 + (-0.745) + ((((-1.470 - 0.677 * priorbiopsy + (0.576 - 0.423 * priorbiopsy) *
    [(psaLog2 - 2)] - 1.043 * (volumeLog2 - 5.5) + 0.68 * dre_outcome)) * 1.4857 + 2.3426)) * 0.78749 + age * 0.02467 + wpirads

    PposRC45DRE_PHI = 1/(1+math.exp(-(val_prior)))

    return PposRC45DRE_PHI
```

Figura 29 - Probabilidad de cáncer de bajo grado con biopsia negativa

```
def pcrc_formula_prev_biopsy_neg_serious(psa, dre_outcome, volume, age, pirads):
    """ Get chance prediction if previous biopsy was negative previously serious"""

    priorbiopsy = 1.0
    # Volume class
    volume_class = volume
    wpirads = 0.0

    if(int(pirads) == 3):
        wpirads = 0.94909
    elif(int(pirads) == 4):
        wpirads = 2.67536
    elif(int(pirads) == 5):
        wpirads = 3.05243

    psaLog2 = math.log(psa) / math.log(2)
    volumeLog2 = math.log(volume_class) / math.log(2)

    val_prior_ser = -5.19288 + (-0.745) + (((((-3.489 - 1.136 * priorbiopsy + (1.075 - 0.434 * priorbiopsy) *
    (psaLog2 - 2) - 1.501 * (volumeLog2 - 5.5) + 1.311 * dre_outcome))] * 0.9420 + 2.1452)) * 0.74027 + age * 0.04286 + wpirads

    PposRC45DRE_PHI_ser = 1/(1+math.exp(-(val_prior_ser)))

    return PposRC45DRE_PHI_ser
```

Figura 30 - Probabilidad de cáncer de alto grado con biopsia negativa

Si el paciente se ha realizado una biopsia previa y ha sido positiva en vez del campo “pi-rads prebiopsia” se usará el campo “pi-rads segunda y consecutivas biopsias” y las funciones serán “pcrc_formula_pre_biopsia_pos” (Figura 32) y “pcrc_formula_pre_biopsia_pos_serious” (Figura 33).

```
def pcrf_formula_prev_biopsy_pos(psa, dre_outcome, volume, age, pirads):
    """ Get chance prediction if previous biopsy was positive previously """

    # Volume class
    volume_class = volume
    wpirads = 0.0

    if(int(pirads) == 3):
        wpirads = -0.38169
    elif(int(pirads) == 4):
        wpirads = 0.61290
    elif(int(pirads) == 5):
        wpirads = 2.01363

    # Based on logistic regression
    psalog2 = math.log(psa) / math.log(2)
    volumelog2 = math.log(volume_class) / math.log(2)

    # Weights/bias
    aux = ((-1.826 + 1.024 * (psalog2 - 2.0) - 1.50 * (volumelog2 - 5.4) + 0.992 * dre_outcome) * 1.139 ) + 1.325

    val_prior = -3.52647 + (-1.165) + aux * 0.78810 + age * 0.04832 + wpirads

    PposRC45DRE_PHI = 1/(1+math.exp(-(val_prior)))

    return PposRC45DRE_PHI
```

Figura 31 - Probabilidad de cáncer de bajo grado con biopsia positiva

```
def pcrf_formula_prev_biopsy_pos_serious(psa, dre_outcome, volume, age, pirads):
    """ Get chance prediction if previous biopsy was positive previously serious """

    volume_class = volume
    wpirads = 0.0

    if(int(pirads) == 3):
        wpirads = 0.44510
    elif(int(pirads) == 4):
        wpirads = 1.47778
    elif(int(pirads) == 5):
        wpirads = 2.61551

    calc_age = age * 0.04055
    calc_psalog2 = ( ( math.log(psa) / math.log(2) ) - 2.0 ) * 1.177
    calc_dre = 1.813 * dre_outcome
    calc_volume = (math.log(volume_class) / math.log(2) - 5.4) * -1.526
    intercept = -3.457

    aux = -4.05287 + (-1.61) + ( ( intercept + calc_psalog2 + calc_dre + calc_volume ) * 0.718 + 1.069 ) * 0.69846 + calc_age + wpirads
    PposRC45DRE_PHI_ser = math.exp(aux)/(1+math.exp(aux))

