



**UNIVERSIDAD CÉSAR VALLEJO**

**FACULTAD DE INGENIERÍA Y ARQUITECTURA  
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS**

**Sistema Experto para Diagnóstico de Enfermedades Pulmonares  
en el Policlínico Monserrat**

**TESIS PARA OBTENER EL TÍTULO PROFESIONAL DE:  
Ingeniero de Sistemas**

**AUTOR:**

Nizama Antunez, Gianmarco Roger ([orcid.org/0000-0001-9787-8060](https://orcid.org/0000-0001-9787-8060))

**ASESOR:**

Mgtr. Liendo Arevalo, Milner David, ([orcid.org/0000-0002-7665-361X](https://orcid.org/0000-0002-7665-361X))

**LÍNEA DE INVESTIGACIÓN:**

SISTEMAS DE INFORMACIÓN Y COMUNICACIONES

**LÍNEA DE RESPONSABILIDAD SOCIAL UNIVERSITARIA:**

Promoción de la salud, nutrición y salud alimentaria

**LIMA – PERÚ**

**2022**

### **Dedicatoria**

A mis padres por darme la oportunidad de disfrutar esta hermosa carrera brindándome apoyo en todo momento y a mi amiga que sido mi soporte en los últimos 3 años de conocernos, siendo así, mi complemento en este tramo de mi carrera universitaria.

### **Agradecimiento**

Agradecer en primer lugar a la bendición de Dios, a mi centro de estudios Universidad Cesar Vallejo por forjarme como profesional, a mi asesor que cumplió un rol muy importante inculcándome y motivándome a seguir adelante y a mis padres sien el apoyo de ellos no hubiera llegado tan lejos.

## Índice de Contenidos

<i>Dedicatoria</i> .....	2
<i>Agradecimiento</i> .....	3
<i>Índice de Contenidos</i> .....	4
<i>ÍNDICE DE FIGURAS</i> .....	5
<i>ÍNDICE DE TABLAS</i> .....	6
<i>RESUMEN</i> .....	7
<i>ABSTRACT</i> .....	8
<i>I. INTRODUCCIÓN</i> .....	9
<i>II. MARCO TEÓRICO</i> .....	15
<i>III. METODOLOGÍA</i> .....	29
3.1. Tipo y diseño de Investigación .....	30
3.2. Variables y Operacionalización .....	31
3.3. Población, muestra y muestreo .....	32
3.4. Técnicas e instrumentos de recolección de datos .....	34
3.5. Procedimientos .....	36
3.6. Método de análisis de datos .....	36
3.7. Aspectos éticos .....	38
<i>IV. RESULTADOS</i> .....	40
<i>V. DISCUSIÓN</i> .....	51
<i>VI. CONCLUSIONES</i> .....	55
<i>VII. RECOMENDACIONES</i> .....	57
<i>REFERENCIAS BIBLIOGRÁFICAS</i> .....	59
<i>ANEXOS</i>	

## ÍNDICE DE FIGURAS

<b>Figura 1.</b> Umbrales de decisión .....	25
<b>Figura 2.</b> Fórmula del índice de entrega a tiempo de diagnóstico .....	26
<b>Figura 3.</b> Fórmula del Porcentaje de pacientes satisfechos .....	26
<b>Figura 4.</b> Estructura de metodología de CommonKADS .....	27
<b>Figura 5.</b> Fórmula para el cálculo de la muestra.....	<b>Error! Bookmark not defined.</b>
<b>Figura 6.</b> Cálculo del nivel de signiificancia .....	<b>Error! Bookmark not defined.</b>
<b>Figura 7:</b> Histograma comparativo del índice de entrega a tiempo de diagnóstico .....	42
<b>Figura 8:</b> Histograma comparativo del porcentaje de pacientes satisfechos .....	43
<b>Figura 9:</b> Histograma antes de la implementación del sistema del índice de entrega a tiempo de diagnóstico .....	44
<b>Figura 10:</b> Histograma después de la implementación del sistema del índice de entrega a tiempo de diagnóstico .....	44
<b>Figura 11:</b> Histograma antes de la implementación del sistema del porcentaje de pacientes satisfechos .....	45
<b>Figura 12:</b> Histograma después de la implementación del sistema del porcentaje de pacientes satisfechos .....	46
<b>Figura 13:</b> Región de rechazo del índice de entrega a tiempo de diagnóstico .....	48
<b>Figura 14:</b> Región de rechazo del Procentaje de pacientes satisfechos .....	50

## ÍNDICE DE TABLAS

<b>Tabla 1.</b> Modelo del diseño de investigación de tipo pre-experimental.....	<b>Error!</b>
<b>Bookmark not defined.</b>	
<b>Tabla 2.</b> Identificación de la población .....	<b>Error! Bookmark not defined.</b>
<b>Tabla 3.</b> Técnicas e Instrumentos de recolección de datos ...	<b>Error! Bookmark not defined.</b>
<b>Tabla 4.</b> Nivel de confiabilidad .....	<b>Error! Bookmark not defined.</b>
<b>Tabla 5.</b> Confiabilidad para el índice de entrega a tiempo de diagnóstico .....	<b>Error! Bookmark not defined.</b>
<b>Tabla 6.</b> Porcentaje de pacientes satisfechos.....	<b>Error! Bookmark not defined.</b>
<b>Tabla 7:</b> Estadísticos descriptivos índice de entrega a tiempo de diagnóstico .....	41
<b>Tabla 8:</b> Estadísticos descriptivos Porcentaje de pacientes satisfechos.....	42
<b>Tabla 9:</b> Prueba de normalidad índice de entrega a tiempo de diagnóstico .....	43
<b>Tabla 10:</b> Prueba de normalidad Porcentaje de pacientes satisfechos .....	45
<b>Tabla 11:</b> Prueba t-student índice de entrega a tiempo de diagnóstico .....	47
<b>Tabla 12:</b> Prueba t-student Porcentaje de pacientes satisfechos .....	49

## RESUMEN

Esta investigación lleva por título “sistema experto para diagnóstico de enfermedades pulmonares en el policlínico Monserrat”, en donde el fin de su investigación era analizar si es que el sistema experto podía mejorar el proceso de diagnóstico de enfermedades, por medio de un software que maneja casos y probabilidades respecto a datos ingresados inicialmente en este caso los datos iniciados son los síntomas y el procesamiento de información genera la probabilidad de enfermedad, a más datos mayor será la exactitud del diagnóstico. En este sentido se planteó como problema principal la siguiente pregunta: ¿De qué forma un sistema experto influye en el diagnóstico de enfermedades pulmonares en el policlínico Monserrat? Como objetivo principal se definió lo siguiente: Determinar de qué forma un sistema experto influye en el diagnóstico de enfermedades pulmonares en el policlínico Monserrat, Y como hipótesis principal se definió la siguiente: Un sistema experto mejora el diagnóstico de enfermedades pulmonares en el policlínico Monserrat. Esta investigación fue de tipo aplicada, por lo que se revisó la implementación del software, además el diseño fue pre experimental por lo que se realizó dos evaluaciones la primera antes de implementar el software y la segunda después de su implementación, en donde se evaluaron los siguientes indicadores: el primero fue el índice de entrega a tiempo de diagnóstico, y el segundo fue el porcentaje de pacientes satisfechos. Esta evaluación fue en base a la muestra, la cual fue de 317 atención de una población de 1800 atenciones. Adicional a ello esta investigación fue desarrollada con el lenguaje de programación PHP bajo el Framework Laravel, y la base de datos MySQL. El marco de trabajo para el desarrollo del software fue Scrum. Y se puede concluir que la índice entrega a tiempo diagnóstico tuvo un aumento desde un 64.48% a un 81.26% lo que equivale un 16.78%. Además, el porcentaje de pacientes satisfechos también tuvo un aumento desde un 54% a un 68% lo que equivale a un 14%. De esta manera hipótesis alterna y rechazando la hipótesis nula

