



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería
Informática**

**Aplicación Web para Gestión
de Datos Genómicos de
Cánceres de Pulmón**



Presentado por Carlos Ortúñez Rojo
en Universidad de Burgos — 5 de julio
de 2021

Tutores: Dr. Jesús Manuel Maudes Raedo y
Dr. José Francisco Díez Pastor



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



D. José Francisco Díez Pastor y D. Jesús Manuel Maudes Raedo, profesores del Departamento de Ingeniería Informática, área de Lenguajes y Sistemas Informáticos.

Exponen:

Que el alumno D. Carlos Ortúñez Rojo, con DNI 71565009A, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado Aplicación Web para Gestión de Datos Genómicos de Cánceres de Pulmón.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 5 de julio de 2021

Vº. Bº. del Tutor:

Vº. Bº. del co-tutor:

D. Jesús Manuel Maudes Raedo

D. José Francisco Díez Pastor

Resumen

El presente proyecto trata sobre el desarrollo de una aplicación web que permite la introducción, modificación, eliminación y visualización de los datos genómicos de pacientes con cáncer de pulmón del Hospital Universitario de Burgos. Esta aplicación web también cuenta con un sistema de autenticación y un sistema de gestión de usuarios.

Todos los datos están guardados en una base de datos relacional para permitir un rápido acceso a estos y ahorrar espacio a la hora de almacenarlos.

Dentro de la visualización de los datos se han incluido varias opciones para permitir a los oncólogos una mejor explotación a la hora de realizar sus estudios. Estas opciones de visualización incluyen: la exportación a un fichero *.xlsx*, la representación en diagramas tanto de sectores como de columnas, la obtención de tablas de frecuencias y de percentiles y la visualización individual de los datos crudos de cada paciente.

Para poder comprobar el correcto funcionamiento de toda la parte de visualización nombrada anteriormente, se ha realizado en *Python* un script para poder generar una base de datos sintética. Permitiendo al usuario elegir diferentes parámetros de las distribuciones que se han considerado más óptimas para el tipo de datos con los que se está trabajando.

Descriptores

Aplicación web, base de datos relacional, Laravel, estudio estadístico, visualización, datos médicos.

Abstract

This project consists in the development of a web application that allows the input, modification, elimination and visualization of the genomic data regarding lung cancers cancer patients from the Hospital Universitario de Burgos. In addition, this web application will have an authentication system and a user management system.

All the data is stored in a relational database to allow quick access and to save space when storing the information.

Within the data visualization, several options have been included to offer oncologists a better exploitation when conducting their studies. These display options include exporting to a file *.xlsx*, the representation in diagrams of both, sectors and columns, the obtention of tables than show frequencies and percentiles and the individual visualization of each patient's raw data.

In order to check the correct performance of the entire display part named above, a *Python* script has been created to be able to generate a synthetic database allowing the user to choose different parameters of the distributions that have been considered most optimal for the type of data we are working with.

Keywords

Web application, relational database, Laravel, statistical study, visualization, medical data.

Índice general

Índice general	III
Índice de figuras	V
Índice de tablas	VII
Introducción	1
1.1. Material ajunto	2
Objetivos del proyecto	3
2.1. Objetivos funcionales	3
2.2. Objetivos técnicos	4
2.3. Objetivos personales	5
Conceptos teóricos	7
3.1. Arquitectura MVC	7
3.2. Encriptación simétrica	8
3.3. Estadística	9
Técnicas y herramientas	13
4.1. Técnicas	13
4.2. Herramientas	14
Aspectos relevantes del desarrollo del proyecto	21
5.1. Reuniones con los oncólogos y preparación del proyecto . . .	21
5.2. Migraciones, seeders y Scripts SQL	22
5.3. Gestión de errores	23

5.4. Problemas con la seguridad de los datos sensibles de los pacientes y solución	24
5.5. Investigación sobre el apartado visual de las gráficas y tablas	25
5.6. Pruebas	32
5.7. Fase de refactorización del código	34
5.8. Opinión de los oncólogos respecto a la aplicación final y al trabajo realizado	36
Trabajos relacionados	37
6.1. Aplicación web de gestión de datos de pacientes Ecuador . .	37
6.2. Gestión de información sobre personas discapacitadas	38
6.3. Gestión de datos para el cáncer infantil	38
6.4. Gestión y visualización de datos de enfermos de cáncer . . .	39
Conclusiones y Líneas de trabajo futuras	41
Bibliografía	43

Índice de figuras

3.1. Arquitectura MVC Laravel.	7
3.2. Gráfica de densidad de la distribución normal.	11
3.3. Tres gráficas de la distribución de poisson según λ	11
3.4. Gráfica de una distribución geométrica.	12
5.1. Método up de la migración de la tabla metástasis.	22
5.2. Método <i>run</i> del <i>seeder</i> de los roles.	23
5.3. Interfaz para la selección de divisiones para la realización de gráficas.	25
5.4. Diagrama de sectores STATA.	26
5.5. Tabla de frecuencias STATA.	26
5.6. Diagrama de sectores del proyecto.	26
5.7. Tabla de frecuencias del proyecto.	27
5.8. Gráfica de vela STATA.	27
5.9. Tabla de percentiles STATA.	28
5.10. Gráfica de vela del proyecto.	28
5.11. Tabla de percentiles del proyecto.	29
5.12. Gráfica kaplan meier STATA.	30
5.13. Gráfica kaplan meier del proyecto.	30
5.14. Opciones de filtrado de pacientes.	31
5.15. Gráfica y tabla de frecuencias filtradas por biomarcador NGS y tratamiento quimioterapia.	32
5.16. Resultado de la ejecución de los test de integración	33
5.17. Resultado de la ejecución de los test de seguridad.	34
5.18. Gráfico de issues Codacy.	35
5.19. Tipos de issues Codacy.	35
5.20. Nota de Codacy.	35

5.21. Opinión del oncólogo Enrique Lastra del trabajo realizado en este proyecto.	36
--	----

Índice de tablas

6.1. Comparación entre el proyecto actual y otro similar desarrollado en Ecuador.	37
6.2. Comparación entre el proyecto actual y otro similar para personas con discapacidad.	38
6.3. Comparación entre el proyecto actual y otro similar desarrollado en Castilla La Mancha.	38
6.4. Comparación entre el proyecto actual y otro similar desarrollado en Barcelona.	39

Introducción

El **cáncer** es un conjunto de enfermedades relacionadas las cuales son provocadas por una anomalía en las células que ocasiona que estas empiecen a dividirse continuamente y provoquen un mal funcionamiento de la zona incluso llegando a expandir a otras partes del cuerpo, a esto se le llama **metástasis** [7]. El **cáncer de pulmón** ocurre cuando este problema surge en el tejido pulmonar, este cáncer es el que más muertes provoca actualmente en el mundo, tanto en hombres como en mujeres, siendo el tabaquismo la principal causa de este problema llegando a suponer hasta el 80 % de los casos [1].

Existe gran variedad en cuanto a los diferentes tipos de **cáncer de pulmón**, los **tratamientos** que se llevan a cabo y las **herramientas de detección** de las que se hacen uso. Dicha pluralidad hace que pueda resultar complicada la toma de decisiones durante el proceso de tratamiento del paciente. Además, tratar eficazmente el problema y adaptarse a las circunstancias personales de cada uno de ellos, puede suponer alargar la vida o incluso salvar a la persona. Por ello, realizar una gestión y organización de los datos eficaz, y contar con herramientas que permitan un fácil acceso y visualización, pueden resultar fundamentales en la toma de decisiones de los tratamientos.

En la actualidad el **HUBU** almacena los datos de los pacientes que han sufrido o sufren cáncer de pulmón en formato texto. Este sistema de visualización conlleva a un análisis poco eficaz de los datos, ya que supone un proceso lento y costoso.

Para solventar este problema, el objetivo de este proyecto ha sido crear una **aplicación web** que sirva como **interfaz** para gestionar todos los datos de los pacientes. Estos estarán almacenados dentro de una **base de datos**

relacional y la web tendrá herramientas de **visualización** de los datos como pueden ser gráficas o tablas permitiendo así, un óptimo análisis de dichos datos y facilitando la toma de decisiones durante los procesos de tratamiento.

También debido a que es posible que los médicos que trabajen con esta **aplicación web** no tengan un alto nivel de informática se ha intentado realizar de la manera más **intuitiva** posible, con muchos **controles de errores** y con una comunicación constante con el equipo de oncólogos del Hospital Universitario de Burgos para lograr la mejor versión posible.

1.1. Material ajunto

En el proyecto como material adjunto se incluye:

- Memoria
- Anexos
- Aplicación web en *Laravel*.
- Scripts *SQL* para generar la base de datos e insertar datos necesarios para la aplicación.
- Script en *Python* para generar la base de datos sintética y generar las gráficas kaplan meier.

Todos los elementos desarrollados durante el proyecto se pueden ver en el repositorio de *GitHub*: https://github.com/CarlosOrtu/TFG_HUBU

Se puede acceder a la aplicación web del proyecto en la siguiente url: <http://tfg-hubu.herokuapp.com/pacientes>. Con el usuario "administrador@gmail.com" y la contraseña "UniversidadDeBurgos".

Objetivos del proyecto

En este apartado se van a especificar los diferentes objetivos, tanto funcionales como técnicos como personales.

2.1. Objetivos funcionales

Los objetivos funcionales que se acordaron con los oncólogos y se han tratado de conseguir con este proyecto son:

- Integrar un inicio de sesión para controlar el acceso a la aplicación web.
- Crear un sistema de gestión de usuarios en el cual los administradores puedan crear, modificar y eliminar usuarios.
- Realizar un sistema de gestión de datos personales donde se permita cambiar tanto la contraseña como el correo del propio usuario.
- Establecer un sistema de gestión de pacientes donde se pueda crear, eliminar y modificar pacientes y todos sus datos de una manera intuitiva y visual.
- Publicar la aplicación web en una plataforma en la nube para permitir a los oncólogos empezar a introducir datos.
- Facilitar la visualización y análisis de los datos con un sistema de gráficas y tablas.
- Realizar un filtrado de pacientes con los atributos que se consideren más importantes.