    return PposRC45DRE_PHI_ser
```

Figura 32 - Probabilidad de cáncer de alto grado con biopsia positiva

6.2.6 Información del historial

En este apartado se explicará la información que muestran los diferentes botones debajo de los gráficos al realizar una predicción.

6.2.6.1 Botón Información de la calculadora.

Se muestra la información de las variables que intervienen en el cálculo de la predicción junto con las acciones recomendadas según la probabilidad de obtener una biopsia positiva (Figura 35).

Una biopsia prostática es recomendada si la probabilidad de una biopsia positiva es $\geq 20\%$. Este límite del 20% es comparable a una predicción positiva sobre un valor de Psa ≥ 4 ng/ml en la población general. Además, como regla general se recomienda vigilancia activa del paciente si el resultado de la predicción de cáncer indolente es $> 70\%$ y tratamiento activo en el resto de los casos.

Estas recomendaciones se hacen en base a al estudio “Impact of Individual Risk Assessment on Prostate Cancer Diagnosis” de Heidi van Vugt el cual trata sobre el uso de la misma calculadora de riesgo de cáncer de próstata que hemos usado en este proyecto. En el estudio de Heidi van Vugt se recomendaba el mismo procedimiento de biopsia prostática, el resultado final del estudio resultó en un 68% de predicciones positivas usando la calculadora frente a un 44% de predicciones positivas sin usar la calculadora, además se encontró un 34% de cánceres de alto grado usando la calculadora frente a un 15% sin usarla. Por lo tanto, frente estos resultados deducimos que si seguimos el mismo procedimiento obtendremos unos resultados positivos. [21]

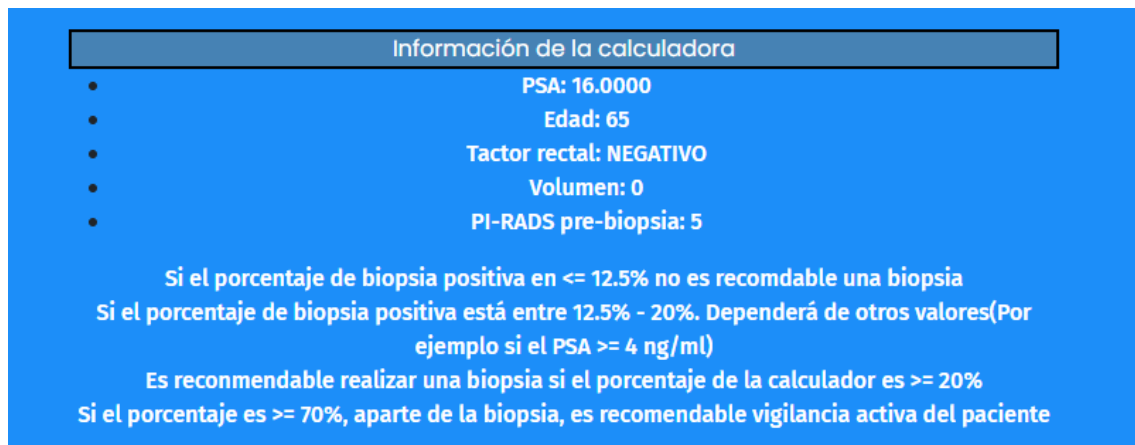


Figura 33 - Información de la calculadora

6.2.6.2 Botón información de los parámetros clínicos-patológicos

Podemos ver la información del historial de los datos clínicos patológicos. En este caso no se muestra información extra de las variables porque la información más relevante se obtiene en la sección de la calculadora (Figura 36). Lo más destacable es la información extra del PSA que informa sobre la probabilidad de cáncer de próstata [23] y los datos obtenidos del número de cores positivos y el % de afectación de estos cores.

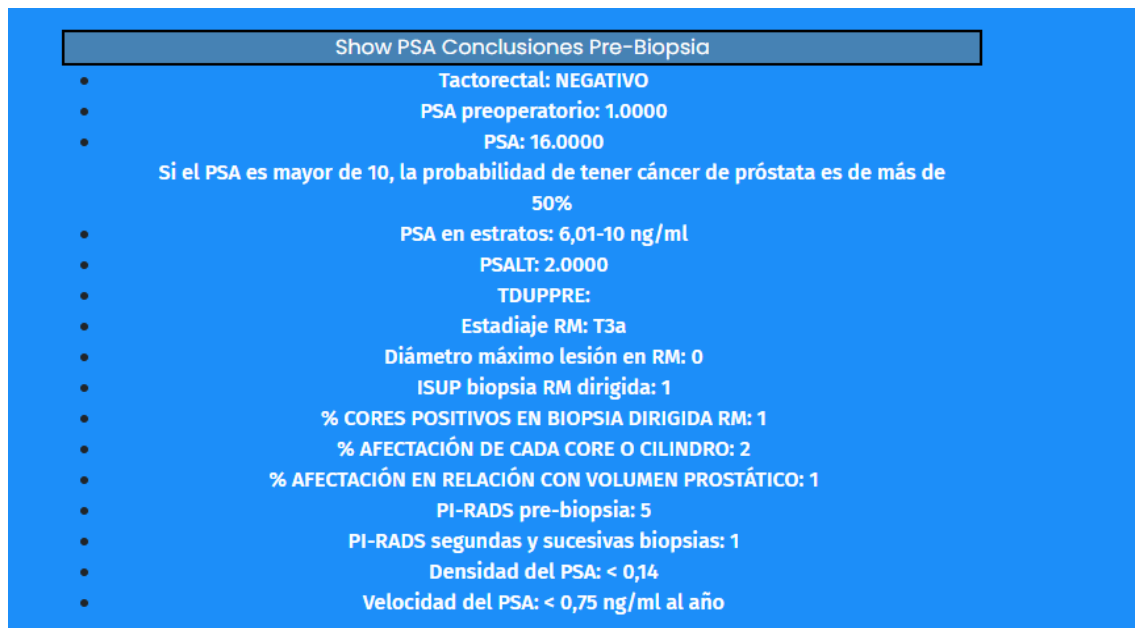


Figura 34 - Información de Clínico Patológicas

6.2.6.3 Botón información de biopsia

Si el paciente se ha sometido a una biopsia podemos ver la información del historial de los datos realizados tras la biopsia además de información sobre estos, posibles tratamientos y su clasificación de estadio. En la información de la biopsia podemos ver mucha información extra (Figura 37). Las partes más destacables son el cTNM [24] del cual podemos sacar un tratamiento recomendado [25] y el valor del estadio [26], el cual se deduce del resto de variables. En la Figura 36 podemos ver como calcular el estadio.

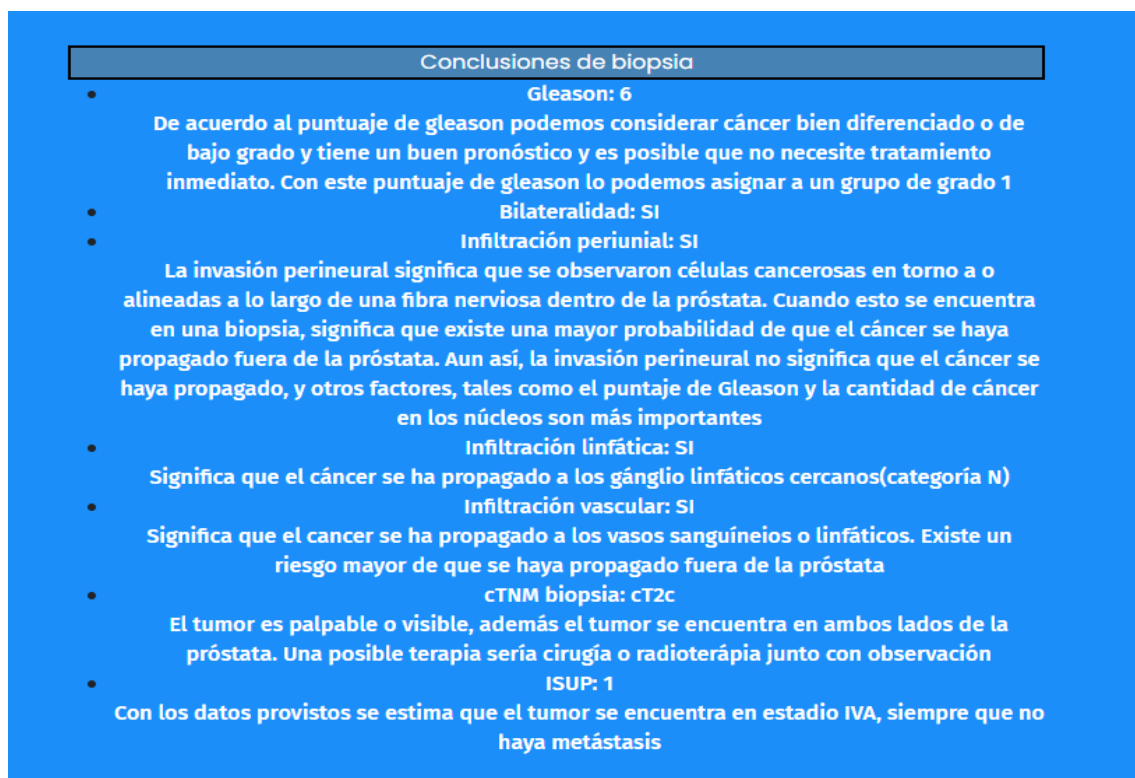


Figura 35 - Información de la biopsia