Palabras clave: Sistema experto, diagnóstico de enfermedades

## ABSTRACT

This study is entitled "Expert system for the diagnosis of pulmonary diseases in the Monserrat polyclinic", with the aim of analyzing whether the expert system could improve the diagnosis process of diseases, through a software that handles cases and probabilities with respect to initially-entered data. In this case, the initial data are the symptoms and the processing of information generates the probability of disease: the more data, the greater the accuracy of the diagnosis. In this sense, the following question was posed as the main problem: How does an expert system influence the diagnosis of pulmonary diseases in the Monserrat polyclinic? The main objective was defined as follows: To determine in what way an expert system influences the diagnosis of pulmonary diseases in the Monserrat polyclinic; and, as main hypothesis, the following was defined: An expert system improves the diagnosis of pulmonary diseases in the Monserrat polyclinic. This research was of applied type, so the implementation of the software was reviewed. Also, the design was pre-experimental so two evaluations were carried out, the first one before implementing the software and the second one after its implementation, where the following indicators were evaluated: the first one was the rate of on-time delivery of diagnosis, and the second one was the percentage of satisfied patients. This evaluation was based on the sample, which was of 317 attended, from a population of 1800 attended to. Additionally, this study was developed with the PHP programming language under the Laravel Framework, and the MySQL database. The software development framework was Scrum. It can be concluded that the on-time diagnosis delivery rate increased from 64.48 % to 81.26 %, which is equivalent to 16.78 %. In addition, the percentage of satisfied patients also increased from 54 % to 68 %, which is equivalent to 14 %. Thus, the alternative hypothesis and the null hypothesis were rejected.

Key words: Expert system, disease diagnosis



## **I. INTRODUCCIÓN**

El mundo ha tenido un cambio drástico a partir de la pandemia mundial ocasionada por el coronavirus. Esta enfermedad ha generado muchas pérdidas y también alejado a la población bastante vulnerable a contraer enfermedades pulmonares o relacionadas. Es por esta razón la gran importancia de investigar cómo la tecnología puede influenciar en el diagnóstico de estas enfermedades para dar soluciones más rápidas y sobretodo preventivas.

Para la OPS (2021) menciona que las enfermedades respiratorias crónicas se encuentran ubicadas entre las principales causas de muerte y discapacidad sobre todo en América latina, enfermedades como obstrucción crónica pulmonar, asma, hipertensión pulmonar, entre otras. Existe una gran cantidad de factores que afectan a los pulmones en las personas, entre ellos el consumo de tabaco, exposición al polvo, contaminantes ambientales, entre otros con los cuales vivimos el día día. En la región de América en el año 2019 las enfermedades respiratorias causaron 534,242 muertes, encabezando a Honduras como el país más afectado, seguido por Argentina y Haití.

La OMS (2020) indica que una de las causas principales de la muerte Y discapacidad son las enfermedades respiratorias, en el mundo 65 millones de personas aproximadamente parecen de la enfermedad pulmonar obstructiva crónica, esta causa una tasa de muertes de 3 millones de personas anuales. Por lo cual la Organización Mundial de la Salud se encuentra en una constante investigación y trabajo para reducir estos problemas, sobre todo en este tiempo de pandemia que las enfermedades se han vuelto incontrolables.

El Ministerio de Salud (2021) menciona que en el país las enfermedades respiratorias sobre todo en temporada de invierno son bastante comunes, y actualmente las personas están sufriendo no sólo enfermedades sino también con el temor de haber contraído Covid 19, pero menciona que es bueno saber diferenciar para evitar tensiones psicológicas, en el caso del coronavirus el dolor en los músculos y el cuerpo, sobretodo en la espalda son bastante fuertes, asimismo de la mano de fiebres altas e irritación a la faringe. Por lo que recomienda mantener

los cuidados necesarios para evitar el contagio, sobretodo en los tiempos de invierno, ya que las enfermedades respiratorias son muy comunes en el país.

Para la Sociedad Peruana de Neumología (2020) menciona que los coronavirus son una extensa familia de virus las cuales causan enfermedades no sólo personas y también a humanos, por medio de infecciones respiratorias que pueden presentarse desde un simple resfriado hasta un síndrome respiratorio de Oriente medio, por lo cual los protocolos de cuidados deben ser respetados y cumplidos, ya que el índice de muertes ha ido en subida y si no existe el cuidado necesario, se puede descontrolar y generar más muertes de las ya dadas en esta pandemia.

En un enfoque direccionado a la empresa, la organización beneficiaria es el Policlínico Monserrat, Ubicada en Carabayllo, el cual brinda el servicio de medicina general y pediátrica, psicología, odontología, laboratorio y ahora pruebas COVID. Se realizó una entrevista al encargado de área el señor Miguel Pérez (Ver anexo2) el cual menciona que el policlínico inicia su proceso de atención al cliente o al paciente, por medio del registro del mismo y creación de la historia, luego de esto se genera una cita para que el doctor pueda atender y generar el diagnóstico de la enfermedad que tiene el paciente, para luego poder generar un tratamiento y las recetas respectivas. Con la situación actual que se viene dando por el coronavirus el policlínico ha empezado a tener problemas en la atención de los pacientes y en la generación de diagnósticos, ya que esta enfermedad está generando efectos secundarios, secuelas y tensión en los pacientes, lo que genera una gran cantidad de pacientes muchas veces desesperados por ser atendidos o por saber el diagnóstico que se tiene. Esto está generando cuellos de botella, y falta de satisfacción a los pacientes por el tiempo de espera no sólo en la atención sino también en la generación de diagnósticos y resultados de los exámenes médicos, Y ahora con el tema del coronavirus, es un poco complicado mantener el distanciamiento de los pacientes que quedan en la espera de una atención, y a veces todos los problemas se juntan y dar una solución es complicada ya que a veces se debe poner prioridad a las emergencias y los pacientes tienen que esperar mucho más tiempo. Esta situación se viene dando desde inicios de pandemia.

En base a todo lo mencionado anteriormente se define como problema principal: ¿De qué forma un sistema experto influye en el diagnóstico de enfermedades pulmonares en el policlínico Monserrat? Y como problemas específicos ¿De qué forma un sistema experto influye en el índice de entrega a tiempo de diagnóstico en el diagnóstico de enfermedades pulmonares en el policlínico Monserrat? Y ¿De qué forma un sistema experto influye en el porcentaje de pacientes satisfechos en el diagnóstico de enfermedades pulmonares en el policlínico Monserrat?

Esta investigación se justifica, ya que al revisar la implementación de un sistema experto lo que se espera es la mejora del proceso diagnóstico de enfermedades pulmonares y así tener una mejor atención a los pacientes generando ahorro de recursos y mejorando la satisfacción de los mismos, continuación se detalla los niveles de justificación.

Justificación Institucional, según López (2019) Los sistemas expertos cada vez más están siendo utilizado en diversos campos donde se requiera procesamiento de datos y toma de decisiones, es por ello permite adaptarse y auto aprender de muchos procesos.

Esta presente investigación apoyará a los pobladores que acuden a diario al policlínico, debido a que un sistema experto para diagnóstico pulmonares, permitirá diagnosticar a una persona si cuenta con problemas del pulmón, tan solo con ingresar síntomas o molestias con la que se cuenta. Con la implementación de este sistema experto, permitirá crear un hito de tecnología para que otros policlínicos, así como instituciones públicas puedan utilizar a favor de sus ciudadanos.

Justificación económica, según Peña (2018), Un sistema experto, además, mejora la productividad al resolver y decidir los problemas más rápidamente. Esto permite ahorrar tiempo y dinero. A veces sin esa rapidez las soluciones obtenidas serían inútiles. Los valiosos conocimientos de un especialista se guardan y se difunden, de forma que, no se pierden, aunque desaparezca el especialista. En los sistemas expertos se guarda la esencia de los problemas que se intenta resolver y

se programa cómo aplicar los conocimientos para su resolución. Ayudan a entender cómo se aplican los conocimientos para resolver un problema. Esto es útil porque normalmente el especialista da por ciertos sus conocimientos y no analiza cómo los aplica.