- Automatizar la creación de una base de datos sintética en segundo plano, que permita distintos tipos de distribuciones estadísticas para poder realizar pruebas y comprobar el correcto funcionamiento de las herramientas nombradas anteriormente como la visualización de los datos en gráficas.

2.2. Objetivos técnicos

Los objetivos técnicos que han sido necesario para llevar a cabo este proyecto son:

- Usar *Laravel* para el *back-end* y todas las herramientas que este framework incluye.
- Emplear *Bootstrap*, *CSS*, *HTML*, *JavaScript* y *jQuery* para el *front-end* de la web y realizar el diseño de está lo mas funcional e intuitivo posible.
- Gestionar bases de datos *MySQL* desde *PhpMyAdmin*.
- Recordar el lenguaje *SQL* para la realización de scripts que permitan crear la base de datos e introducir los datos necesarios para el correcto funcionamiento de la web.
- Usar la librería *Google Charts* para la representación de las diferentes gráficas.
- Utilizar *Python* y alguna librería de este lenguaje para la creación de la base de datos sintética que se ejecute en segundo plano.
- Utilizar *pandas* para el procesamiento de los datos de los pacientes para permitir la visualización de gráficas de supervivencia.
- Usar *GitHub* para permitir un control de versiones sobre el proyecto.
- Realizar el desarrollo del proyecto siguiendo la metodología *SCRUM* sobre el proyecto y usar *ZenHub* para permitir una mejor gestión de las tareas y del tiempo de cada una de estas.

2.3. Objetivos personales

- Ayudar al departamento de oncología del Hospital Universitario de Burgos.
- Profundizar en los campos del desarrollo y diseño web.
- Poner en práctica todo lo aprendido durante estos cuatro años de carrera.
- Acabar mi ciclo formativo en la universidad con un proyecto funcional y que tenga una utilidad real.
- Aprender a usar L^AT_EX para una mejorar la documentación de mis futuros proyectos.

Conceptos teóricos

En este apartado se van a tratar los conceptos teóricos más importantes del proyecto, empezando con la arquitectura del framework de back-end, siguiendo con la seguridad y acabando con los conceptos estadísticos.

3.1. Arquitectura MVC

La arquitectura Modelo-Vista-Controlador sirve para separar el código dependiendo de la funcionalidad de este, así conseguir lograr 3 capas con un trabajo específico en cada una de ellas[10]. Actualmente algunos frameworks como Laravel integran este tipo de arquitectura pero añadiendo capas extra como el enrutado y los middleware como podemos ver en la imagen 3.1.

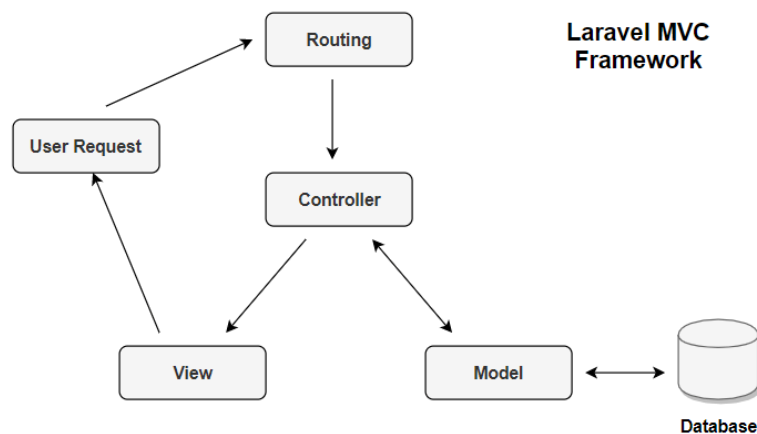


Figura 3.1: Arquitectura MVC Laravel.

Modelo

Es la capa que se encarga de interactuar con los datos, por lo tanto tendrá que tener la capacidad de añadir, modificar y eliminar los datos que sean necesarios. Lo normal suele ser usar alguna librería o algún mecanismo que incluya el framework para no tener que trabajar directamente con consultas SQL y permitir trabajar con objetos y clases lo que supone una abstracción de base de datos y persistencia en objetos[10].

Vista

Las vistas son la capa que va a servir como interfaz al usuario y le va a permitir interactuar y visualizar los datos sin trabajar directamente con ellos ya que de esto se encargan las otras dos capas. Las vistas suelen estar escritas en HTML y CSS principalmente, pudiendo incluir otros lenguajes como JavaScript o PHP e incluso procesadores de plantillas como Blade Templates de Laravel [10].

Controlador

Los controladores son la capa encargada de implementar la lógica de negocio que va a actuar como enlace entre las acciones que realice el usuario en la vista y los modelos. Pueden estar escritos en muchos lenguajes como PHP, Python, NodeJS, Java, C# o Ruby [10].

3.2. Encriptación simétrica

La encriptación de datos es un método mediante el cual un conjunto de caracteres se modifican para convertirlos en otros completamente diferentes de los cuales no se podrá volver a obtener el conjunto original a no ser que se disponga de la clave o las claves necesarias. Existen dos tipos diferentes de encriptación: simétrica y asimétrica[9].

La encriptación simétrica se basa en lograr tanto el cifrado como el descifrado mediante el uso de una sola clave, por lo tanto la seguridad de este método depende en la capacidad que se va a tener para que una persona no autorizada no tenga acceso a esta clave[28]. Estos algoritmos tienen dos tipos dependiendo de la división que se haga del mensaje original, cifrado de bloque y cifrado de flujo[14].

Cifrado de bloque

En el cifrado de bloque el mensaje original se divide en bloques de un tamaño fijo de bits, normalmente suelen ser de un tamaño 64 o 128 bits, estos bloques se encriptan de manera fija para todos sus elementos. El inconveniente de este tipo de cifrado es que si existen secuencias repetidas de caracteres en el texto original, también existirán estos caracteres repetidos en el texto encriptado[3]. Existen diferentes esquemas de cifrado dentro de los cifrados de bloque, los más importantes son DES, 3-DES, RC2, RC5, RC6 y AES[28].

AES

El algoritmo AES (*Advanced Encryption Standar*) agrupa el mensaje original en bloques de longitud variable, aunque el estándar fija el tamaño en 128 bits y la representación de estos bloques sería una matriz de 4x4 bytes. Lo mismo ocurre con la clave, tiene tres posibles longitudes 128, 192 y 256 y en el caso del AES-128, la contraseña se almacena de la misma manera que cada segmento del mensaje, en una matriz de 4x4. A parte de esta clave inicial, se generan un conjunto de 10 subclaves, una por cada iteración del algoritmo. Cada una de estas iteraciones realiza 4 operaciones básicas SubBytes, ShiftRows, MixColumns y AddRoundKey y para descifrar se realizan las mismas iteraciones y los mismos algoritmos pero de forma inversa[26].

3.3. Estadística

Conceptos estadísticos

Media: Es una medida de tendencia que nos indica el punto medio de un conjunto de datos[4].

$$Media(x) = \bar{x} = \frac{\sum_{i=1}^n x_i}{n} \quad (3.1)$$

Varianza: Es una medida de dispersión que nos indica la distancia de un conjunto de datos a su media[6].

$$Varianza(x) = \sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}} \quad (3.2)$$

Desviación: Es la varianza al cuadrado[6].

$$Desviación(x) = \sigma^2 = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1} \quad (3.3)$$

Percentil: Es una medida de posición que indica el porcentaje de datos que están por debajo de un valor[2].

Oblicuidad: Es una medida que nos indica que tan asimétrica es una distribución respecto a su media[5].

$$Oblicuidad(x) = \frac{\sum_{i=1}^n (x_i - \bar{x})^3}{(n - 1) \cdot S^3} \quad (3.4)$$

Curtosis: Es una medida que según aumenta su valor nos indica que existe una concentración de valores tanto cerca de la media como muy lejos de ella, al tiempo que en los valores intermedios disminuye esta concentración[30].

$$Curtosis(x) = \frac{\sum_{i=1}^n (x_i - \bar{x})^4}{(n - 1) \cdot S^4} \quad (3.5)$$

Corrección de Bessel: La corrección de Bessel se usa para corregir el sesgo estadístico cuando no se está trabajando con la población entera sino que se está trabajando con una muestra y consiste en sustituir n en el divisor por $n-1$, como he realizado en todos los conceptos anteriormente explicados[29].

Distribuciones de probabilidad

Las distribuciones son funciones que indican la probabilidad que ocurra cierto suceso. Según sea el tipo de variable continua o discreta se elegirá un tipo de distribución u otro.

Distribución normal: Es una distribución de valores continuos definida por dos parámetros, la media μ y la desviación σ , la función de densidad es simétrica respecto a su media y dependiendo de la desviación va a tener una anchura determinada[8].

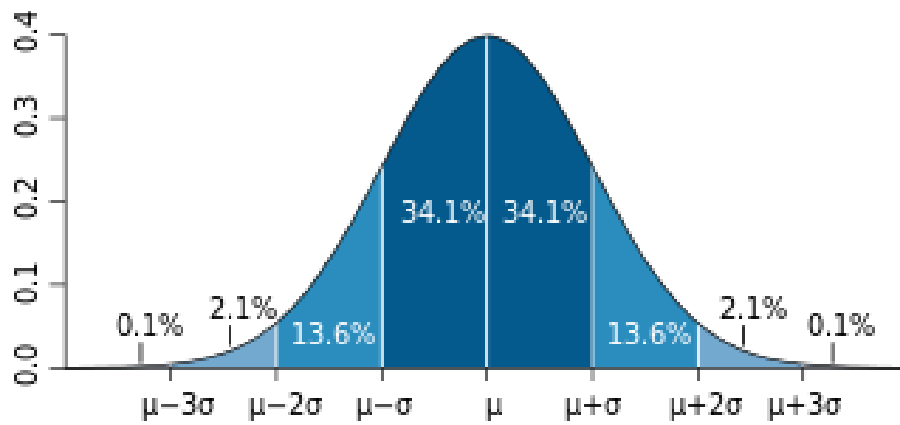


Figura 3.2: Gráfica de densidad de la distribución normal.