```

const calcEstadio = ([data]) =>{
  const sumaGleason = data['gleasonPrimarioBiopsiaProstática'] + data['gleasonSecundarioBiopsiaProstática']
  //Estadio I
  if((data['ctnmBiopsia'] === 'cT1' || data['ctnmBiopsia'] === 'cT2a') && data['ilinf'] === 'NO' && data['gleasonEnBiopsia'] === '6' && data['psa'] < 10){
    return "Con los datos provistos se estima que el tumor se encuentra en estadio I, siempre que no haya metástasis"
  }
  //Estadio IIA
  if((data['ctnmBiopsia'] === 'cT1' || data['ctnmBiopsia'] === 'cT2a') && data['ilinf'] === 'NO' && data['gleasonEnBiopsia'] === '6' && (data['psa'] >= 10 || data['psa'] < 20)){
    return "Con los datos provistos se estima que el tumor se encuentra en estadio IIA, siempre que no haya metástasis"
  }
  if((data['ctnmBiopsia'] === 'cT2b' || data['ctnmBiopsia'] === 'cT2c') && data['ilinf'] === 'NO' && data['gleasonEnBiopsia'] === '6' && data['psa'] < 20){
    return "Con los datos provistos se estima que el tumor se encuentra en estadio IIA, siempre que no haya metástasis"
  }
  //Estadio IIB
  if((data['ctnmBiopsia'] === 'cT1' || data['ctnmBiopsia'] === 'cT2a' || data['ctnmBiopsia'] === 'cT2b' || data['ctnmBiopsia'] === 'cT2c')
    && data['ilinf'] === 'NO' && data['gleasonEnBiopsia'] === '7 (3+4)' && data['psa'] < 20){
    return "Con los datos provistos se estima que el tumor se encuentra en estadio IIB, siempre que no haya metástasis"
  }
  //Estadio IIC
  if((data['ctnmBiopsia'] === 'cT1' || data['ctnmBiopsia'] === 'cT2a' || data['ctnmBiopsia'] === 'cT2b' || data['ctnmBiopsia'] === 'cT2c')
    && data['ilinf'] === 'NO' && data['gleasonEnBiopsia'] === '7 (4+3)' || sumaGleason === 8 && data['psa'] < 20){
    return "Con los datos provistos se estima que el tumor se encuentra en estadio IIC, siempre que no haya metástasis"
  }
  //Estadio IIIA
  if((data['ctnmBiopsia'] === 'cT1' || data['ctnmBiopsia'] === 'cT2a' || data['ctnmBiopsia'] === 'cT2b' || data['ctnmBiopsia'] === 'cT2c')
    && data['ilinf'] === 'NO' && sumaGleason <= 8 && data['psa'] >= 20){
    return "Con los datos provistos se estima que el tumor se encuentra en estadio IIIA, siempre que no haya metástasis"
  }
  //Estadio IIIB
  if((data['ctnmBiopsia'] === 'cT3a' || data['ctnmBiopsia'] === 'cT3b' || data['ctnmBiopsia'] === 'cT4' && data['ilinf'] === 'NO' && sumaGleason <= 8){
    return "Con los datos provistos se estima que el tumor se encuentra en estadio IIIB, siempre que no haya metástasis"
  }
  //Estadio IIIC
  if((sumaGleason === 9 || sumaGleason === 10)){
    return "Con los datos provistos se estima que el tumor se encuentra en estadio IIIC, siempre que no haya metástasis"
  }
  //Estadio IVA
  if(data['ilinf'] === 'SI'){
    return "Con los datos provistos se estima que el tumor se encuentra en estadio IVA, siempre que no haya metástasis"
  }
}

```

Figura 36 - Cálculo del estadio

6.2.6.4 Botón información de la prostatectomía

Si el paciente se ha sometido a una prostatectomía podemos ver la información de los datos realizados tras la intervención (Figura 38). Las conclusiones de la prostatectomía es la sección que menos información muestra, solamente los valores del historial médico de la visita. Ya que consideramos que, si ha sido necesario realizar una prostatectomía el paciente ya tenía un cáncer diagnosticado y se le recomendó un tratamiento, la propia prostatectomía, por lo que no se considera un caso relevante para este proyecto.

Conclusiones de la prostatectomía	
•	Tratamiento: PROSTATECTOMÍA RADICAL
•	GLEASON PIEZA PRL: 7 (4+3)
•	GLEASON PRIMARIO: (3-5): 4
•	GLEASON SECUNDARIO: (3-5): 3
•	BILATERALIDAD: SI
•	LOCALIZACIÓN EN PIEZA: ZFMA
•	MULTIFOCALIDAD: SI
•	VOLUMEN TUMORAL (% PIEZA): 23.0000
•	EXTENSIÓN EXTRACASPULAR: SI
•	INVASIÓN VESÍCULAS SEMINALES: NO
•	INFILTRACIÓN PERINEURAL PIEZA:
•	INFILTRACIÓN LINFÁTICA PIEZA: NO
•	INFILTRACIÓN VASCULAR PIEZA: SI
•	PIN ALTO GRADO PIEZA: NC
•	MÁRGENES QUIRÚRGICOS POSITIVOS: NO
•	TNM PIEZA: pT2c

Figura 37 - Información de la prostatectomía

Se mantiene la información de la prostatectomía y los datos evolutivos para recaudar la máxima información posible de cara a poder realizar un modelo de datos útil