El desarrollo de sistema experto para diagnóstico de enfermedades pulmonares en el policlínico Monserrat, facilitará la detección de alguna enfermedad pulmonar en base a síntomas que sean ingresados por los pacientes, es por ello que cualquier persona desde cualquier lugar del mundo con conexión a internet puede lograr acceder al sistema experto y al colocar sus síntomas podrá validar si se encuentra con problemas de salud o se encuentra totalmente sano; el sistema experto estará aprendiendo con los síntomas planteados día tras día pues a lo largo del tiempo podrá diagnosticar con más precisión alguna enfermedad pulmonar.

Justificación Operativa, según López (2019), La utilización de los sistemas expertos permite generar valor a las empresas, genera nuevos niveles empresariales y permite agilizar procesos donde se requiera un humano como experto, es decir el sistema puede sustituir al experto. (p, 19)

Actualmente el policlínico Monserrat, no cuenta con Sistema experto para el diagnóstico de enfermedades pulmonares para sus pacientes, es por ello que los pacientes tienen que separar cita esperar mucho tiempo para ser atendido por un personal médico, es por ello que con el sistema experto el paciente podrá ingresar sus malestares y logrará tener un diagnóstico para determinar problemas del pulmón.

Justificación Tecnológica, según Tundidor (2020), las organizaciones evolucionan a pasos agigantados permitiendo el uso de TIC'S en sus procesos mejorando la productividad de sus áreas, logrando contar con soluciones prácticas y tener ventaja competitiva sobre otras organizaciones.

El policlínico Monserrat, estará innovando en tecnología en tiempos de crisis por la pandemia del COVID 19, ya que permitirá mediante un sistema experto detectar con tan solo digitar algunos síntomas si la persona cuenta con un problema pulmonar, así como también con los síntomas el sistema podrá aprender debido a su sistema de inteligencia artificial, este sistema permitirá en muchos casos reemplazar a un especialista de la salud, es por ello que se espera que con esta investigación otros distritos puedan invertir en tecnología y/o sistemas para el beneficio de su comunidad.

Luego de esto se definen los objetivos, como el principal: Determinar de qué forma un sistema experto influye en el diagnóstico de enfermedades pulmonares en el policlínico Monserrat y como objetivos específicos: Determinar de qué forma un sistema experto influye en el índice de entrega a tiempo de diagnóstico en el diagnóstico de enfermedades pulmonares en el policlínico Monserrat y Determinar de qué forma un sistema experto influye en el porcentaje de pacientes satisfechos en el diagnóstico de enfermedades pulmonares en el policlínico Monserrat.

Como hipótesis principal se define la siguiente: Un sistema experto mejora el diagnóstico de enfermedades pulmonares en el policlínico Monserrat y como hipótesis específicas: Un sistema experto mejora el índice de entrega a tiempo de diagnóstico en el diagnóstico de enfermedades pulmonares en el policlínico Monserrat y Un sistema experto mejora el porcentaje de pacientes satisfechos en el diagnóstico de enfermedades pulmonares en el policlínico Monserrat.

## **II. MARCO TEÓRICO**

En un entorno nacional, en el año 2020, Medrano Hector presentó su tesis de título “Diseño e implementación de un sistema experto para el diagnóstico de desnutrición en niños menores de 2 años en el área de pediatría del centro de salud Tupac Amaru – Chaupimarca” Se presentó dicha investigación en la Universidad Nacional Daniel Alcides Carrión. Se necesita un software en los establecimientos médicos que logre brindar un diagnóstico eficiente que determine los diversos tratamientos para la desnutrición en niños menores de 24 meses. Por dicha causa se estableció como objetivo fundamental el diseño e implementación de un sistema experto que optimice el diagnóstico de dicho problema antes mencionado en el área pediátrica del centro de salud Tupac Amaru – Chaupimarca. Esta presente tesis es de tipo descriptiva – correlacional – aplicada, utiliza el diseño no experimental de corte Transversal, su población la conforman 95 personas y toma como muestra a 48. Finalmente se tuvo como conclusión que se logró una mejora en el diagnóstico del problema antes mencionado, logrando que se incrementara de manera positiva la satisfacción de los usuarios.

En el año 2020, los autores Bernaldo Ronel y Raymundo Lisset presentaron su tesis titulada “Propuesta de implementación de la tecnología emergente basada en inteligencia artificial (AI) en diagnóstico médico en Hospital Cayetano Heredia” dicha tesis se presentó en la Universidad Tecnológica del Perú. Por la existencia de diferentes problemas en los centros de salud de Lima, se tiene como principal objetivo implementar la tecnología chatbot en base a la inteligencia artificial (IA) en diagnóstico de salud. En esta investigación se usó la metodología cualitativa, además, se toma como población y muestra a 53 personas. Y se toma como conclusión que esta investigación ha demostrado potencial y eficiencia, facilitando al establecimiento médico en la atención de pacientes y en optimizar a la organización.

En el 2020 Carlos Garcia desarrolló su investigación en la Universidad Federico Villarreal titulada sistema inteligente móvil basado en redes neuronales como solución para poder desarrollar el diagnóstico previo de enfermedades renales. Donde menciona que esta investigación tuvo como finalidad generar una herramienta la cual facilite el diagnóstico previo del estado como se encuentra el



riñón en pacientes que se encuentran sanos y también en pacientes que tienen enfermedades renales, por medio de un algoritmo de redes neuronales a través de un aplicativo móvil. Se utilizó una población de 268 imágenes de las cuales 154 fueron de pacientes sanos y 154 de pacientes enfermos, para poder generar la base de conocimiento. El tipo de investigación es la aplicada y el diseño experimental logrando un índice de sensibilidad al 100% y de especificidad a un 99%. Concluyendo que el software de reconocimiento fue lo suficientemente eficiente y los resultados fueron aceptables.

En el año 2019, Carita Fiorella realizó su tesis, la cual presenta el título “sistema experto para el diagnóstico de paraparesia espástica tropical asociado al htlv – 1.” Dicha investigación fue presentada en la Universidad Nacional José María Arguedas. Existe un estimado de más de 100 mil infectados de retrovirus los cuales desconocen dicha infección, la cual causa Paraparesia Espástica Tropical (PET). El desconocimiento y falta de información dificulta al diagnóstico y tratamiento, lo que trae severas consecuencias. Por tal motivo, se tiene como principal objetivo el desarrollo de un sistema experto para obtener el más acertado diagnóstico de PET. Y se tiene como conclusión que se obtuvo satisfactoriamente un sistema experto a modo de prototipo que, en base a las reglas determinadas por expertos médicos especialistas, ayuda en el diagnóstico de la enfermedad antes referida.

En el 2018 Villanueva Sergio y Taboada Wilmar desarrollaron su investigación titulada sistema experto de tipo probabilístico para detectar el diagnóstico de infecciones respiratorias agudas en el hospital Provincial de Bolívar, en la Universidad nacional de Trujillo. En la cual se desarrolló un sistema experto de tipo probabilístico como herramienta para el apoyo médico y técnico de enfermería para la toma de decisiones acertadas. La metodología de desarrollo fue basado en tres niveles bajo el estándar de la metodología CommonKADS, ya que se tuvo como objetivo el desarrollar este sistema experto. La población que se utilizó fue de 27 tipos de enfermedades, y la muestra fue conformada por sólo cinco de ellas. El diseño fue no experimental del tipo descriptivo simple y el tipo aplicada. Se concluye que se logró la recolección y análisis de información para generar el proceso de diagnóstico. También se logró elaborar una base de datos información

de más de 500 historias clínicas. Y se logró la implementación de la metodología ComonKADS en sus tres niveles

En el año 2018, Arias Victor presentó su tesis denominada “Sistema experto para el diagnóstico de enfermedades respiratorias crónicas en el distrito la esperanza – provincia de Trujillo” la cual fue presentada en la Universidad Privada del Norte. La tercera mayor causa de muerte en el Perú y el mundo, por eso el objetivo fundamental es el desarrollo de un sistema experto que ofrezca un confiable diagnóstico de enfermedades crónicas respiratorias para el Distrito de la Esperanza. Esta tesis tiene un diseño pre experimental, su población es de 105 pacientes con algún problema respiratorio, de muestra se eligieron a 15 pacientes. Y como conclusión se tiene que, dicho sistema experto tuvo un mayor porcentaje de correctos diagnósticos en comparación al médico experto.