Distribución de poisson: Es una distribución de valores discretos la cual está definida por un parámetro λ , y este parámetro representará el valor que es más probable de obtener dentro de la gráfica.

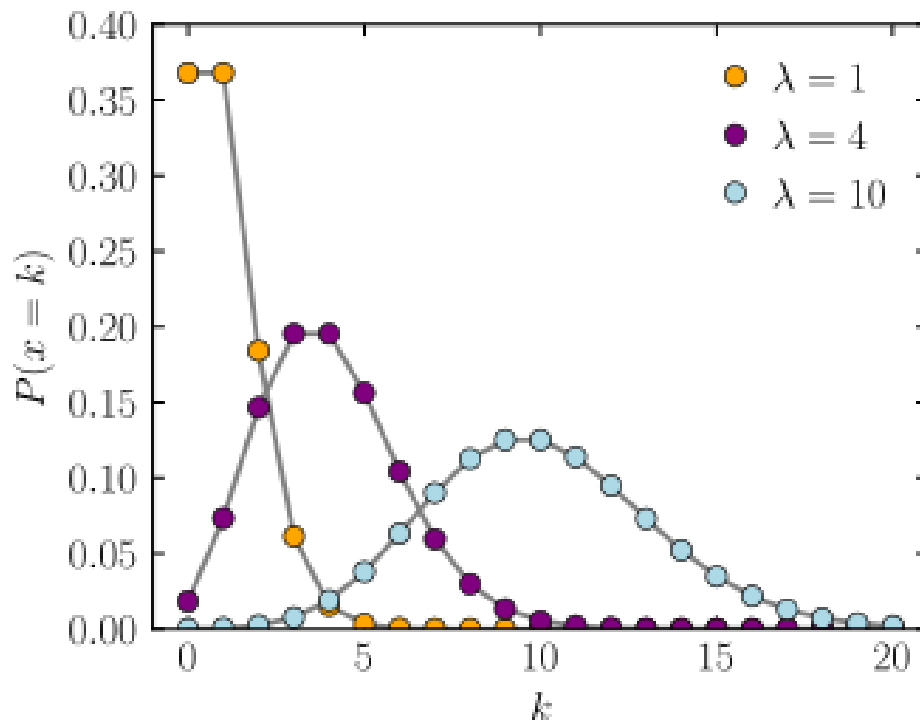


Figura 3.3: Tres gráficas de la distribución de poisson según λ .

Distribución geométrica: Es una distribución de valores discretos la cual esta definida por un parámetro p , y este parámetro indica la probabilidad de obtener el resultado.

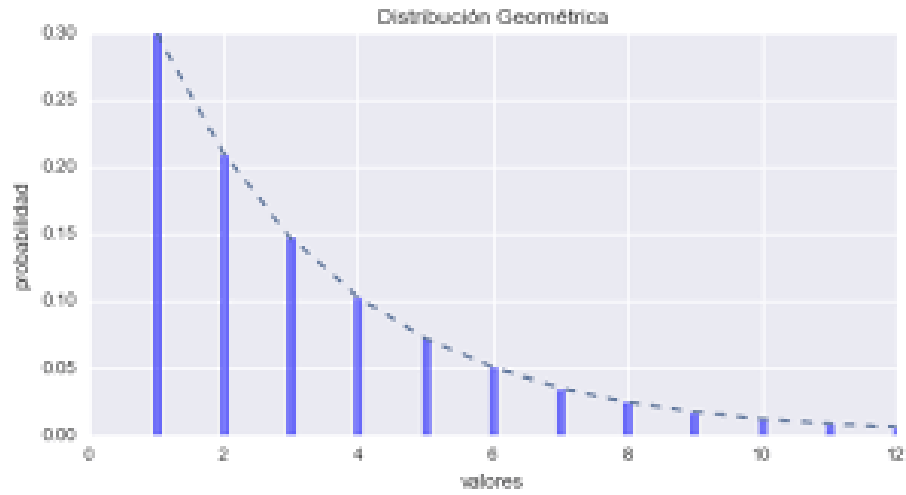


Figura 3.4: Gráfica de una distribución geométrica.

Técnicas y herramientas

4.1. Técnicas

En este apartado se va a explicar la técnica de desarrollo de software que ha seguido y una explicación de porque se ha optado por esta opción y como se ha adaptado al proyecto.

Scrum

Las metodologías ágiles son aquellas que se basan en dar una mayor libertad a la colaboración con el cliente y a la realización del proyecto en iteraciones. Según el manifiesto ágil esta nueva metodología se basa en cuatro principios: Priorizar al individuo y a las iteraciones sobre el proceso y las herramientas, el desarrollo del software sobre la documentación, colaboración con el cliente antes que un contrato fijo y la capacidad de responder a los cambios antes que un plan fijo[17].

Debido a las características del proyecto se decidió que una metodología ágil era lo más adecuado para llevar a cabo su desarrollo ya que se comenzó con una idea general, por lo tanto a la hora de ir realizando el proyecto iba a ser necesario realizar cambios sobre las partes ya finalizadas y en vista de que a medida que avanzará el proyecto se iban a ir añadiendo funcionalidades, por lo tanto no se puede establecer un plan fijo desde el comienzo[17].

De todas las metodologías ágiles para desarrollar el proyecto se eligió Scrum. Este método se basa en el desarrollo de sprints, en los cuales se realizan partes del proyecto final ya funcionales y reuniones periódicas. Estos sprints suelen durar entre 1 ó 2 semanas[16]. Esta metodología está pensada para trabajar con equipos enteros de desarrollo dentro de una empresa, pero

para este proyecto se ha adaptado. Los roles quedan divididos de la manera que el *product owner* son los oncólogos, el *scrumMaster* son ambos tutores y el equipo de desarrollo soy yo, las reuniones y las duraciones de los sprint también han sido levemente modificadas haciéndolas menos estrictas debido a que no siempre era posible para los oncólogos realizar una reunión a la finalización de los sprint.

4.2. Herramientas

En este apartado se van a explicar todas las herramientas usadas para llevar a cabo cada parte del proyecto y se van a nombrar las diferentes opciones que se tuvieron en cuenta .

Infraestructura de desarrollo local

WAMP

- Herramientas consideradas: [MAMP¹](#), [XAMPP²](#), [WAMP³](#).
- Herramienta elegida: [WAMP³](#).

[WAMP³](#) es un entorno de desarrollo web local pensando para usarse en Windows el cual contiene Apache, MySQL, PHP y el administrador de bases de datos PhpMyAdmin. Por lo tanto este entorno cumple con los objetivos técnicos que se tenían para el desarrollo de la aplicación web.

Host web

Heroku

- Herramientas consideradas: [Ionos⁴](#), [AWS amazon⁵](#), [Heroku⁶](#).
- Herramienta elegida: [Heroku⁶](#).

[Heroku⁶](#) es una plataforma de computación en la nube que da soporte a varios lenguajes como Python, Java y PHP[21]. También cuenta con una serie

¹<https://www.mamp.info/en/windows/>

²<https://www.apachefriends.org/es/index.html>

³<https://www.wampserver.com/en/>

⁴<https://www.ionos.es/alojamiento/alojamiento-web>

⁵<https://aws.amazon.com/es/ec2/dedicated-hosts/>

⁶<https://www.heroku.com/>

de extensiones que resultaban interesantes para nuestro proyecto, estas son, una base de datos MySQL que permite la gestión desde MySQL Workbench y un protocolo HTTPS para implementar a nuestra web. También para aplicaciones pequeñas como la que se está desarrollando en este trabajo el host resulta gratuito y la política de privacidad de datos concordaba con los requisitos necesarios para poder alojar los datos de los pacientes.

Back-end

Laravel

- Herramientas consideradas: [django](#)⁷, [.NET](#)⁸, [Laravel](#)⁹.
- Herramienta elegida: [Laravel](#)⁹.

[Laravel](#)⁹ es un framework de PHP que se basa en la arquitectura MVC (3.1) y tiene como objetivo usar una sintaxis elegante, permitiendo crear código de manera sencilla e implementando muchas funcionalidades[24]. Tiene varias utilidades que hemos utilizado dentro del proyecto, por ejemplo las plantillas Blade las cuales permiten usar código PHP e incluir lógica dentro de las vistas de manera muy sencilla y cómoda y también Eloquent ORM el cual nos permite una manera sencilla de manipular los elementos de una base de datos creando un modelo por cada una de las tablas de la base de datos y esto nos permite manipular estos modelos sin tener que ejecutar sentencias SQL.

Gestor de paquetes

Composer

[Composer](#)¹⁰ es un gestor de paquetes para PHP, se usa para poder integrar librerías de terceros en tu proyecto de una manera muy sencilla. Para este proyecto se ha utilizado para incorporar las librerías que se han usado de PHP y Laravel (4.2) y también el propio Laravel (4.2).

⁷<https://www.djangoproject.com/>

⁸<https://docs.microsoft.com/es-es/dotnet/>

⁹<https://laravel.com/>

¹⁰<https://getcomposer.org/>

Front-end

Bootstrap

- Herramientas consideradas: [Bootstrap¹¹](#), [Tailwind¹²](#).
- Herramienta elegida: [Bootstrap¹¹](#).

[Bootstrap¹¹](#) es una librería de HTML, CSS y JavaScript que permite dar estilo a una página web de manera muy sencilla. Esta librería funciona de manera que asignando clases determinadas a los elementos HTML esos elementos van a tener un formato, una posición o un estilo determinado.

jQuery

[jQuery¹³](#) es una librería de JavaScript la cual nos permite manipular los elementos HTML, añadir animaciones o ejecutar peticiones Ajax de una manera mucho más rápida y sencilla que si lo hiciéramos directamente con Vanilla JavaScript.

Base de datos

MySQL

- Herramientas consideradas: [MySQL¹⁴](#), [PostgreSQL¹⁵](#).
- Herramienta elegida: [MySQL¹⁴](#).

[MySQL¹⁴](#) es un sistema gestor de bases de datos relacionales. Se decidió usar MySQL en lugar de PostgreSQL debido al rendimiento de ambos gestores de bases de datos en una aplicación Laravel, como se puede comprobar en la siguiente web ([Comparación MySQL y PostgreSQL¹⁶](#)) MySQL tiene bastante mejor rendimiento.