Capítulo 7 Desarrollo y documentación de API REST

En este capítulo se describirá el diseño de la API REST. La API está formada por tres grupos: medicalHistorys, prostascreen_api y users.

7.1 MedicalHistorys

En Figura 39 que se muestra a continuación se establece la relación entre cada endpoint utilizado en el módulo de medicalHistorys, el cual se encarga de gestionar todo lo relacionado con los historiales médicos, y los métodos HTTP [27] asociados a estos, así como una breve explicación de la funcionalidad de cada uno de ellos que se muestra en la Figura 40.

Endpoint	GET	POST	PUT	DELETE
medicalHistory/newMedicalHistory		X		
medicalHistory/getAllMedicalHistorys	X			
medicalHistory/getMedicalHistory/<int:nhist>	X			
medicalHistory/search	X			
medicalHistory/updateMedicalHistory/<int:nhist>			X	
medicalHistory/archive/<int:nhist>		X		
medicalHistory/getPrediccion/<int:nhist>	X			

Figura 38 - Tipos de endpoints utilizados en medicalHistorys

Ahora se describirá cada endpoint.

Endpoint	Descripción
medicalHistory/newMedicalHistory	Creación de un nuevo historial médico
medicalHistory/getAllMedicalHistorys	Obtención de todos los historiales médico
medicalHistory/getMedicalHistory/<int:nhist>	Obtención de la inforamción detallada de un historial médico
medicalHistory/search	Buscar historail médico según su número de historial
medicalHistory/updateMedicalHistory/<int:nhist>	Actualizar la información de un historail médico
medicalHistory/archive/<int:nhist>	Archivar un historial médico
medicalHistory/getPrediccion/<int:nhist>	Obtener la predicción de un historial médico

Figura 39 - Descripción de endpoints utilizados en medicalHistorys

7.2 Prostrascreen_api

En Figura 41 que se muestra a continuación se establece la relación entre cada endpoint utilizado en el módulo de prostascreen_api, el cual se encarga de gestionar el login y el log out, y los métodos HTTP asociados a estos (Figura), así como una breve explicación de la funcionalidad de cada uno de ellos que se muestra en la Figura 42.

Endpoint	GET	POST	PUT	DELETE
prostascreen_api/login		X		
prostascreen_api/logout		x		

Figura 40 - Tipos de endpoints utilizados en prostascreen_api

Ahora se describirá cada endpoint.

Endpoint	Descripción
prostascreen_api/login	Iniciar sesión
prostascreen_api/logout	Cerrar sesión

Figura 41 - Descripción de endpoints utilizados en prostascreen_api

7.3 Users

En la Figura 43 que se muestra a continuación se establece la relación entre cada endpoint utilizado en el módulo de users, el cual se encarga de gestionar la información de usuario, y los métodos HTTP asociados a estos, así como una breve explicación de la funcionalidad de cada uno de ellos que se muestra en la Figura 44.

Endpoint	GET	POST	PUT	DELETE
users/home	X			
users/update/<int:id>		X		

Figura 42 - Tipos de endpoints utilizados en users

Ahora se describirá cada endpoint.

Endpoint	Descripción
users/home	Muestra la información de usuario
users/update/<int:id>	Actualiza la información del usuario

Figura 43 - Descripción de endpoints utilizados en users

Capítulo 8 Conclusiones y trabajo a futuro

En este capítulo se desarrollarán las conclusiones extraídas al realizar la aplicación, así como lista de posibles mejoras para la aplicación.

8.1 Conclusiones

En este proyecto se ha desarrollado una aplicación web capaz de registrar y almacenar usuarios e historiales médicos de vistas de pacientes. Los usuarios podrán llevar un seguimiento de los historiales de las visitas, modificarlos y archivarlos.

Además, se ha implementado una calculadora capaz de, utilizando la información de la visita de un paciente, calcular la probabilidad de que esta sufra de cáncer de próstata y recomendar acciones a seguir.

Si el paciente ya se ha realizado una biopsia prostática y ha dado positivo, la aplicación también mostrará información extra sobre el resultado de la biopsia y en base a el resultado recomendará acciones a seguir. Por último, si el paciente se ha sometido a una prostatectomía la aplicación mostrará la información de esta.