En el 2018 Franz Gupioc y Brayan Peña, desarrollaron su investigación titulada sistema experto médico para lograr la mejora del diagnóstico de pacientes con depresión en el centro médico Santa Lucía de moche. El objetivo principal de esta investigación fue mejorar el diagnóstico de pacientes con depresión en el centro de salud por medio de un sistema experto, además de incrementar la cantidad promedio diagnósticos, reducir el tiempo de diagnósticos y aumentar el nivel de satisfacción del personal. La población y la muestra estuvo definida por cuatro psicólogos y cuatro diagnósticos semanales, la metodología para el desarrollo fue Iconix con un diseño experimental, pre experimental y un tipo de investigación aplicada. Logrando como resultado que el número de diagnósticos entregados a tiempo aumentó un 45.83%, se redujo el tiempo promedio del diagnóstico en un 71.12% y se aumentó el el nivel de satisfacción en un 35.4%.

En el 2018 Padilla Mariluz, Santa Cruz Chamoro y Evans Jeans desarrollo de investigación titulada desarrolló un sistema experto médico para lograr el mejoramiento del diagnóstico de enfermedades gastroenterológicas en el hospital de Pampas de Tayacaja. La cual menciona como objetivo principal es determinar de qué manera un sistema experto puede influenciar en el diagnóstico de pacientes en las enfermedades gastroenterológicas. Esta investigación fue de tipo aplicada con un

nivel explicativo por medio de un método inductivo deductivo, utilizando el diseño experimental, pre experimental además utilizó una muestra de 77 pacientes, y se logró aumentar el nivel de satisfacción en un 29.4%, además el nivel de satisfacción del diagnóstico tuvo un aumento del 29.2%, validando que un sistema experto si mejoró el proceso de diagnóstico de enfermedades gastroenterológicas.

En un entorno internacional, en el año 2020, Sainz Leisi, et, all. realizaron un artículo de investigación de título “Sistemas de expertos desarrollados en el mundo para la detección, el diagnóstico y el tratamiento del cáncer” en el Centro Provincial de Genética Médica. Pinar del río, el cual resume que, los sistemas expertos se encuentran dentro de las ramas de inteligencia artificial. El trabajo presente tuvo como primordial objetivo caracterizar sistemas expertos internacionales para lograr detectar el cáncer, diagnosticarlo y dar tratamiento. Y se toma como conclusión que un sistema experto garantiza en una gran parte de los casos un nivel de acierto el cual oscila entre el 87 y el 100%, lo cual favorece a la detección de un cáncer temprano para que se pueda dar un eficaz tratamiento.

En el año 2020, Serrano Juan presentó su tesis titulada “Prototipo De Aplicación Móvil Para La Identificación De Mazorcas De Cacao Enfermas Haciendo Uso De Visión Por Computadora Y Aprendizaje De Máquina” en la Universidad Autónoma de Bucaramanga. Se presentó una reducción de la producción de cacao, debido a la enfermedad Monilia y las inundaciones, por esto se propuso como principal objetivo el desarrollo de un prototipo de app móvil para detectar mazorcas enfermas. Y se tuvo como conclusión que el uso de estas tecnologías puede utilizarse en diversos lugares de investigación, logrando que un gran porcentaje de procesos puedan impulsar a los países a tener novedosos retos.

En el año 2020, Mar Omar, et all. Presentaron su trabajo de investigación denominado “Sistema de apoyo al diagnóstico médico de COVID-19 mediante mapa cognitivo difuso” en la Universidad de las Ciencias Informáticas. El cual tuvo como fundamental objetivo el desarrollo de un sistema de apoyo al diagnóstico de COVID-19. Y se obtuvo como conclusión que dicho sistema logra ser un sistema confiable al momento de tomar decisiones en el diagnóstico de la Covid-19, puesto

que obtiene criterios partiendo del modelo de las relaciones causales, logrando hacerlo extensibles a diversos tipos de emergencias.

En el 2020 Marciela Escobar desarrolló su tesis titulada desarrollo de software médico para el apoyo del diagnóstico de enfermedades respiratorias en el consultorio médico ConsulMed. La cual se ejecuta con el fin de reducir las frecuentes enfermedades respiratorias en los niños ecuatorianos. El objetivo de esta investigación fue el desarrollo del software médico para realizar el apoyo diagnóstico de enfermedades respiratorias en el consultorio médico. El diseño de esta investigación fue pre experimental y el tipo aplicada. La población fue definida por 24 pacientes que sufren enfermedades respiratorias. Y se concluye que al revisar la implementación del proyecto se pudo generar la satisfacción de las necesidades del centro médico por medio de una aplicación que permitió el apoyo en el diagnóstico de enfermedades respiratorias, gracias al uso de una base de conocimientos la cual permitió la generación de información.

En el año 2019 Elaine Valeria desarrollo de su tesis para Maestria titulada reconocimiento de patrones de imágenes médicas para establecer diagnósticos previos en trastornos pulmonares. En la Universidad técnica de Ambato. La cual menciona que como objetivo principal fue el desarrollar un prototipo para un sistema de reconocimiento de patrones imágenes el cual permite establecer diagnósticos previos sobre trastornos pulmonares, esta investigación fue de tipo descriptiva y el nivel fue causal experimental. Respecto a la población se utilizaron 100 casos de personas entre hombres y mujeres. Y se pudo concluir qué al desarrollar un sistema de diagnóstico si permitió realizar el mismo, pero sin embargo se requirió la intervención de un experto para poder validar el diagnóstico final y darle la veracidad respectiva.

En el 2019 Dimas Aditya desarrollo su tesis titulada “Systematic Literature Review Of Expert System, Fuzzy Logic And Artificial Neural Network Applications” en donde menciona que El desarrollo de la inteligencia artificial tiene un gran impacto en nuestra vida diaria, uno de los impactos está en la aplicación en el campo industrial. La industria de servicios y manufactura es uno de los campos

industriales más grandes que afectan el crecimiento económico. Esta investigación intenta analizar la influencia del desarrollo de la inteligencia artificial en la industria de servicios y manufactura como grandes sectores en el crecimiento económico mediante la revisión sistemática de 55 artículos de investigación de los últimos 5 años (de 2015 a 2019) y recopilados de fuentes relacionadas, incluidas actas de conferencias y académicos. diarios Las técnicas de inteligencia artificial revisadas en esta investigación son sistema experto, lógica difusa y red neuronal artificial. Esta revisión presenta las aplicaciones del sistema experto, la lógica difusa y las aplicaciones de redes neuronales artificiales en la industria de servicios y fabricación y sugiere una aplicación adicional para una mayor exploración de la aplicación.

En el año 2019 Mirón realizó su tesis para Maestría titulada sistema de ayuda al diagnóstico médico basado en Deep Learning en la Universidad autónoma de Madrid. En donde menciona que se realizó un estudio a las diferentes librerías Ing Frameworks enfocados a la deep Learning en phyton. Tiene como objetivo principal el desarrollar una aplicación para el apoyo diagnóstico médico para las enfermedades relacionadas al cáncer con pulmón. Para esto se realizó un análisis de los distintos Frameworks para poder entrenar al modelo utilizado en Python, se generó una base de conocimientos para a partir de ellas mejorar el proceso de detección de patología del pulmón. Generando de esta forma una mejora significativa en el proceso de diagnóstico, la cual contando con la base de conocimientos permitió generar mejores resultados y aproximaciones más exactas de los diagnósticos.

En el año 2018, Cabello María realizó su tesis de título “Arquitectura de Pizarrón de un sistema experto” presentada en la Universidad de Colima, el cual menciona en su epílogo: Puesto que, más que el desarrollo de un sistema experto se presentó una arquitectura multirrazonamiento, se enfocó a la discusión y descripción de las características funcionales y estructurales de dicha arquitectura. La importancia de lo expuesto en este libro es que se abarcan importantes problemas para la inteligencia artificial, no obstante, es posible optimizar la arquitectura de un sistema experto para lograr igualar o superar a un experto

humano según lo expuesto, en el sentido de su estandarización, confiabilidad, disponibilidad y utilidad de conocimientos y optimización de recursos.