¹¹<https://getbootstrap.com/>

¹²<https://tailwindcss.com/>

¹³<https://jquery.com/>

¹⁴<https://www.mysql.com/>

¹⁵<https://www.postgresql.org/>

¹⁶<https://medium.com/web-developer/mysql-vs-postgresql-performance-test-with-laravel-api-for-simple-eloquent-queries-on-1-million-6e0e6f1005b8>

PhpMyAdmin

PhpMyAdmin¹⁷ es una herramienta de gestión de bases de datos MySQL escrita en PHP. Permite ejecutar sentencias SQL mediante una interfaz gráfica muy intuitiva y sencilla de utilizar.

MySQL Workbench

MySQL Workbench¹⁸ es la herramienta oficial de gestión bases MySQL, es similar a PhpMyAdmin a diferencia que esta herramienta tiene una aplicación de escritorio. En principio no se tenía pensado trabajar con esta herramienta, pero a la hora conectar una base de datos a la aplicación web subida en Heroku era necesario usar MySQL Workbench para administrar esta base de datos.

Documentación

L^AT_EX

- Herramientas consideradas: **Open Office**¹⁹, **L^AT_EX**²⁰.
- Herramienta elegida: **L^AT_EX**²⁰.

L^AT_EX²⁰ es un sistema de composición de textos de alta calidad y su principal uso es para artículos científicos.

Se decidió usar **L^AT_EX** por encima de Open Office debido a que se vio como una oportunidad de aprender una herramienta nueva que podía llegar a ser útil para futuros trabajos.

Editores de texto

Overleaf

Overleaf²¹ es un editor online de **L^AT_EX** que permite ir compilando el documento y ver lo cambios al instante, lo cual agiliza mucho el trabajo a la hora de redactar documentos.

¹⁷<https://www.phpmyadmin.net/>

¹⁸<https://www.mysql.com/products/workbench/>

¹⁹<https://www.openoffice.org/es/>

²⁰<https://www.latex-project.org/>

²¹<https://es.overleaf.com/>

Sublime Text 3

- Herramientas consideradas: [Visual Studio Code](#)²², [Sublime Text 3](#)²³.
- Herramienta elegida: [Sublime Text 3](#)²³.

[Sublime Text 3](#)²³ es un editor de texto cómodo, sencillo y fácil de usar, viene incluido con la herramienta Kite que sirve para autocompletar código en distintos lenguajes. Para este proyecto se ha utilizado este editor para redactar todo el código del proyecto, el cual incluye la aplicación web en Laravel, el script Python para hacer la base de datos sintética y los script SQL. Se ha elegido Sublime Text 3 sobre Visual Studio Code debido a que ya había estado usando este editor de texto antes y por lo tanto estaba más familiarizado con el.

Gestión del proyecto

GitHub

[GitHub](#)²⁴ es una herramienta que sirve para subir, almacenar código en la nube y también llevar un control de versiones. GitHub se ha utilizado para llevar a cabo la metodología Scrum gracias a las milestones las cuales he usado para simular los sprints y las issues las cuales son todas las tareas que se han ido realizando a lo largo del proyecto.

ZenHub

[ZenHub](#)²⁵ es una herramienta de gestión de proyecto la cual se integra en GitHub. ZenHub nos permite organizar las issues en pipelines y establecer cuanto tiempo se va a tardar en realizar cada una de las tareas.

Análisis de la calidad del código

Codacy

[Codacy](#)²⁶ es una herramienta de análisis de código que permite detectar problemas en el código como problemas de seguridad, código repetido o código sin usar.

²²<https://code.visualstudio.com/>

²³<https://www.sublimetext.com/3>

²⁴<https://github.com/>

²⁵<https://www.zenhub.com/>

²⁶<https://www.codacy.com>

Librerías

Numpy

Librería de Python que permite crear vectores y matrices y muchas funciones para trabajar con estos. Se ha utilizado en el proyecto para la parte estadística de la creación de la base de datos sintética.

mysql.connector

Librería de Python que permite conectar una base de datos MySQL y realizar operaciones SQL sobre esta. Se ha utilizado en el proyecto para insertar los datos en la base de datos sintética.

phpunit

Framework de PHP que permite crear pruebas unitarias. Se ha utilizado para realizar las pruebas unitarias de la aplicación web.

faker

Librería de Python que permite crear nombre, apellidos y fechas aleatorias. Se ha utilizado para la creación de la base de datos sintética.

laravel/ui

Paquete de autenticación de Laravel. Se ha utilizado para el login de la aplicación web.

maatwebsite/excel

Librería de Laravel que permite exportar e importar datos de un fichero Excel. se ha utilizado para poder exportar la base de datos a un fichero en Excel.

openssl

Librería de PHP que permite la encriptación de datos. Se ha utilizado para encriptar los datos sensibles de los pacientes.

Google Charts

- Herramientas consideradas: [Chart.js](#), [Google Charts](#).

- Herramienta elegida: **Google Charts**.

Librería de JavaScript que permite la visualización de datos en gráficas interactivas. Se he utilizado para todo el apartado de visualización de datos.

DataTables

Librería de JavaScript que permite la creación de tablas interactivas. Se ha utilizado para todas las tablas que hay implementadas en la web.

pandas

Librería de python que permite una sencilla manipulación de conjuntos de datos. Se ha utilizado para el procesamiento de los datos necesarios para dibujar las gráficas kaplan meier.

lifelines

Librería de python que permite realizar distintos analisis de supervivencia. Se ha utilizar para dibujar las graficas kaplan meier.

Aspectos relevantes del desarrollo del proyecto

5.1. Reuniones con los oncólogos y preparación del proyecto

Al principio del proyecto no se conocía muy bien cuál iba a ser el alcance de este ya que solo se sabía el propósito general, una aplicación web que permitiera la gestión de los datos de los pacientes con cáncer de pulmón y que facilitará una mejor visualización de estos. Pero como el desarrollo web era un campo que me llamaba mucho la atención y el hecho que mi proyecto fuera a tener una utilidad real y encima en un campo como la medicina que cualquier mejora o avance puede suponer mejorar la calidad de vida de muchas personas, fue motivo suficiente para elegir este trabajo.

Las dos primeras semanas de proyecto se basaron única y exclusivamente en realizar reuniones con los oncólogos Raquel Gómez Bravo y Enrique Lastra Aras para recoger las historias de usuario del proyecto, entender los datos que se iban a recoger y valorar las distintas posibilidades que había para crear la estructura de la base de datos relacional, adaptar el diseño a lo que ellos buscaban y elegir las herramientas que me fueran a permitir llevar a cabo este proyecto.

Las primeras reuniones fueron complicadas de afrontar debido a que era la primera vez que se trataba con clientes reales y la primera vez que aplicaba los conocimientos adquiridos de ingeniería de software en un escenario real, pero gracias a la colaboración de los oncólogos y a la ayuda de los tutores pudimos acabar entendiéndonos perfectamente y redactando lo que sería el comienzo del proyecto.

5.2. Migraciones, seeders y Scripts SQL

Al comienzo del desarrollo de la aplicación web se empezó a trabajar en local con el entorno de desarrollo WAMP (4.2), pero con la perspectiva de que en el futuro este proyecto iba a tener que ser integrado tanto en un host online como probablemente en el servidor del HUBU. Por lo tanto no solo iba a tener que realizar la aplicación si no también una manera de poder migrar tanto esta como la base de datos. Esta fue una de las razones por las cuales se selecciono Laravel (4.2) ya que se conocía que tenía dos herramientas para realizar esto de manera sencilla.

La primera herramienta son las migraciones, Laravel nos permite crear clases que hereden de *Migration* con dos funciones, la función *up* la cual sirve para crear la tabla con sus correspondientes campos y referencias (5.1) y la función *down* que elimina la tabla. Para ejecutar todas las migraciones y crear la estructuras de tablas completa solo hace falta ejecutar el comando *php artisan migrate* en la dirección en la que este instalado nuestro proyecto Laravel [22].

```
public function up()
{
    Schema::create('metastasis', function (Blueprint $table) {
        $table->id('id_metastasis');
        $table->unsignedBigInteger('id_enfermedad');
        $table->string('tipo');
        $table->foreign('id_enfermedad')->references('id_enfermedad')->on('enfermedades')
            ->onDelete('cascade')->onUpdate('cascade');
    });
}
```

Figura 5.1: Método up de la migración de la tabla metástasis.

La segunda herramienta son los *seeders*, Laravel nos permite insertar datos en las tablas ya creadas creando clases que hereden de *Seeder*, estas clases tendrán un solo método llamado *run* el cual insertará los datos dentro de la tabla especificada (5.2). Para ejecutar todos los *seeders* y introducir los datos solo hace falta ejecutar el comando *php artisan db:seed* en la misma dirección que las migraciones [23].

```
public function run()
{
    DB::table('roles')->insert([
        'id_rol' => '1',
        'descripcion' => 'Administrador del sistema',
    ]);

    DB::table('roles')->insert([
        'id_rol' => '2',
        'descripcion' => 'Oncólogo',
    ]);
}
```

Figura 5.2: Método *run* del *seeder* de los roles.

Por último también se han realizado scripts en SQL para realizar lo mismo que he comentado anteriormente en las migraciones y en los *seeders*, el motivo de esto es para que en el caso que estas herramientas fallen se pueda seguir poniendo a punto la base de datos ejecutando los scripts SQL.

5.3. Gestión de errores

Para este trabajo se han tenido muy en cuenta dos factores, el primero que los médicos que vayan a usar la aplicación es muy probable que no estén familiarizados con la informática y el segundo que a la hora de realizar las representaciones estadísticas tener dos datos iguales pero escritos de manera diferente podía suponer un problema ya que se iban a tomar como datos diferentes.

Para solventar este problema se ha intentado que la mayoría de los formularios de entrada de datos sean desplegados, así eliminando el error humano por completo, y en los campos que esto no era posible, añadir validadores que implementa Laravel para poder comprobar que estos datos están bien introducidos, por ejemplo que la fecha del diagnóstico no pueda ser anterior a la fecha de la primera consulta.