Se han planteado una serie de ideas para mejorar las funcionalidades de la aplicación que se describirán a continuación.

El código del proyecto se puede descargar en el siguiente enlace:

8.2 Trabajo a futuro

A continuación, se lista varias mejoras para la funcionalidad de la aplicación:

- **Diferentes calculadoras:** implementación de diferentes calculadoras que usen diferentes parámetros para tener la mayor certeza posible del diagnóstico.
- **Calculadora tras biopsia:** implementación de una calculadora que recomiende el mejor tratamiento en cuestión de los datos obtenidos de la biopsia utilizando un modelo de datos.
- **Calculadora tras prostatectomía:** en este momento la aplicación solamente recomienda tratamientos tras haber usado la calculadora o haber realizado una biopsia. Una calculadora para recomendar el mejor tratamiento posible tras una prostatectomía sería muy útil.
- **Estudiar la evolución de un paciente:** seguir avanzando en el estudio de la evolución de un paciente una vez se le ha dado un mal pronóstico para poder recomendar le mejor tratamiento a seguir posible.
- **Actualización de la interfaz:** adaptar la aplicación para representar la información sobre la evolución de un paciente de la manera más clara e intuitiva posible.

- **Crear un modelo de datos:** usar la aplicación lo máximos posible y recopilar toda la información para, en un futuro, poder crear un modelo de datos que ayude al diagnóstico de cáncer de próstata

Chapter 8 Conclusions and future work

In this chapter the conclusions drawn when carrying out the application will be developed, as well as a list of possible improvements for the application.

8.1 Conclusions

In this project, a web application capable of registering and storing users and medical records of patient visits has been developed. Users will be able to track, modify, and archive visit histories.

In addition, a calculator capable of calculating the probability that the patient suffers from prostate cancer and recommending actions to follow has been implemented using the information from a patient's visit.

If the patient has already realized a prostate biopsy and it has been positive, the application will also show extra information about the biopsy result and, based on the result, will recommend actions to follow. Finally, if the patient has undergone a prostatectomy, the application will show the information on this.

A series of ideas have been proposed to improve the functionalities of the application that will be described below.

8.3 Future work

Below is a list of several improvements to the functionality of the application:

- **Different calculators:** implementation of different calculators that use different parameters to have the greatest possible certainty of the diagnosis.
- **Calculator after biopsy:** implementation of a calculator that recommends the best treatment based on the data obtained from the biopsy.
- **Post Prostatectomy Calculator:** At this time the app only recommends treatments after you have used the calculator or performed a biopsy. A calculator to recommend the best possible treatment after a prostatectomy would be very useful.
- **Study the evolution of a patient:** continue advancing in the study of the evolution of a patient once a poor prognosis has been given in order to recommend the best possible treatment to follow.
- **Update of the interface:** adapt the application to represent the information on the evolution of a patient in the clearest and most intuitive way possible.
- **Create a data model:** use the application as much as possible and collect all the information to, in the future, be able to create a data model that helps diagnose prostate cancer.

Capítulo 9 Aportaciones individuales

9.1 Gonzalo Hernández Hernández

- Fue el responsable de creación y diseño e implementación de la mayoría de las vistas de la aplicación
- Participó en el diseño y la implementación de la base de datos.
- Participó en la implementación de algunos endpoints, así como su desarrollo para la API.