En el 2018 Gonzales Neilys, Estrada Vivian y Febles Aylin, desarrollaron su investigación titulada Estudio y selección de las técnicas de Inteligencia Artificial para el diagnóstico de enfermedades, la cual tiene como objetivo realizar la comparación de técnicas de inteligencia artificial que se utilizan con frecuencia para el diagnóstico de enfermedades, cuando se tienen datos registrados respecto al comportamiento de las enfermedades y cómo es que afectan a la población. Esta investigación se apoyó en la teoría de la decisión multicriterio discreta. Y se obtuvo como resultados que se logró identificar la técnica de inteligencia artificial más eficiente para el diagnóstico de enfermedades la cual generó mayor certeza por medio del uso de la teoría de decisión multicriterio discreta, la cual permitió evaluar signos síntomas y distintos factores de riesgos que los enfermos presentan día a día.

Luego se empieza con la definición de la variable independiente, la cual es sistema experto: según López (2019), Un sistema experto es un programa desarrollado para computadoras que permite solucionar problemas generales tratando de emular el comportamiento humano por medio del conocimiento computacional con inteligencia artificial y un dominio.

También para Ortegón (2020), Los sistemas expertos, son aplicaciones que utilizan el conocimiento humano capturados en una computadora para resolver diversos problemas que normalmente son resueltos por humanos expertos, estos sistemas pueden procesar razonamiento que utilizan los expertos para tomar muchas decisiones de dominio, y son utilizados para diversos ámbitos profesionales.

Además, para De Pablos (2019) “Un sistema experto es una aplicación que, con ayuda de una computadora, simula e imita el comportamiento de un humano, este aplicativo tiene aprendizaje debido a la inteligencia artificial, el objetivo principal no es reemplazar el comportamiento de un humano, si no más bien el apoyar con conocimiento al humano.

A continuación se define que es la arquitectura de Software: La arquitectura es un anteproyecto para un sistema. Proporciona una abstracción asociada para administrar el sistema complejidad y establecer una comunicación y mecanismo de coordinación entre elementos. Se define una resolución estructurada para cumplir con todos los requisitos técnicos y necesidades operativas, mientras que optimizar los atributos de calidad comunes como el rendimiento y seguridad. Las características describen las expectativas del código en niveles operativos y técnicos. El desempeño y baja tolerancia a fallas, cuantificabilidad y responsabilidad de la unidad de área son la clave de las características (Jaiswal. 2019. p.1).

Para esta investigación científica se plasmó como solución a los problemas un software experto para el diagnóstico de enfermedades, este sistema tuvo que desarrollarse bajo la arquitectura del tipo cliente servidor y fue desarrollada con el Framework php y la base de datos mysql por medio de un algoritmo de probabilidades y situaciones en base a la información ingresada. (Ver anexo 10)

Y como la variable dependiente: Diagnóstico de enfermedades pulmonares

Para Schanen-Bergot (2019) el diagnóstico es un proceso por el cual a través de una consulta especializada se obtiene la confirmación en la enfermedad o discapacidad del paciente, dependiendo del tipo de paciente los diagnósticos pueden darse en un tiempo más prolongado.

Para Blanco, Bosch y Hernández (2019) menciona que el diagnóstico médico es un proceso que se centra en la entidad nosológica la cual representa alteraciones estructurales y también funcionales, para la determinación de enfermedades.

Para la UTPL (2018) menciona que el proceso de diagnóstico clínico es un procedimiento por el cual el profesional de la salud identifica una enfermedad o el estado que se encuentra el paciente por medio de herramientas que permiten definir su cuadro clínico actual.

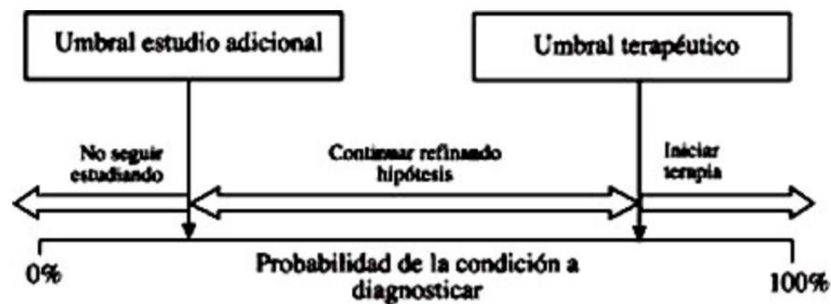
Según lo que menciona Hoon (2020) menciona que el proceso de diagnóstico en su mayoría de veces es un proceso intuitivo del cual sus componentes no son explícitos. En base a sus componentes se puede desglosar las siguientes etapas:

La primera fase es la generación de hipótesis diagnósticas: Durante la cita que se realiza con el paciente, se generan diagnósticos hipotéticos, los cuales permitirán que se realice de manera correcta el diagnóstico definitivo. Por ejemplo, un paciente con fiebre las hipótesis pueden ser tan simples como un cuadro infeccioso o más específicas como una neumonía. Existen muchos métodos por el cual los médicos generan las hipótesis, uno de estos es la aproximación gestáltica al diagnóstico, el cual se trata de que el médico al observar los elementos de los síntomas o signos, se forma de manera intuitiva una imagen mental de lo que puede estar ocurriendo al paciente. Para la asignación de una probabilidad, se tienen en cuenta tres principios heurísticos los cuales son: representatividad, disponibilidad y anclaje y ajuste.

Hoon (2020) La segunda fase es el Refinamiento de las hipótesis diagnósticas: esta es la segunda etapa del proceso de diagnóstico, luego de haber planteado en las primeras hipótesis. En ésta se recolectan nuevas piezas de información tales como examen físico o exámenes adicionales los cuales van a permitir discriminar las hipótesis antes generadas. Es importante aclarar que este proceso no es secuencial ni unidireccional, sino que constantemente se encuentra evocado a distintas y nuevas hipótesis y se van eliminando otras a medida que se va teniendo más información. Durante el proceso de diagnóstico, también se realiza la secuencia de probabilidad de pre Test, Test y probabilidad post Test, Y se va repitiendo cada vez que se tiene mayor información sobre la posible enfermedad. Este proceso es intuitivo y puede realizarse cuantitativamente, para eso se debe comenzar con la asignación de probabilidad concreta a los diagnósticos que se quieren evaluar, la probabilidad pretexto. Normalmente se considera un número limitado de alternativas diagnósticas, las cuales intentan confirmar o descartar.



Hoon (2020) La tercera fase es la verificación del diagnóstico: en este paso se miden los niveles de certeza para confirmar o descartar el diagnóstico, en base a umbrales de decisión. Existe un punto en el cual ya no es necesario realizar nuevos Test, Sino que se debe iniciar con el diagnóstico concordante, a este se le llama umbral terapéutico. Y aquel al cual se le deja de requerir nuevos Test porque se estima que la probabilidad es demasiada baja y no se justifica, se le denomina umbral de estudio o umbral diagnóstico. Los umbrales se visualizan en la siguiente figura:



**Figura 1.** *Umbrales de decisión*

Hoon (2020) Esos umbrales de decisión no son fijos para todas las enfermedades ni para todos los pacientes, sino que van variando dependiendo de ellas. Lo que determina si un hombre se ubica más arriba o más abajo será el costo. En este caso el costo se refiere a algo mucho más que dinero, incluye el costo de dejar pasar un diagnóstico importante, también de tratar a un paciente sano o con una enfermedad distinta, con una terapia potencial tóxica, riesgos de los tests, valores y preferencias del paciente etc. El objetivo de este nivel es poder detectar qué tipo de tratamiento es el que se debe seguir para el paciente, de todas formas, para todos los pacientes siempre existirá una probabilidad distinta y un diagnóstico diferente.