5.4. Problemas con la seguridad de los datos sensibles de los pacientes y solución

Una vez completada la primera parte del proyecto, la cual supuso todo el desarrollo de la gestión de usuarios y de pacientes, se pensó que la mejor manera de que los oncólogos fueran usando la aplicación web y buscando tanto errores como posibles mejoras, era subir esta a un host online. Esto en principio no resultó ningún problema, hasta que nos dimos cuenta que podíamos estar vulnerando la ley de privacidad de datos al publicar datos de pacientes sin su consentimiento en servidores externos al hospital. Para solventar este problema se tomaron tres medidas de seguridad.

La primera fue buscar un host online cuyos servidores estuvieran en Europa para que siguiera la misma ley de protección de datos que en España y que este tuviera una política de protección de datos de acuerdo a lo que nosotros necesitábamos. Encontramos Heroku (4.2), el cual es un host de páginas web en la nube con servidores localizados en Irlanda [20] y la política de privacidad de los datos de esta web dice que su personal no puede acceder ni interactuar con los datos alojados en sus servidores a no ser que se pida permiso previo al usuario o por mandato del gobierno [19]. Por lo tanto se seleccionó este host ya que se vio que cumplía a la perfección con los requisitos de privacidad que teníamos.

La segunda medida fue cifrar los datos sensibles mediante una encriptación simétrica 3.2. Para llevar a cabo esta tarea se usó la librería openssl 4.2 y se creó una clase con 2 métodos, el primero se encarga de encriptar los datos cuando estos son introducidos o modificados en la base de datos y el segundo se encarga de desencriptar los datos cuando estos van a ser visualizados por el usuario en la web.

La tercera medida fue el protocolo HTTPS ya que la única brecha de seguridad que quedaba en la aplicación era que al usar el protocolo HTTP se podía acceder e interceptar las peticiones que se realizaban desde el cliente al servidor y por lo tanto tener acceso tanto a las contraseñas de los usuarios como a los datos sensibles de los pacientes. Para esto se consiguió un certificado gratuito SSL [27]. Pero después de conseguir este certificado apareció otro problema, la versión gratuita de Heroku no permitía integrar un certificado SSL externo y por lo tanto tuve que pasar la cuenta a la versión hobby por un precio de 7\$ para poder integrar el protocolo HTTPS dentro de la web.

5.5. Investigación sobre el apartado visual de las gráficas y tablas

Una de las incertidumbres que surgió a la hora de empezar con el apartado de la visualización de los datos era que no sabía muy bien ni como realizar la interfaz ni como representar los datos obtenidos. Para lo primero, después de darle vueltas se decidió realizar un diseño sencillo el cual permitiera las divisiones según el tipo de dato como se puede ver en la imagen 5.3.

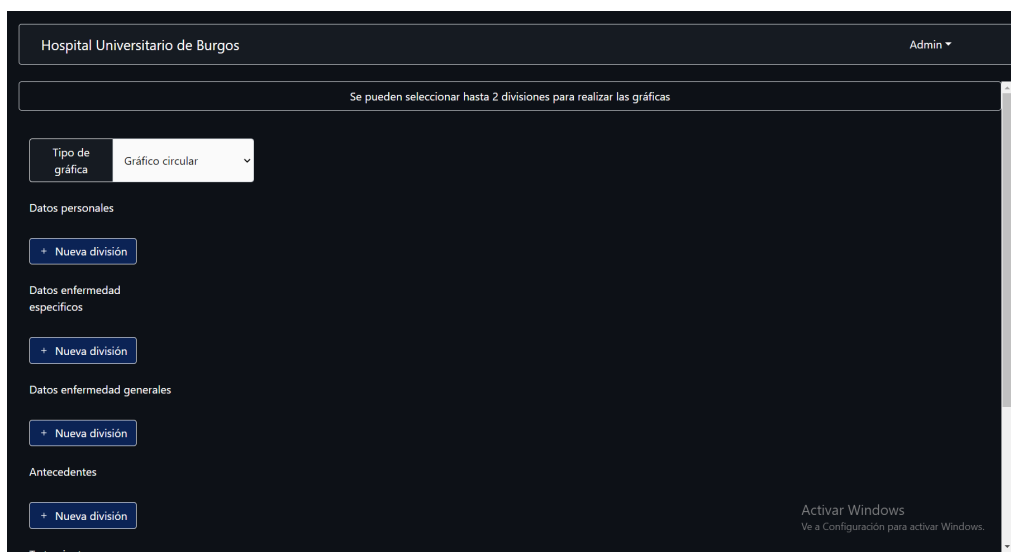


Figura 5.3: Interfaz para la selección de divisiones para la realización de gráficas.

Para la visualización de las gráficas se investigó sobre diferentes programas estadísticos y al final se decidió por realizar las tablas y gráficas similares al programa STATA²⁷.

Como se puede observar en las siguientes imágenes, tenemos por un lado tanto el diagrama de sectores 5.4 como la tabla de frecuencias de STATA 5.5 y el diagrama de sectores 5.6 y la tabla de frecuencias de nuestra aplicación web 5.7.

²⁷<https://www.stata.com/>

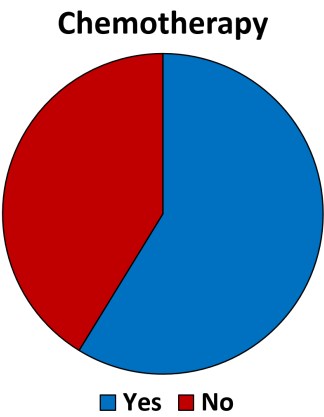


Figura 5.4: Diagrama de sectores STATA.

. tabulate SEX			
Sexo	Freq.	Percent	Cum.
Mujer	241	36.40	36.40
Hombre	421	63.60	100.00
Total	662	100.00	

Figura 5.5: Tabla de frecuencias STATA.

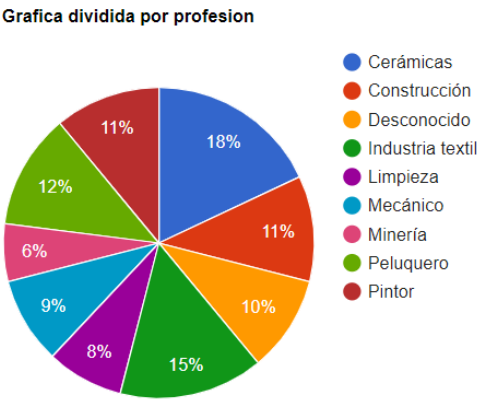


Figura 5.6: Diagrama de sectores del proyecto.

5.5. INVESTIGACIÓN SOBRE EL APARTADO VISUAL DE LAS GRÁFICAS Y TABLAS

27

profesion	Frecuencia	Porcentaje	Acumulada
Cerámicas	18	18	18
Construcción	11	11	29
Desconocido	10	10	39
Industria textil	15	15	54
Limpieza	8	8	62
Mecánico	9	9	71
Minería	6	6	77
Peluquero	12	12	89
Pintor	11	11	100
Total	100	100	

Figura 5.7: Tabla de frecuencias del proyecto.

Para los atributos numéricos se pidió otro tipo de gráficas y tablas para mejorar el análisis, ambas se basan en los percentiles (3.3) y los conceptos teóricos explicado en el apartado 3.3. En las siguiente imágenes podemos observar tanto la gráfica de vela 5.8 como la tabla de percentiles de STATA 5.9 y la gráfica de vela 5.10 y la tabla de percentiles de nuestra aplicación web 5.11.

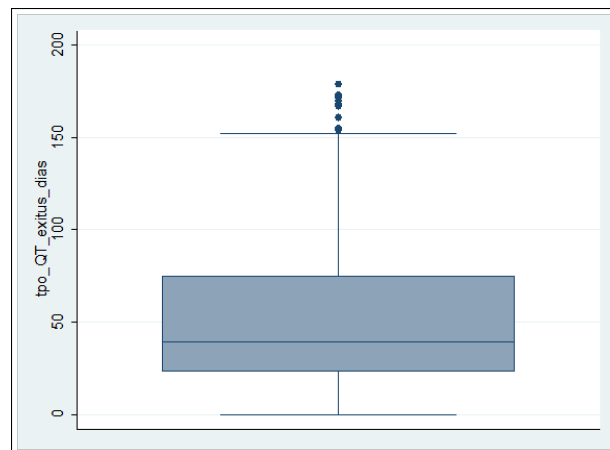


Figura 5.8: Gráfica de vela STATA.

```
. summarize tpo_QT_exitus_dias, detail
```

tpo_QT_exitus_dias					
Percentiles		Smallest			
1%	0	0			
5%	5	0			
10%	12	0	Obs		387
25%	23	0	Sum of Wgt.		387
50%	39		Mean		53.97158
		Largest	Std. Dev.		41.85686
75%	75	170			
90%	122	172	Variance		1751.997
95%	138	173	Skewness		.9958483
99%	170	179	Kurtosis		3.109361

Figura 5.9: Tabla de percentiles STATA.

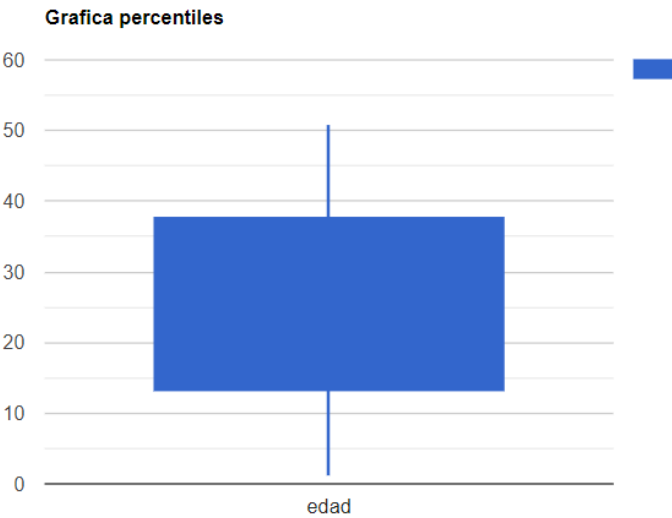


Figura 5.10: Gráfica de vela del proyecto.