- Investigó el uso de la calculadora para poder integrarla en la aplicación y usara diferentes parámetros dependiendo de si el paciente había pasado por una biopsia positiva o no
- Investigó el proyecto “Impact of Individual Risk Assessment on Prostate Cancer Diagnosis” para poder recomendar la mejor acción posible dependiendo del resultado de la calculadora
- Investigó sobre la biopsia prostática y los datos resultantes para poder mostrar toda la información posible al médico, poder determinar el estadio del cáncer y poder recomendar el mejor tratamiento posible
- Realizó la memoria

9.2 Roberto Torres Prensa

Comenzó siendo parte de este proyecto, pero debido a que no pudo matricularse finalmente para participar en el TFG debido al número de créditos que le faltaban, a partir de febrero de 2022 el proyecto pasó a realizarse únicamente por Gonzalo Hernández Hernández.

Aun así, participó en la primera mitad del proyecto:

- Se encargó de organizar la estructura de paquetes del proyecto.
- Participó en el diseño y la implementación de la base de datos.
- Se encargó de realizar el desarrollo y la implementación de la mayoría de la parte back
- Se encargó de gestionar el deploy en heroku así cómo se aseguró de que al hacer un commit al github del proyecto el commit se hiciera y también en heroku para que así la aplicación estuviera siempre actualizada.
- Participó en en desarrollo y corrección de errores de múltiples vistas
- Investigó una calculadora con la mayor documentación posible y la replicó en la aplicación

Bibliografía

- [1] React JS. <https://es.reactjs.org/>.
- [2] JavaScript. <https://developer.mozilla.org/es/docs/Web/JavaScript>.
- [3] Angular JS. <https://angularjs.org/>.
- [4] Django. <https://www.djangoproject.com/>.
- [5] Xampp. <https://www.apachefriends.org/es/about.html>.
- [6] Heroku. <https://www.heroku.com/>.
- [7] Git. <https://git-scm.com/>.
- [8] ORM Django. https://tutorial.djangogirls.org/es/django_orm/.
- [9] Hipertensión arterial.
[https://medlineplus.gov/spanish/ency/article/000468.htm#:~:text=Una%20presi%C3%B3n%20arterial%20alta%20\(hipertensi%C3%B3n,se%20denomina%20presi%C3%B3n%20arterial%20elevada..](https://medlineplus.gov/spanish/ency/article/000468.htm#:~:text=Una%20presi%C3%B3n%20arterial%20alta%20(hipertensi%C3%B3n,se%20denomina%20presi%C3%B3n%20arterial%20elevada..)
- [10] Diabetes mellitus. <https://www.cancer.gov/espanol/publicaciones/diccionarios/diccionario-cancer/def/diabetes-mellitus>.
- [11] Dutasterida. <https://medlineplus.gov/spanish/druginfo/meds/a603001-es.html>.
- [12] Finasterida. <https://medlineplus.gov/spanish/druginfo/meds/a698016-es.html>.
- [13] PSA. <https://www.cancer.gov/espanol/tipos/prostata/hoja-informativa-psa>.
- [14] Estadíaje. [https://www.clinicbarcelona.org/asistencia/enfermedades/cancer/estadificacion-tnm#:~:text=El%20estadíaje%20\(o%20estadificaci%C3%B3n\)%20se,el%20mejor%20plan%20de%20tratamiento..](https://www.clinicbarcelona.org/asistencia/enfermedades/cancer/estadificacion-tnm#:~:text=El%20estadíaje%20(o%20estadificaci%C3%B3n)%20se,el%20mejor%20plan%20de%20tratamiento..)
- [15] ISUP.
https://www.scielo.sa.cr/scielo.php?script=sci_arttext&pid=S1409-00152017000100237.
- [16] PI-RADS. <https://www.elsevier.es/es-revista-radiologia-119-articulo-resonancia-magnetica-prostata-lectura-con-S0033833816301692>.

- [17] Tipo histiológico. <https://www.cancer.gov/espanol/publicaciones/diccionarios/diccionario-cancer/def/grado-histologico>.
- [18] Sistema de puntuación Gleason.
<https://medlineplus.gov/spanish/ency/patientinstructions/000920.htm>.
- [19] Clasificación TNM.
<https://urologiabe.com/2018/02/19/comprendiendo-el-estadiaje-del-cancer-tnm/>.
- [20] The European Randomized study of Screening for Prostate Cancer.
[https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4427906/#:~:text=The%20European%20Randomized%20study%20of%20Screening%20for%20Prostate%20Cancer%20\(ERSPC,testing%20in%20eight%20European%20countries..](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4427906/#:~:text=The%20European%20Randomized%20study%20of%20Screening%20for%20Prostate%20Cancer%20(ERSPC,testing%20in%20eight%20European%20countries..)