La dimensión seleccionada para esta investigación es: Verificación del diagnóstico, el cual tiene como indicador el Índice de entrega a tiempo de Diagnóstico (IETD), que para Taype et al. (2019) menciona que la acción de cumplir los plazos de entrega de los resultados responde a la calidad que el paciente o médico perciba, ya que al brindar los resultados en los plazos adecuados puede tener gran implicancia en el cuidado del paciente. Este indicador se calcula con la siguiente fórmula:

$$IETD = \frac{NDET}{NDE} \times 100$$

**Figura 2.** *Fórmula del Índice de entrega a tiempo de diagnóstico*

Dónde IETD representa al Índice de entrega a tiempo de Diagnóstico; NDET representa al: Número de diagnósticos Entregados a Tiempo y el NDE es el: Número de Diagnósticos Entregados

El segundo indicador a medir es el porcentaje de pacientes satisfechos: Para Arenal (2020) este indicador permite la medición de la satisfacción de los pacientes atendidos, respecto al servicio que se le ha brindado por parte de la institución médica, permite medir la calidad de la atención en general que se les da a los pacientes. Este indicador se mide con la siguiente fórmula:

$$PPS = \frac{NPS}{TPA} \times 100$$

**Figura 3.** *Fórmula del Porcentaje de pacientes satisfechos*

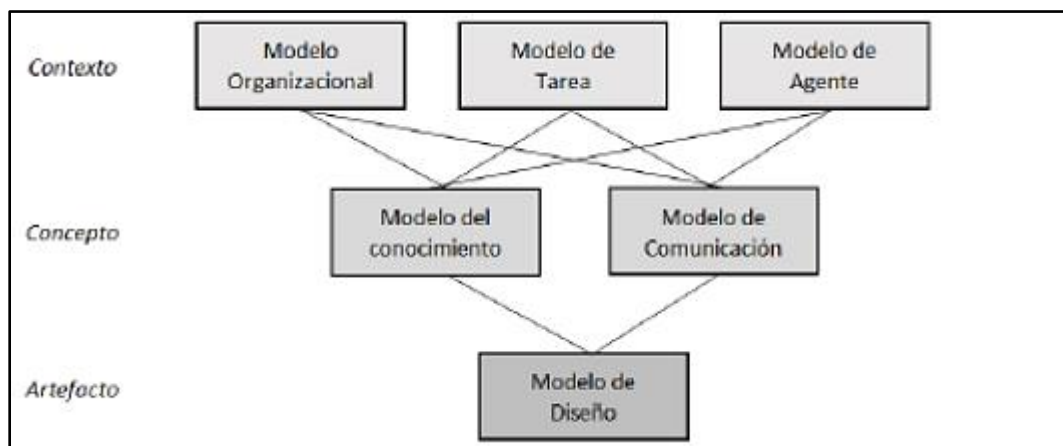
Dónde PPS representa al Porcentaje de pacientes satisfechos; NPS representa al: Número de pacientes satisfechos y TPA representa al: Total de pacientes atendidos

Respecto a las metodologías para el desarrollo del sistema experto se utilizará la metodología CommonKADS, para Ole y Bjorn (2018) CommonKADS es la metodología líder para respaldar la ingeniería del conocimiento estructurado. Se ha desarrollado gradualmente y ha sido validado por muchas empresas y universidades en el contexto del programa europeo de TI ESPRIT. Ahora es el estándar europeo de facto para el análisis del conocimiento y el desarrollo de sistemas intensivos en conocimiento, y ha sido adoptado en su totalidad o parcialmente incorporado en los métodos existentes por muchas empresas importantes en Europa, así como en los EE. UU. Y Japón.

Ole y Bjorn (2018) menciona que CommonKADS permite detectar las oportunidades y los cuellos de botella en la forma en que las organizaciones

desarrollan, distribuyen y aplican sus recursos de conocimiento y, por lo tanto, brinda herramientas para la gestión del conocimiento corporativo. CommonKADS también proporciona los métodos para realizar un análisis detallado de tareas y procesos intensivos en conocimiento. Finalmente, CommonKADS apoya el desarrollo de sistemas de conocimiento que respaldan partes seleccionadas del proceso empresarial. ¿Quién puede utilizar CommonKADS? Esta metodología está orientado para los ingenieros de software que están en el negocio de construir sistemas de TI intensivos en conocimiento que necesitan satisfacer las necesidades de organización del cliente, CommonKADS ofrece un estándar de facto para el desarrollo de sistemas que asegura una solución de alta calidad basada en componentes reutilizables y respaldada a través de prácticas directrices y herramientas. Para administradores de conocimiento CommonKADS ofrece métodos para crear descripciones generales de tareas intensivas en conocimiento dentro del proceso empresarial general, así como técnicas para el análisis detallado del conocimiento, el desarrollo del conocimiento y el almacenamiento del conocimiento. ¿Qué tiene de diferente CommonKADS? A diferencia de otros enfoques de ingeniería del conocimiento, CommonKADS proporciona un vínculo claro con el desarrollo moderno orientado a objetos y utiliza notaciones compatibles con UML. El producto CommonKADS se ha desarrollado durante unos 15 años y ahora se utiliza en una amplia variedad de proyectos de aplicación y es el estándar interno de un número creciente de empresas.

En la siguiente figura se puede visualizar la estructura de la metodología



**Figura 4.** Estructura de metodología de CommonKADS

El marco de trabajo para el desarrollo del proyecto fue SCRUM, Scrum es un marco de gestión de proyectos, utilizado con frecuencia en los negocios y la industria para gestionar y monitorear proyectos complejos. Eso consiste en ceremonias, roles y artefactos que brindan una visión general, inducen la retroalimentación y el diálogo de manera sistemática y mantienen a los empleados encaminados para lograr los objetivos finales del proyecto. Según los desarrolladores, hay tres principios fundamentales que subrayan transparencia, inspección y adaptación (Schwaber & Sutherland, 2020). El marco de Scrum tiene como objetivo contribuir a un entorno social en el que todos los participantes asuman su responsabilidad y se sientan responsables de lograr el objetivo formulado. Además, participan activamente en todas las fases de todo el proyecto (transparencia). Además, se alienta a los empleados participantes a proporcionar comentarios sobre la calidad de productos intermedios (inspección). Además, se les invita a proponer mejoras para mejorar la calidad tanto del producto como del flujo de trabajo durante el proyecto.

Los actores de Scrum son: Scrum master; eso que el que garantiza que este Marco de trabajo se esté realizando de manera ordenada y que la comunicación entre cada uno de sus componentes se realiza de manera fluida, para poder lograr los objetivos del proyecto. Product owner, es aquel que conoce todo el proceso completo y es el que plasman los requerimientos funcionales para que el proyecto pueda ser desarrollado y team member: es el equipo de personas que desarrolla propiamente el proyecto, Tales como programadores, diseñadores, arquitectos, etc. (Vogelzang, Admiraal y Van Driel. 2021. p.721 y 722).

### **III. METODOLOGÍA**

### 3.1. Tipo y diseño de Investigación

#### Tipo de investigación

Para Ramos (2018) menciona que la investigación aplicada es un trabajo original que se utiliza para adquirir nuevos conocimientos, pero que se conduce hacia una finalidad práctica determinada. Este tipo de investigación es aplicada ya que se realizará la implementación de una solución sistemática para el diagnóstico de enfermedades pulmonares con esta solución es el sistema experto.

#### Diseño del estudio

Según Correa (2019), en la investigación experimental, se logra conocer las causas y efectos que se logró dentro de un grupo de población, se permite la comparación de grupos específicos, logrando así una información clasificada y confiable.

Según Casas (2018), En una investigación pre- experimental, no se permite realizar comparaciones con los distintos grupos estadísticos, no se logra contar jun con nivel de control, los investigadores no pueden contar con una información confiable debido a que no se logra conocer los efectos que se generaron en la investigación

Esta investigación es de tipo experimental, pre experimental, ya que al ser una investigación aplicada se realizará la evaluación de los indicadores en los tiempos, un primer tiempo antes del desarrollo del sistema experto y un segundo tiempo después del desarrollo del sistema experto, el diseño por experimental se detalla en la siguiente figura:

**Tabla 1.** Modelo del diseño de investigación de tipo pre-experimental

<b>Grupos</b>	<b>Pre-test</b>	<b>Programa</b>	<b>Post-test</b>
G Experimental (NA)	O1	X	O2

Fuente: Casas (2018)

En donde G, es el grupo experimental que se evaluará, O1 es el primer tiempo antes de la implementación del sistema, X es el sistema experto y O2 son los resultados de la segunda evaluación.