	Percentiles	Smallest		
1%	1			
5%	3			
10%	5		Obs	100
25%	13			
50%	27		Mean	26.3
		Largest	Std. Dev.s	14.346716796127
75%	38			
90%	44		Variance	205.82828282828
95%	49		Skewness	-0.09473801472216
99%	51		Kurtosis	1.8692361172061

Figura 5.11: Tabla de percentiles del proyecto.

Como último tipo de gráfica se pidió la kaplan meier, la cual sirve para realizar un análisis de la supervivencia de los pacientes [31]. En las siguientes imágenes podemos observar la gráfica kaplan meier de STATA 5.12 y la gráfica kaplan meier de nuestra aplicación web 5.13.

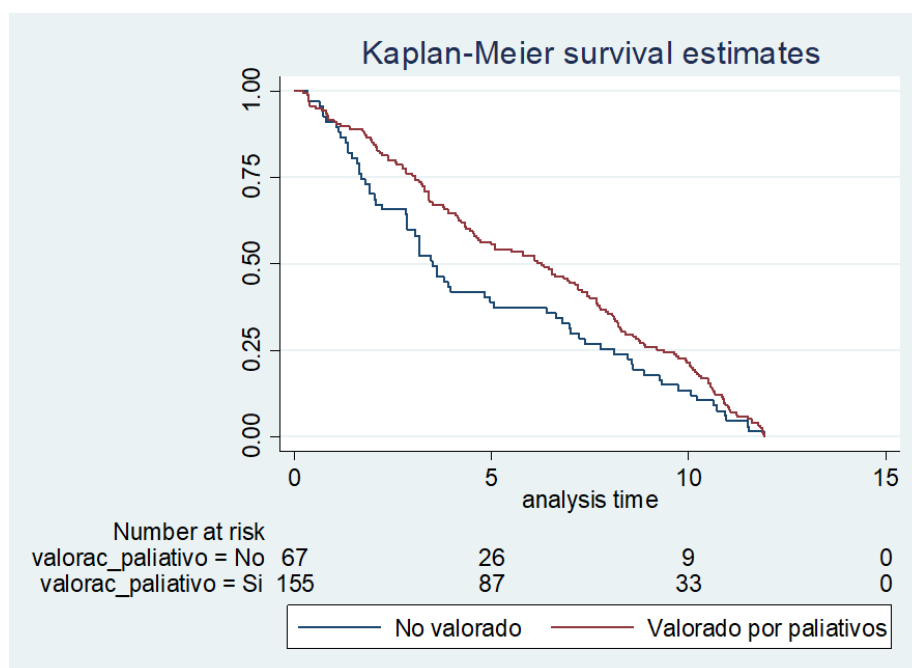


Figura 5.12: Gráfica kaplan meier STATA.

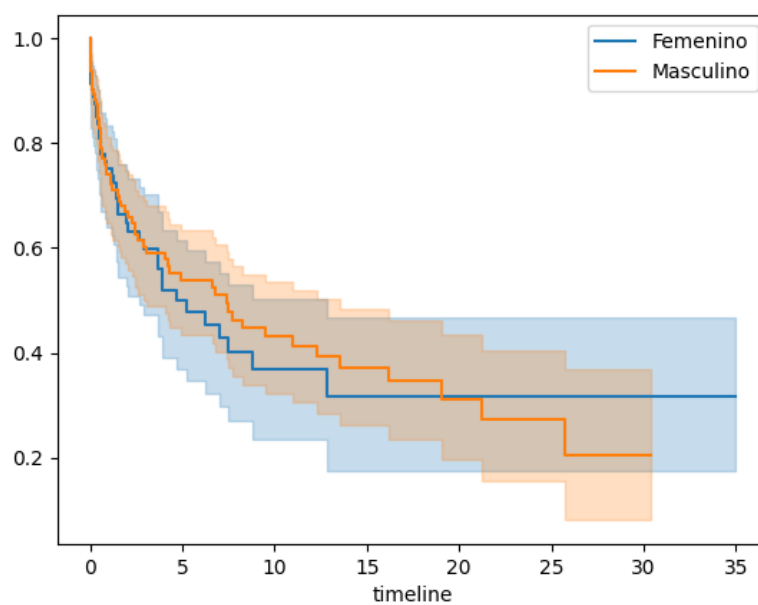


Figura 5.13: Gráfica kaplan meier del proyecto.

5.5. INVESTIGACIÓN SOBRE EL APARTADO VISUAL DE LAS GRÁFICAS Y TABLAS

31

En todas las gráficas se ha incluido una opción de filtrado, la cual permite realizar una selección de los paciente por dos de sus atributos, los biomarcadores y los tratamientos ²⁸, las gráficas que aparecen después de realizar este filtro serán exclusivamente con los datos de estos pacientes que cumplan con los atributos seleccionados.



The image shows a web interface titled "Filtrador de pacientes" (Patient Filter). It features two columns of checkboxes for selection. The left column, labeled "Biomarcadores" (Biomarkers), lists: NGS, PDL1, EGFR, ALK, ROS1, KRAS, BRAF, HER2, NTRK, FGFR1, RET, MET, PI3K, and TMB. The right column, labeled "Tratamientos" (Treatments), lists: Radioterapia (Radiotherapy), Quimioterapia (Chemotherapy), and Cirugía (Surgery). At the bottom center, there is a green button labeled "Realizar filtrador" (Apply filter).

Figura 5.14: Opciones de filtrado de pacientes.

²⁸Requisito de los oncólogos

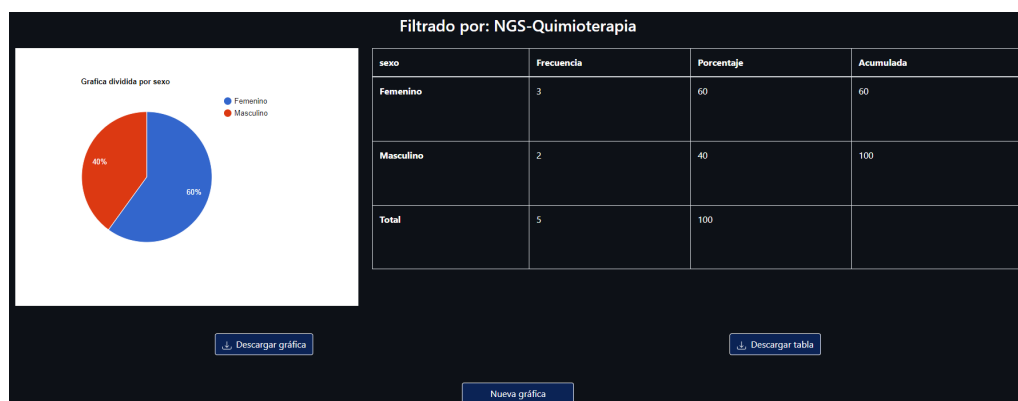


Figura 5.15: Gráfica y tabla de frecuencias filtradas por biomarcador NGS y tratamiento quimioterapia.

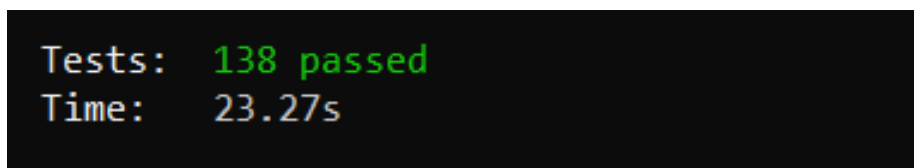
5.6. Pruebas

La realización de pruebas es una parte fundamental en el desarrollo de software para conseguir un producto de calidad y sin errores. Debido a que el objetivo de este proyecto era desarrollar una aplicación web que iba a ser usada por los médicos para la introducción de datos se vió como algo necesario la realización de pruebas que nos aseguren que la web funciona al 100 %, que los datos se almacenan correctamente en la base de datos y que no existe ningún error.

Para esto se llevaron a cabo tres tipos de pruebas: pruebas de integración, pruebas de validación y pruebas de seguridad.

Pruebas de integración

Las pruebas de integración son aquellas que se encargan de probar que los elementos unitarios de código funcionan correctamente en conjunto [25]. En nuestro proyecto se han realizado este tipo de pruebas para comprobar la introducción, modificación y eliminación tanto de los datos de los pacientes como de los datos de los usuarios. Para hacer estas comprobaciones se han realizado 138 test los cuales se pueden ejecutar accediendo a la carpeta del proyecto y ejecutando el comando *php artisan test*, en la imagen 5.16 podemos ver el resultado de los test.



```
Tests: 138 passed
Time: 23.27s
```

Figura 5.16: Resultado de la ejecución de los test de integración

Pruebas de validación

Las pruebas de validación son aquellas que se encargan de comprobar que los requisitos y las expectativas del cliente se han cumplido. Este proceso suele realizarse en presencia del cliente o de los usuarios finales [13]. Para la validación de este proyecto se han llevado a cabo dos tipos de pruebas: alfa y beta.

Pruebas alfa

Las pruebas alfa son aquellas que se realizan en presencia del cliente. En mi caso este tipo de pruebas se realizaron desde mi portátil ejecutando la aplicación en local y en presencia de los oncólogos, a los cuales se les permitió la utilización de la aplicación web con todas sus funcionalidades para comprobar que era exactamente lo que querían. Esta prueba fue mucho mejor de lo esperado ya que los únicos errores que se encontraron fueron problemas de nomenclatura en los datos y algún elemento fuera de lugar, es decir problemas que no me iban a llevar mucho tiempo solucionar.