- [21] H. v. Vugt, Estudio "Impact of Individual Risk Assessment".
https://swopresearch.nl/wp-content/uploads/2018/04/Tekst_proefschrift_dr._Van_Vugt_-_Impact_studie.pdf.
- [22] Gráfico "PieChart". <https://apexcharts.com/react-chart-demos/pie-charts/>.
- [23] Probabilidad de cáncer en base al psa.
<https://www.cancer.org/es/cancer/cancer-de-prostata/deteccion-diagnostico-clasificacion-por-etapas/pruebas-de-deteccion-para-el-cancer-de-prostata.html>.
- [24] Valor cTNM. <https://www.cancer.org/es/cancer/cancer-de-prostata/deteccion-diagnostico-clasificacion-por-etapas/clasificacion-por-etapas.html>.
- [25] E. C. Paula J. Fonseca, «cTNM,» de *Genética y oncología*, 2018, p. 186.
- [26] Valor del estadio.
[https://www.cancer.net/es/tipos-de-cancer/cancer-de-prostata/estadios-y-gradoss#:~:text=Estadio%20IIIA%3A%20el%20c%C3%A1ncer%20se,de%20grados%20I%2D4\)..](https://www.cancer.net/es/tipos-de-cancer/cancer-de-prostata/estadios-y-gradoss#:~:text=Estadio%20IIIA%3A%20el%20c%C3%A1ncer%20se,de%20grados%20I%2D4)..)
- [27] HTTP. https://es.wikipedia.org/wiki/Protocolo_de_transferencia_de_hipertexto.

ANEXO I Guía del usuario

Para poder acceder a la aplicación el primer paso es acceder a la web a través de su URL: <https://prostascreen.herokuapp.com/>. Una vez accedes a la web se te muestra la pantalla de login en la Figura 43 se muestra, donde puedes acceder mediante un email y contraseña.

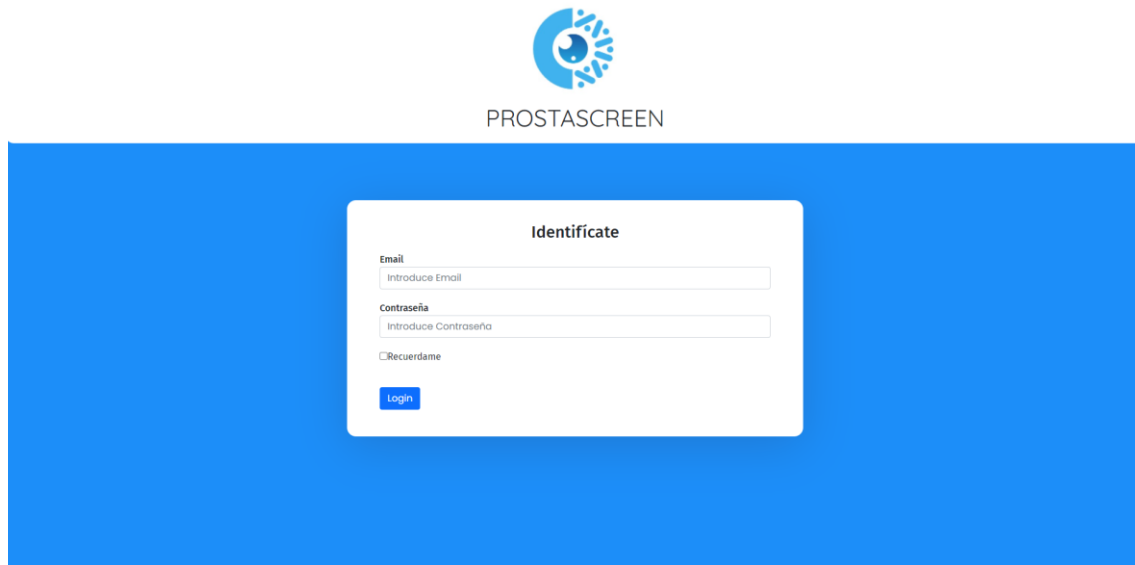


Figura 44 - Pantalla de login

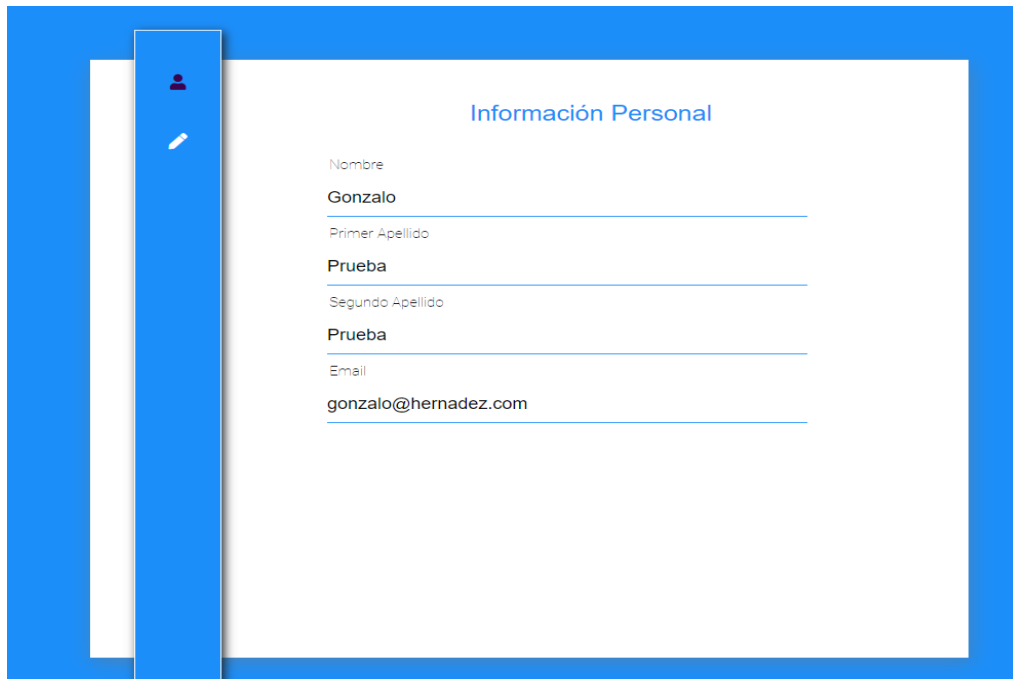
Como se ha comentado anteriormente, en esta fase inicial el único usuario capaz de acceder a la aplicación sería el doctor Víctor Manuel Carrero López.

Una vez accedemos a la aplicación se nos muestra el home el cual simplemente tiene el header como vemos en la Figura 44.



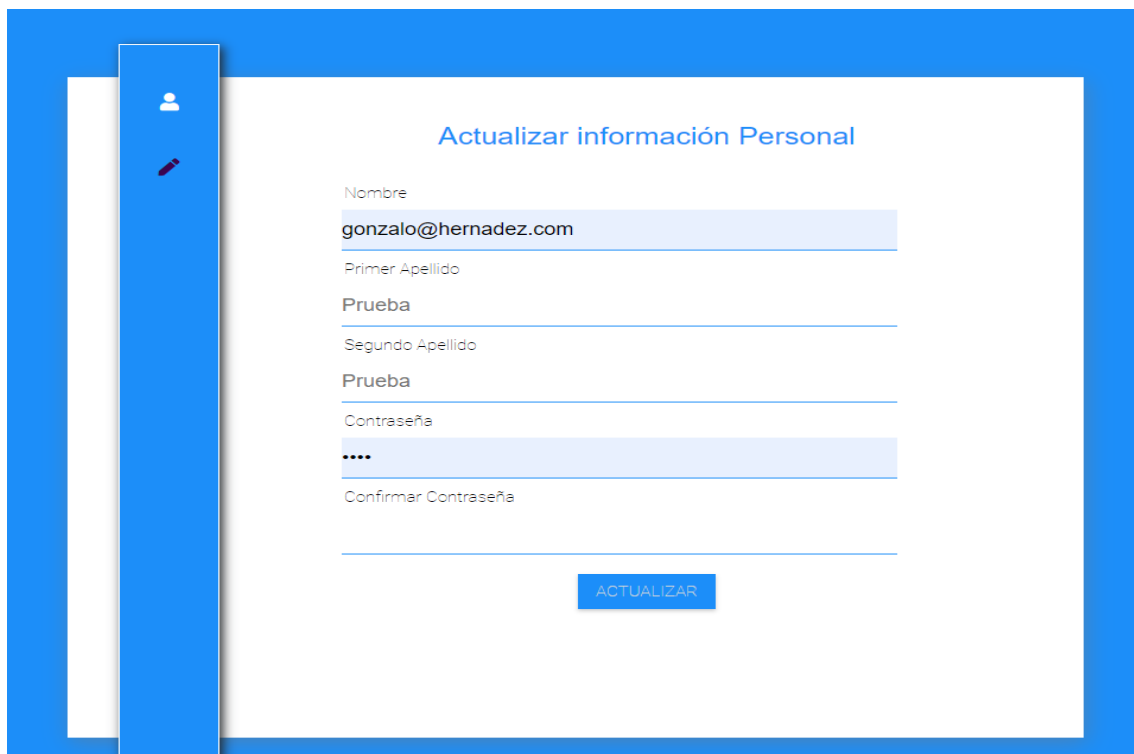
Figura 45- Home

Desde aquí si pulsamos encima del nombre de usuario en la parte derecha de header podremos ver la información del perfil (Figura 45), además tendremos la opción de modificar esta información si pulsamos sobre el icono del lápiz (Figura 46).



The screenshot shows a web interface with a blue header and a blue sidebar. The sidebar contains a user icon and a pencil icon. The main content area is titled 'Información Personal' and contains a form with the following fields: 'Nombre' (Gonzalo), 'Primer Apellido' (Prueba), 'Segundo Apellido' (Prueba), and 'Email' (gonzalo@hernandez.com). The form is displayed in a light blue box.

Figura 46- Información personal



The screenshot shows a web interface with a blue header and a blue sidebar. The sidebar contains a user icon and a pencil icon. The main content area is titled 'Actualizar información Personal' and contains a form with the following fields: 'Nombre' (gonzalo@hernandez.com), 'Primer Apellido' (Prueba), 'Segundo Apellido' (Prueba), 'Contraseña' (****), and 'Confirmar Contraseña'. The form is displayed in a light blue box. At the bottom right of the form is a blue button labeled 'ACTUALIZAR'.

Figura 47 - Modificar información personal

En la parte de arriba de la pantalla seguiremos teniendo presente el header en todo momento, si pulsamos el botón de nuevo expediente en la parte izquierda se nos mostrará la vista para

poder crear un nuevo historial médico. En la Figura 47 vemos lo que se nos muestra al pulsar el botón de nuevo expediente.

The screenshot shows the 'Nuevo Expediente' (New Record) form in the PROSTASCREEN application. At the top, there is a navigation bar with buttons for 'Expedientes', 'Nuevo Expediente', and 'Obtener Predicción'. The central header features the PROSTASCREEN logo and a user profile for 'Roberto' with a 'Cerrar Sesión' button. On the left, a sidebar lists various medical history categories: FILIACIÓN, SOCIODEMOGRÁFICAS, ANTECEDENTES, CLÍNICO-PATOLÓGICAS, BIOPSIA PROSTÁTICAS, TRAS PROSTATECTOMÍA, EVOLUTIVOS, and a blue 'ENVIAR FORMULARIO' button. The main content area is titled 'Filiación' and contains four input fields: 'NHIS' (text), 'FECHACIR' (date, dd/mm/aaaa), 'Nº DE VISITA' (text), and 'FECHA DE VISITA' (date, dd/mm/aaaa). Each date field has a calendar icon for selection.