## 3.2. Variables y Operacionalización

### Definición Conceptual

#### **Variable Independiente: Sistema experto**

López (2019), Un sistema experto es un programa desarrollado para computadoras que permite solucionar problemas generales tratando de emular el comportamiento humano por medio del conocimiento computacional con inteligencia artificial y un dominio.

#### **Variable Dependiente: Diagnóstico de enfermedades pulmonares**

Para Schanen-Bergot (2019) el diagnóstico es un proceso por el cual a través de una consulta especializada se obtiene la confirmación en la enfermedad o discapacidad del paciente, dependiendo del tipo de paciente los diagnósticos pueden darse en un tiempo más prolongado.

### Definición Operacional

#### **Variable Independiente: Sistema experto**

López (2019), Un sistema experto, es aquella aplicación tecnológica que permitirá mejorar el proceso de diagnóstico de enfermedades, por medio de una base de conocimiento, el cual permitirá brindar las hipótesis de las posibles enfermedades que pueda tener el paciente, en base a los síntomas detectados.

#### **Variable Dependiente: Diagnóstico de enfermedades pulmonares**

Para Schanen-Bergot (2019), El proceso de diagnóstico de enfermedades se realizará por medio del sistema experto, el cual servirá de apoyo ya que este posee una serie de datos los cuales serán ingresados por los expertos, para poder automatizar y agilizar el proceso.

### 3.3. Población, muestra y muestreo

#### Población

Según Salazar (2020), La población es un grupo de elementos que estamos investigando para poder llegar a diversas conclusiones, una población es un grupo de personas u objetos que cuentan con características comunes que se desean investigar, de igual manera se clasifican en dos tipos de población: finita e infinita

Para la definición de la población, primero se identificó el objeto de estudio, para ambos indicadores, se tuvo como objeto de estudio las atenciones médicas, las cuales se menciona en la entrevista que al mes son 1800 atenciones, por lo cual se toma como población 1800 atenciones.

**Tabla 2.** Identificación de la población

INDICADOR	CANTIDAD DE POBLACIÓN	TIPO DE POBLACIÓN
índice de entrega a tiempo de diagnóstico	1800	Atenciones
Porcentaje de pacientes satisfechos		

#### Muestra

Según Desiderio (2019) la muestra estadística, brinda características similares que se investiga de una población con la finalidad de determinar y identificar patrones que ayuden a la investigación.

Para el cálculo de la muestra, se implementa una fórmula la cual permitirá obtener la cantidad representativa para realizar la evaluación de ambos indicadores, la fórmula es la siguiente:

$$n = \frac{Z^2 \cdot N \cdot p \cdot q}{(N - 1) \cdot E^2 + Z^2 \cdot p \cdot q}$$

**Figura 5.** Fórmula para el cálculo de la muestra



Dónde:

**n:** Tamaño de la muestra

**Z:** Intervalo de confianza

**N:** Tamaño de la población

**p:** Porción o frecuencia esperada

**q:** Complemento de "p" ( $q = 1 - p$ )

**E:** Error de Muestreo

Los datos para la elaboración del cálculo son:

n = Muestra a realizar

N = Población

Z = Nivel de confianza 95%

p = Proporción de éxito 50%

q = Proporción de fracaso 50%

E = Error de estimación (5%=0.05)

A realizar la implementación de la fórmula, el resultado obtenido fue el siguiente:

$$n = \frac{0.95^2 * 0.5 * 0.5 * 1800}{(0.05^2 * (1800 - 1)) + (0.95^2 * 0.5 * 0.5)}$$

$$n=316.59 \Rightarrow 317 \text{ atenciones}$$

## **Muestreo**

Según Gaviria (2019), el muestreo es un proceso o diversos métodos con la finalidad de conseguir una muestra de una o más poblaciones finita o infinita para validar las hipótesis. Existen 02 muestreos más comunes: Muestreo probabilístico - El muestreo probabilístico está destinado para un proceso aleatorio, de igual manera ocurre con las unidades que componen la muestra. Muestreo no probabilístico- El investigador es la persona responsable de seleccionar la muestra que requiera de los distintos grupos de población.

Según Gaviria (2019) menciona el tipo de muestreo no probabilístico estratificado, ya que se va evaluar solo atenciones relacionadas a patologías pulmonares. Y se realizará la estratificación en grupos denominados estratos.

Parece investigación se empleará el muestreo no probabilístico estratificado, agrupando las 317 atenciones en 30 días, ya que la evaluación se empleará en un tiempo de un mes.

### 3.4. Técnicas e instrumentos de recolección de datos

**Fichaje:** Para Vilches (2020) el fichaje es una técnica de recolección de datos, la cual se basa en la ficha de registro o ficha de información. La cual tiene el fin de recopilar datos de manera ordenada y organizada.

**Ficha:** Para Vilcehs (2020) la ficha de registro puede ser tanto Virtual como en una hoja de papel, permiten la recopilación de información para luego realizar el procesamiento de datos.

**Tabla 3.** Técnicas e Instrumentos de recolección de datos

Indicador	Técnica	Instrumento	Informante
Índice de entrega a tiempo de diagnóstico	Fichaje	Ficha	Cliente
Porcentaje de pacientes satisfechos	Fichaje	Ficha	Cliente

### Confiabilidad y Validez

Galeano (2020) Menciona que la técnica de confiabilidad es aquella que permite al investigador validar el nivel de confianza de algún instrumento como en este caso por medio de la prueba de Pre test y re test. Gracias al coeficiente de Pearson, se podrá validar la confiabilidad de los instrumentos de recolección de datos, y se debe seguir los niveles de la siguiente tabla:

**Tabla 4.** Nivel de confiabilidad

Escala	Nivel
$0.00 < \text{sig.} < 0.20$	Muy bajo
$0.20 \leq \text{sig.} < 0.40$	Bajo
$0.40 \leq \text{sig.} < 0.60$	Regular
$0.60 \leq \text{sig.} < 0.80$	Aceptable
$0.80 \leq \text{sig.} < 1.00$	Elevado

Fuente: Galeano (2020)

Para calcular la confiabilidad de los indicadores, se realizó la prueba de pre y re test, los resultados se visualizan a continuación:

Para el indicador índice de entrega tiempo de diagnóstico, el resultado fue de 0.644, que según la tabla se puede concluir que el nivel de confiabilidad de la ficha de registro para este indicador es aceptable.

**Tabla 5.** Confiabilidad para el índice de entrega a tiempo de diagnóstico

<b>Correlaciones</b>			
		Indice_entrega_atiempo_diagnostico_Pre	Indice_entrega_atiempo_diagnostico_Re
Indice_entrega_atiempo_diagnostico_Pre	Correlación de Pearson	1	,644**
	Sig. (bilateral)		,000
	N	30	30
Indice_entrega_atiempo_diagnostico_Re	Correlación de Pearson	,644**	1
	Sig. (bilateral)	,000	
	N	30	30

\*\* . La correlación es significativa en el nivel 0,01 (bilateral).

Para el segundo indicador el porcentaje de pacientes satisfechos, el resultado fue de 0.728, que según la tabla se puede concluir que el nivel de confiabilidad de la ficha de registro para este indicador es también aceptable:

**Tabla 6.** Porcentaje de pacientes satisfechos

<b>Correlaciones</b>			
		porcentaje_pacientes_satisfechos_pre	porcentaje_pacientes_satisfechos_re
porcentaje_pacientes_satisfechos_pre	Correlación de Pearson	1	,728**
	Sig. (bilateral)		,000
	N	30	30
porcentaje_pacientes_satisfechos_re	Correlación de Pearson	,728**	1
	Sig. (bilateral)	,000	
	N	30	30

\*\* . La correlación es significativa en el nivel 0,01 (bilateral).