Pruebas beta

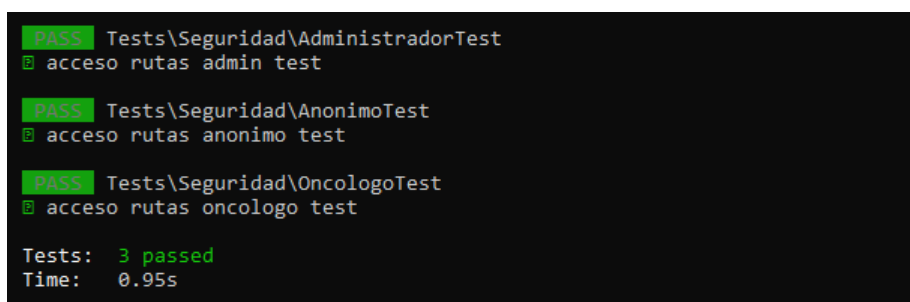
Las pruebas beta son aquellas realizadas por los usuarios finales y en los equipos de los clientes sin la presencia del desarrollador. En mi caso este tipo de pruebas se realizaron cuando la aplicación web se subió al host online Heroku 4.2 y los oncólogos, que en el caso de este proyecto eran tanto los clientes como los usuarios finales, pudieron probar la aplicación web al completo en un entorno real.

Pruebas de seguridad

Las pruebas de seguridad son aquellas que tratan de encontrar vulnerabilidades dentro del software. Como ya se comentó en el punto 5.4, se había realizado todo lo posible para conseguir una buena seguridad dentro de la aplicación, pero luego me di cuenta que podía existir otro problema y esté

era que los middleware no estuvieran funcionando correctamente en alguna de las rutas y cualquier usuario pudiera acceder a ellas sin estar autenticado o sin tener el rol necesario y por lo tanto tener acceso a datos sensibles. Para esto se redactaron pruebas de seguridad en las cuales se realizaron diversas comprobaciones:

- Que un usuario sin autenticarse no pueda acceder a ninguna de las rutas de la web.
- Que un usuario con rol oncólogo pueda acceder a todas las rutas excepto a las reservadas para administradores.
- Que un usuario con el rol administrador tenga acceso a todas las rutas de la web.



```
PASS Tests\Seguridad\AdministradorTest
[+] acceso rutas admin test

PASS Tests\Seguridad\AnonimoTest
[+] acceso rutas anonimo test

PASS Tests\Seguridad\OncologoTest
[+] acceso rutas oncologo test

Tests: 3 passed
Time: 0.95s
```

Figura 5.17: Resultado de la ejecución de los test de seguridad.

5.7. Fase de refactorización del código

Una vez finalizada la parte de programación del proyecto, se llevó a cabo un proceso de refactorización para asegurar un mínimo de calidad dentro de nuestro código. Para realizar esto se uso la herramienta Codacy [4.2](#).

La primer vez que se ejecutó el análisis de código de Codacy se obtuvo una nota de B y 626 *issues* lo cual supone un 14%. Las *issues* se dividen en 6 tipos: *Code Style*, *Security*, *Error Prone*, *Performance*, *Compatibility* y *Unused Code*, y en este proceso de refactorización se basó en la eliminación de todas las *issues* menos las de *Code Style* ya que se consideraron las menos importantes. Se consiguió reducir hasta un 6% como se puede ver en la imagen [5.18](#) donde la mayoría son de *Code Style* y el resto son de ficheros internos Laravel [5.19](#) y se consiguió una calificación de A en la calidad del código [5.20](#).



Figura 5.18: Gráfico de issues Codacy.

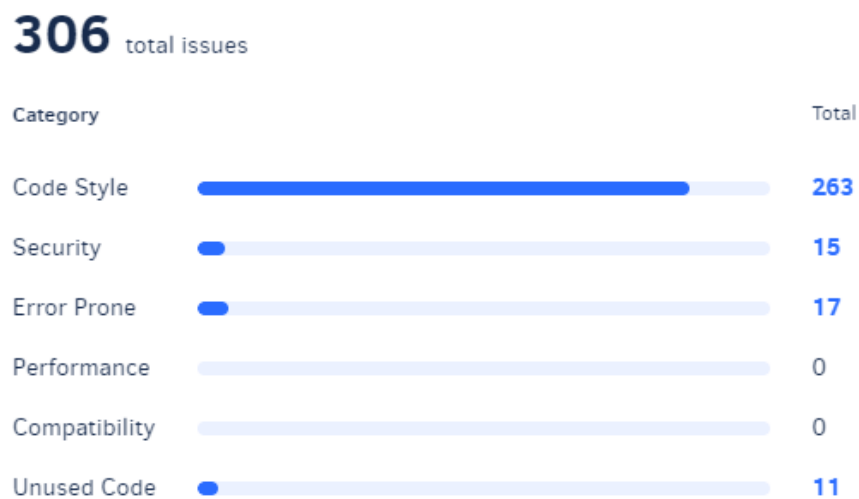


Figura 5.19: Tipos de issues Codacy.

Repository name	Grade	Issues	Complexity	Duplication
TFG_HUBU	A	6%	0%	2%

Figura 5.20: Nota de Codacy.

Se puede acceder al informe de calidad desde https://app.codacy.com/gh/CarlosOrtu/TFG_HUBU/dashboard?utm_source=github.com&utm_medium=referral&utm_content=CarlosOrtu/TFG_HUBU&utm_campaign=Badge_Grade o desde el readme del proyecto en GitHub.

5.8. Opinión de los oncólogos respecto a la aplicación final y al trabajo realizado

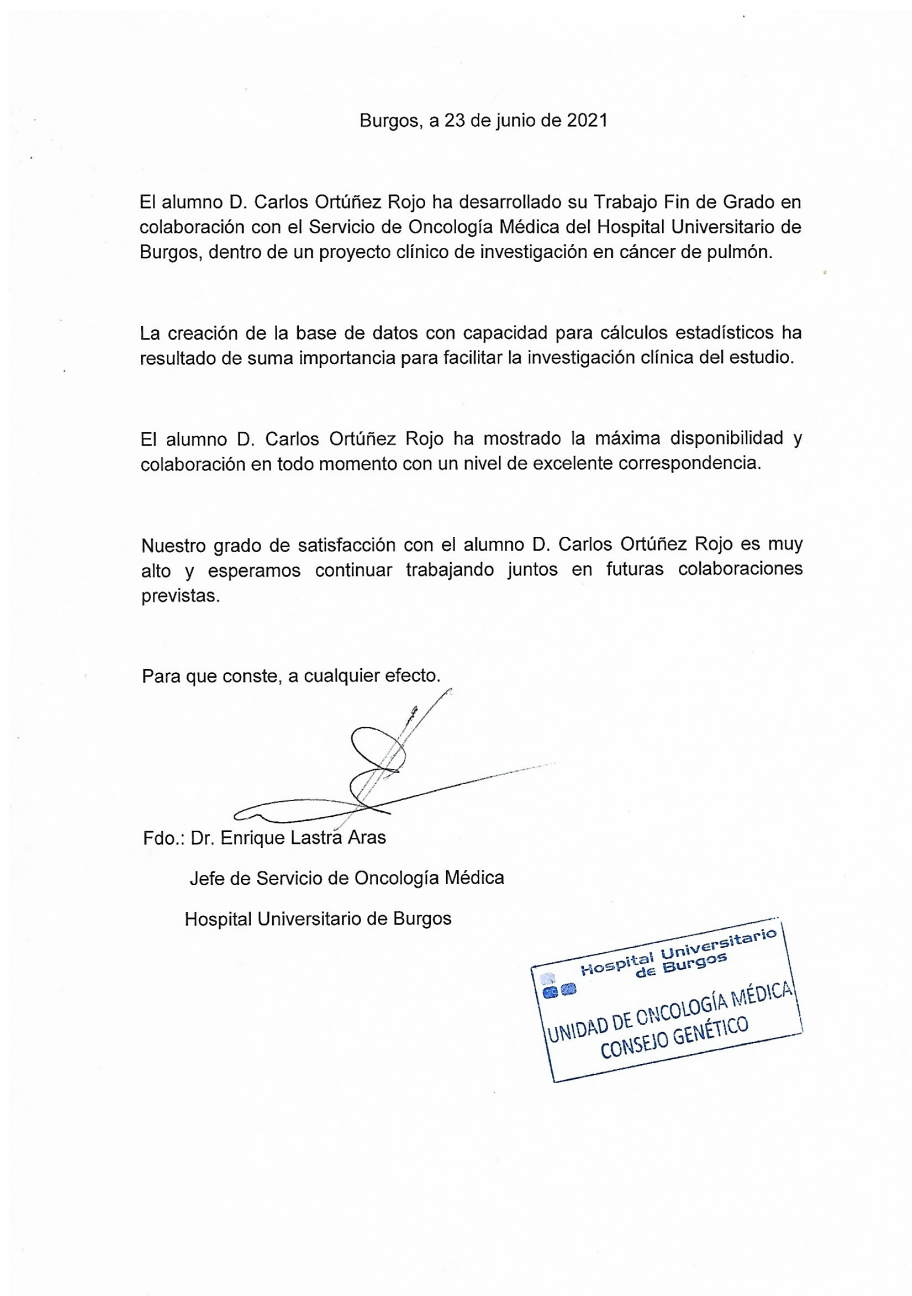


Figura 5.21: Opinión del oncólogo Enrique Lastra del trabajo realizado en este proyecto.

Trabajos relacionados

Este apartado se van exponer diferente herramientas similares a la desarrollada en este trabajo.

6.1. Aplicación web de gestión de datos de pacientes Ecuador

Aplicación web que permite almacenar la información de los paciente y de su correspondiente atención médica y también la asignación de turnos para las citas de los paciente [18].

Funcionalidades	TFG HUBU	Gestión de datos Ecuador
Gestión de pacientes	X	X
Visualización mediante gráficas	X	
Visualización mediante tablas	X	
Creación de base de datos sintética	X	
Exportación de datos	X	
Gestión de citas de pacientes		X

Tabla 6.1: Comparación entre el proyecto actual y otro similar desarrollado en Ecuador.

6.2. Gestión de información sobre personas discapacitadas

Aplicación web que permite la gestión de datos de los pacientes que sufren discapacidad, un sistema de visualización de estadísticas y un calendario para la gestión de pacientes [11].

Funcionalidades	TFG HUBU	Gestión de pacientes con discapacidad
Gestión de pacientes	X	X
Visualización mediante gráficas	X	
Visualización mediante tablas	X	X
Creación de base de datos sintética	X	
Exportación de datos	X	
Calendario para la gestión de pacientes		X

Tabla 6.2: Comparación entre el proyecto actual y otro similar para personas con discapacidad.