Figura 48 - Nuevo historial

En la Figura 47 vemos como la vista está dividida en dos partes: en la parte izquierda tenemos los distintos grupos de variables que podemos introducir en el historial de la visita del paciente (Figura 48), mientras que en la zona central se encuentra la vista para poder introducir los datos del historial. Algunos datos tienen valores seleccionables mientras que otros los tenemos que introducirlos nosotros mismos.

This image displays a vertical list of variable groups for the medical history form. The groups are: FILIACIÓN, SOCIODEMOGRÁFICAS, ANTECEDENTES, CLÍNICO-PATOLÓGICAS, BIOPSIA PROSTÁTICAS, TRAS PROSTATECTOMÍA, EVOLUTIVOS, and a blue button labeled 'ENVIAR FORMULARIO' at the bottom.

Figura 49 - Grupos de variables

Una vez introducidos los datos, podemos ir a la sección de enviar formulario para poder registrar el historial. Para poder registrar el historial tenemos que asegurarnos de que los datos obligatorios estén rellenos y de que estén correctos.

Una vez con el historial registrado podemos pulsar en el botón de Expedientes en la parte de arriba a la izquierda del header para que se nos muestre la lista con los historiales. En esta vista tenemos la opción de buscar un historial en concreto si lo deseamos. En la Figura 49 podemos ver un ejemplo de como se muestra la lista de historiales.

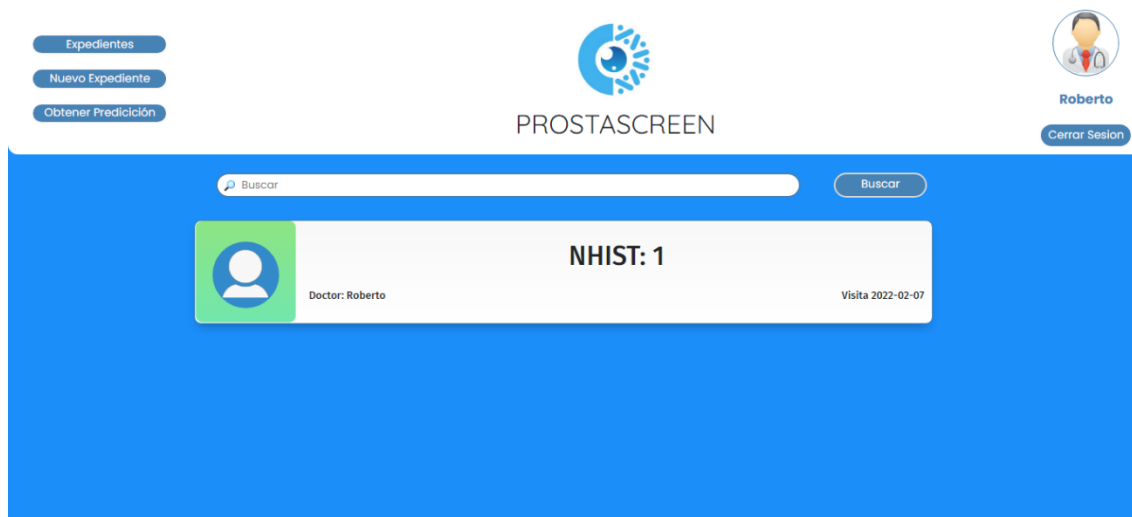


Figura 50 - Lista de historiales

Si pulsamos sobre la tarjeta de un historial podremos acceder a la información detallada del historial. En la Figura 50 se nos muestra la primera parte del historial completo, junto con los botones de modificar y archivar a la izquierda.

Figura 51 - Información detallada del historial

Desde aquí tenemos un par de opciones, la primera sería modificar algún campo del historial, para esto pulsamos el botón de modificar en la parte izquierda. Una vez realizados los cambios pulsamos el botón de enviar formulario al final de la página.

También podemos archivar el archivo y hacer que no se vuelva a mostrar en la aplicación al pulsare botón que está encima del de modificar.

Por último, si un paciente ha realizado varias visitas al final de la página se mostrará el número de visitas como se ve en la Figura 51, si queremos ir a una en específico simplemente clicamos en el número que queramos.

Antes de Enviar el formulario, revisa que todos los campos son correctos, y que no hay ningún campo obligatorio en blanco

Otras Visitas: 1

Figura 52 - Diferentes vistas

Para poder obtener la predicción de un paciente tenemos que pulsar en el botón de Obtener predicción en la parte de abajo a la izquierda del header.

Una vez pulsamos se nos mostrará una barra de búsqueda, allí escribiremos el número del historial del que queramos hacer una predicción y pulsamos el botón de obtener predicción que aparece.

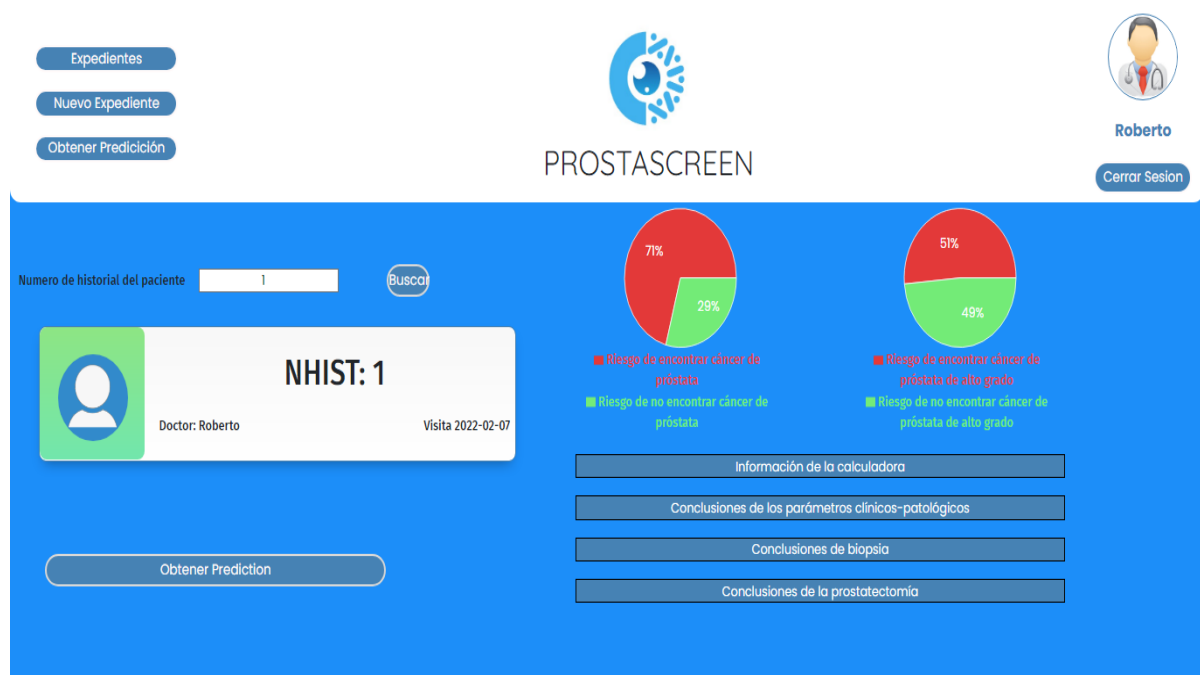


Figura 53 - Vista general al obtener una predicción

En la parte derecha de la pantalla aparecerá dos gráficos con las probabilidades de que el paciente sufra un cáncer indolente y un cáncer de alto grado respectivamente. Debajo de los gráficos aparecerán 4 botones: el primero mostrará la información relativa al cálculo de la probabilidad y las acciones a seguir en función del resultado, el segundo botón muestra la información de los parámetros clínico-patológicos de historial, el tercero mostrará la información del resultado obtenido de la biopsia si el paciente se la ha realizado además de tratamientos recomendados. El cuarto mostrará la información de la prostatectomía si el paciente se ha sometido a la operación. En las Figuras 52 podemos el resultado de realizar una predicción mientras que en la Figura 53 se nos muestra los cuatro botones de información.

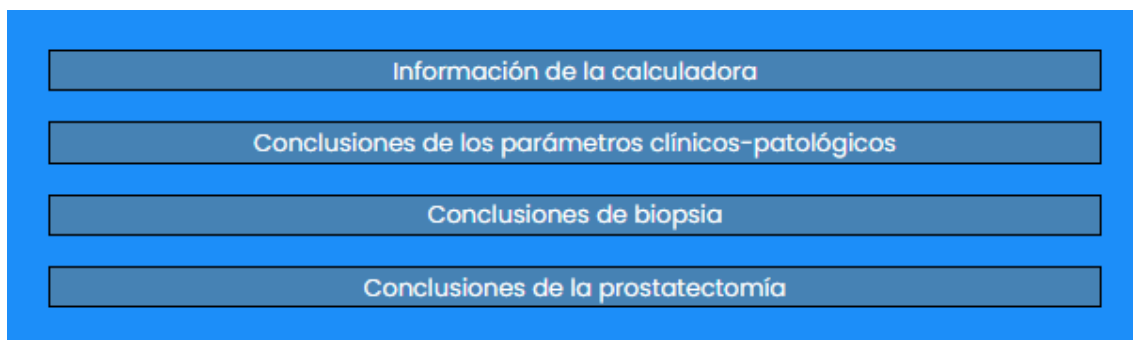


Figura 54 - Botones con la información de la calculadora y el historial