6.3. Gestión de datos para el cáncer infantil

Aplicación web que permite la gestión de dato de pacientes que sufren cáncer infantil, así como un sistema de gestión de consultas y un sistema de exportación de datos. [12].

Funcionalidades	TFG HUBU	Gestión cáncer infantil
Gestión de pacientes	X	X
Visualización mediante gráficas	X	
Visualización mediante tablas	X	
Creación de base de datos sintética	X	
Exportación de datos	X	X
Gestión de consultas		X

Tabla 6.3: Comparación entre el proyecto actual y otro similar desarrollado en Castilla La Mancha.

6.4. Gestión y visualización de datos de enfermos de cáncer

Continuación de un trabajo de fin de grado que consiste en la adicción de métodos de visualización a un sistema de gestión de datos de pacientes. Para esto usa diferentes estilos de gráficas y un sistema de visualización de datos individuales de cada paciente [15].

Funcionalidades	TFG HUBU	Gestión y visualización de datos
Gestión de pacientes	X	X
Visualización mediante gráficas	X	X
Visualización mediante tablas	X	
Creación de base de datos sintética	X	
Exportación de datos	X	X
Gráficas tipo <i>Bubble</i> y <i>Tree</i>		X

Tabla 6.4: Comparación entre el proyecto actual y otro similar desarrollado en Barcelona.

Conclusiones y Líneas de trabajo futuras

Conclusiones

Tras haber finalizado el proyecto se puede decir que se han completado todos los objetivos que se habían establecido al principio de este. Como resultado se ha obtenido una aplicación web que permite un eficiente almacenamiento y una sencilla visualización de los datos de pacientes. La idea de este trabajo es ser usado en el HUBU pero con alguna modificación podría implementarse en otros hospitales que tengan el mismo problema.

A nivel personal la realización de este trabajo, sobre todo al principio, ha sido un poco agobiante debido a que se contaba con la presión extra de estar trabajando para unas personas que confiaban en que el proyecto saliera adelante sin ningún problema. Pero a medida que iba avanzando y sobre todo al finalizar el proyecto, el resultado ha sido muy gratificante, ya que no solo he sido capaz de trabajar con esa presión añadida sino que el resultado ha sido bueno y los oncólogos están muy contentos con la labor realizada. También ser consciente que mi aplicación va a ser usada y va ayudar a gente es lo que más positivamente valoro de este proyecto.

A nivel formativo he adquirido conocimientos tanto de diseño como de desarrollo web los cuales son unos campos que desconocía hasta el momento y de los cuales estaba muy interesado en aprender. También al tener pensado seguir llevando el mantenimiento de esta aplicación web se podrá seguir aprendiendo y aumentando los conocimientos sobre estas ramas de la informática.

Como conclusión final puedo decir que estoy bastante contento y orgulloso del trabajo que he llevado a cabo, pero me han quedado algunas cosas

pendientes como la inteligencia artificial, que por falta de datos (ya que los oncólogos están empezando ahora a introducirlos en la base de datos) no se ha podido llevar a cabo y también me hubiera gustado disponer de más tiempo para pulir más el diseño y añadir algún tipo de gráfica más.

Lineas de trabajo futuras

Con los oncólogos del HUBU se acordaron unas líneas de trabajo futuras ya que por contratiempos y por temas burocráticos ha habido tareas que han quedado aún pendientes. A continuación voy a listar todas estas tareas:

- Migrar tanto los datos ya introducidos por los oncólogos en la base de datos como la aplicación web al servidor del HUBU.
- Añadir nuevas gráficas que los oncólogos vean necesarias en el futuro.
- Continuar con el mantenimiento de la web e implementando todas aquellas mejoras o funcionalidades que los oncólogos necesiten.
- Mejorar el diseño de la web haciéndola más intuitiva y adaptándola a las necesidades de los oncólogos.
- Realizar algún modelo de predicción el cual se entrene con los datos introducidos en la aplicación por los oncólogos y permita recomendar los tratamientos o medicamentos a cada paciente según sus características.

Bibliografía

- [1] Cáncer de pulmón: síntomas, diagnóstico y tratamiento. clínica universidad de navarra. <https://www.cun.es/enfermedades-tratamientos/enfermedades/cancer-pulmon>. [Accedido 06/13/2021].
- [2] Definición de percentil - qué es, significado y concepto. <https://definicion.de/percentil/>. (Accessed on 06/15/2021).
- [3] Diferencia entre el cifrado de flujo y el cifrado de bloque / seguridad | la diferencia entre objetos y términos similares. <https://es.sawakinome.com/articles/security/difference-between-stream-cipher-and-block-cipher.html>. (Accessed on 06/14/2021).
- [4] Media, varianza y desviación típica de la distribución binomial. <https://www.superprof.es/apuntes/escolar/matematicas/probabilidades/distribucion-binomial/media-y-varianza-de-la-distribucion-binomial.html>. (Accessed on 06/15/2021).
- [5] Oblicuidad (probabilidad) - wikipedia, la enciclopedia libre. [https://es.wikipedia.org/wiki/Oblicuidad_\(probabilidad\)](https://es.wikipedia.org/wiki/Oblicuidad_(probabilidad)). (Accessed on 06/15/2021).
- [6] Varianza - qué es, definición y significado | 2021 | economipedia. <https://economipedia.com/definiciones/varianza.html>. (Accessed on 06/15/2021).
- [7] ¿qué es el cáncer? - instituto nacional del cáncer. <https://www.cancer.gov/espanol/cancer/naturaleza/que-es>. [Accedido 06/13/2021].

- [8] Ayuda epidat 4 distribuciones de probabilidad octubre2014. 2014. (Accessed on 06/15/2021).
- [9] ¿qué es y en qué consiste la encriptación de datos? | econectia. <https://www.econectia.com/blog/que-es-encriptacion-de-datos>, 2017. (Accessed on 06/14/2021).
- [10] Miguel Angel Alvarez. Qué es mvc. <https://desarrolloweb.com/articulos/que-es-mvc.html>, 2020. (Accessed on 06/15/2021).
- [11] Yudi Castro Blanco. Aplicación web para gestionar información sobre personas discapacitadas. http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1028-99332018000400710. (Accessed on 06/29/2021).
- [12] María López Carrasco. Sistema de gestión de datos de pacientes supervivientes a largo plazo de un cáncer infantil. https://ruidera.uclm.es/xmlui/bitstream/handle/10578/19012/TFG_Mar%c3%ada%20L%c3%b3pez%20Carrasco.pdf?sequence=1&isAllowed=y. (Accessed on 06/29/2021).
- [13] J M Drake. Verificación y validación situación dentro del proceso de desarrollo. 2009. (Accessed on 06/18/2021).
- [14] David Dunning. ¿cuál es la diferencia entre el cifrado de flujo y el cifrado de bloque? https://techlandia.com/diferencia-cifrado-flujo-cifrado-bloque-info_548015/. (Accessed on 06/14/2021).
- [15] Albert Guzmán Espadas. Aplicación para evaluar el grado de caquexia en enfermos de cáncer parte ii: Visualización y gestión de datos. <https://core.ac.uk/download/pdf/148785009.pdf>. (Accessed on 06/29/2021).
- [16] Manuel Trigas Gallego. Metodología scrum.
- [17] Jorman Garcia, Bryan Jorge Plat, and Pablo Salazar. Metodologías Ágiles en el desarrollo de software.
- [18] GUZMÁN ALARCÓN FREDDY GEOVANNY. Aplicación web para mantener el control y almacenamiento de datos de las historias clínicas de los pacientes del hospital del día de la universidad central del ecuator. <http://www.dspace.uce.edu.ec/bitstream/25000/13117/1/T-UCE-0011-328.pdf>. (Accessed on 06/28/2021).

- [19] Heroku. Heroku security | heroku. <https://www.heroku.com/policy/security#:~:text=amazon.com%2Fsecurity-,Privacy,to%20customer%20privacy%20and%20transparency>. (Accessed on 06/18/2021).
- [20] Heroku. Regions | heroku dev center. <https://devcenter.heroku.com/articles/regions>. (Accessed on 06/17/2021).
- [21] Heroku. What is heroku | heroku. <https://www.heroku.com/what>. (Accessed on 06/16/2021).
- [22] Laravel. Database: Migrations - laravel - the php framework for web artisans. <https://laravel.com/docs/8.x/migrations>. (Accessed on 06/17/2021).
- [23] LAravel. Database: Seeding - laravel - the php framework for web artisans. <https://laravel.com/docs/8.x/seeding>. (Accessed on 06/17/2021).
- [24] Laravel. Laravel - the php framework for web artisans. <https://laravel.com/>. (Accessed on 06/16/2021).
- [25] Glenn Lee. Tipos de pruebas de software: diferencias y ejemplos - loadview. <https://www.loadview-testing.com/es/blog/tipos-de-pruebas-de-software-diferencias-y-ejemplos/>, 2020. (Accessed on 06/18/2021).
- [26] Adrián Pousa. Algoritmo de cifrado simétrico aes. aceleración de tiempo de cómputo sobre arquitecturas multicore. 2011. (Accessed on 06/14/2021).
- [27] punchesalar. Ssl certificate generator: Free letsencrypt ssl in minutes - punchesalad. <https://punchsalad.com/ssl-certificate-generator/>. (Accessed on 06/18/2021).
- [28] Julio César Mendoza T. Dialnet-demonstraciondecifradosimetricoyasimetrico-5972695. (Accessed on 06/14/2021).
- [29] Wikipedia. Corrección de bessell - wikipedia, la enciclopedia libre. https://es.wikipedia.org/wiki/Correcci%C3%B3n_de_Bessel. (Accessed on 06/15/2021).
- [30] Wikipedia. Curtosis - wikipedia, la enciclopedia libre. <https://es.wikipedia.org/wiki/Curtosis>. (Accessed on 06/15/2021).

- [31] Wikipedia. Estimador de kaplan-meier - wikipedia, la enciclopedia libre. https://es.wikipedia.org/wiki/Estimador_de_Kaplan-Meier. (Accessed on 06/18/2021).