### 3.5. Procedimientos

El procedimiento para el desarrollo de esta investigación se basa en el diseño, ya que es pre experimental y tendrá dos tiempos de evaluación, se debe realizar una primera recolección de datos antes de la implementación de la solución, ósea antes de la implementación del sistema experto, estos datos serán el cálculo de los indicadores. Luego se realizará la implementación del sistema experto y este brindará reportes de los indicadores. Para finalmente realizar la comparación entre ambos resultados y generar el método de análisis de datos.

### 3.6. Método de análisis de datos

Para el método de análisis, se realiza por medio de tres pasos, en primer lugar, el análisis descriptivo, seguido de la prueba normalidad para finalmente realizar la prueba de hipótesis.

**Análisis descriptivo:** Para Ochoa y Yunkor (2020) el análisis descriptivo es un estudio donde se toma en consideración una población con sus dimensiones espacial y temporal. Además, en este sentido de acuerdo al número de variables de estudio, se tiene a un grupo denominada univariados o analíticos que tiene como mínimo dos variables y están asociados al análisis estadístico bivariado y multivariado.

**Prueba de normalidad:** Para Fávero (2019) la prueba de normalidad se realiza en base a la cantidad de la muestra, se utiliza la prueba de Shapiro Wilk o de Kolmogorov; para el primer caso es cuando la muestra es menor igual a 50, el segundo caso es cuando la muestra es mayor a 50. Además, si el nivel de significancia es mayor igual a 0.05, entonces se adopta una institución normal o paramétrica, de lo contrario se adopta una distribución normal o no paramétrica.

#### Nivel de Significancia

Margen de error:  $X = 0.05 = 5\%$  (error)

Nivel de confiabilidad:  $1 - X = 0.95 = 95\%$

$$T = \frac{(\bar{x}_1 - \bar{x}_2)}{\sqrt{\frac{(n_1 - 1)S_1^2 + (n_2 - 1)S_2^2}{n_1 + n_2 - 2} \left(\frac{1}{n_1} + \frac{1}{n_2}\right)}}$$

**Figura 6.** Cálculo del nivel de significancia

**Prueba de hipótesis:**

**Indicador:** índice de entrega a tiempo de diagnóstico

**H1:** Un sistema experto mejora el índice de entrega a tiempo de diagnóstico en el diagnóstico de enfermedades pulmonares en el policlínico Monserrat

**Dónde:**

ETDA<sub>a</sub>: índice de entrega a tiempo de diagnóstico antes del sistema experto

ETD<sub>d</sub>: índice de entrega a tiempo de diagnóstico después del sistema experto

**Hipótesis H1<sub>0</sub>:** Un sistema experto no mejora el índice de entrega a tiempo de diagnóstico en el diagnóstico de enfermedades pulmonares en el policlínico Monserrat

$$H1_0: ETD_a - ETD_d \leq 0$$

$$H1_0: ETD_a > ETD_d$$

**Hipótesis H1<sub>a</sub>:** Un sistema experto mejora el índice de entrega a tiempo de diagnóstico en el diagnóstico de enfermedades pulmonares en el policlínico Monserrat

$$H1_a: ETD_a - ETD_d > 0$$

$$H1_a: ETD_d > ETD_a$$

## Indicador: Porcentaje de pacientes satisfechos

**H2:** Un sistema experto mejora el porcentaje de pacientes satisfechos en el diagnóstico de enfermedades pulmonares en el policlínico Monserrat.

**Dónde:**

**PPS<sub>a</sub>:** Porcentaje de pacientes satisfechos antes del sistema experto

**PPS<sub>d</sub>:** Porcentaje de pacientes satisfechos después del sistema experto

**Hipótesis H2<sub>0</sub>:** Un sistema experto no mejora el porcentaje de pacientes satisfechos en el diagnóstico de enfermedades pulmonares en el policlínico Monserrat.

$$H1_0: PPS_a > PPS_d$$

$$H1_0: PPS_a - PPS_d \leq 0$$

**Hipótesis H2<sub>a</sub>:** Un sistema experto mejora el porcentaje de pacientes satisfechos en el diagnóstico de enfermedades pulmonares en el policlínico Monserrat.

$$H1_a: PPS_a - PPS_d > 0$$

$$H1_a: PPS_d > PPS_a$$

### 3.7. Aspectos éticos

Según Hirsch y Navia (2018), indica que los aspectos éticos se refieren a los principios, reglas y valores señalados generalmente para la investigación profesional. Además, se indica la presencia de términos específicos tales como: confidencialidad, mencionar la verdad, beneficiar a comunidades y mantener un consentimiento informado.

Esta investigación es completamente original, el investigador garantiza que el título no ha sido utilizado en otra investigación u otra institución educativa.

Esta investigación contiene las citas correctamente redactadas y también correctamente referencias en la parte de la bibliografía.

Esta investigación no es una copia ni parcial y total de ninguna otra investigación.

El autor garantiza la confidencialidad de la información brindada por la empresa beneficiaria.

#### **IV. RESULTADOS**



#### 4.1. Análisis Descriptivo

A continuación, se desarrolla el análisis descriptivo donde realizamos una comparación entre los dos resultados obtenidos, antes de la implementación del software y después

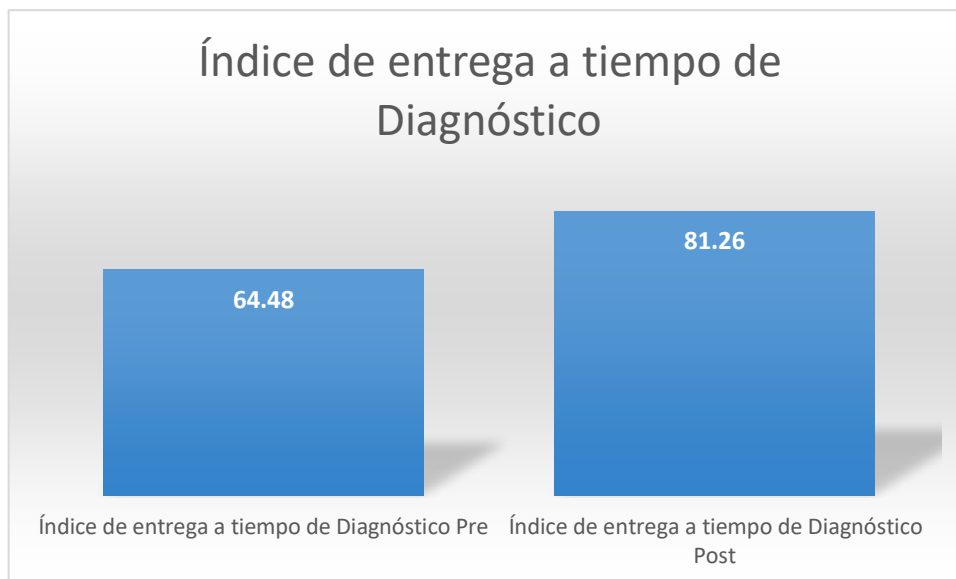
#### ÍNDICE DE ENTREGA A TIEMPO DE DIAGNÓSTICO

**Tabla 7.** Estadísticos descriptivos índice de entrega a tiempo de diagnóstico

Estadísticos descriptivos					
	N	Mínimo	Máximo	Media	Desv. Desviación
IETD_pre	30	50,00	88,89	64,4860	9,65476
IETD_post	30	63,64	91,67	81,2640	7,72313
N válido (por lista)	30				

En la tabla anterior se puede visualizar los resultados del análisis descriptivo del indicador índice de entrega a tiempo de diagnóstico, en donde el promedio antes de la implementación del sistema fue de 64.48% y después de la implementación del sistema fue un 81.26%. Además de ello se visualiza los valores mínimos y máximos que se presentaron en ambos tiempos. Y también se puede visualizar el nivel de desviación, el cual se encuentra dentro de los parámetros normales ya que no existe una diferencia muy grande de los mismos, validando esta manera que el índice de entrega a tiempo de los diagnósticos aumentó un 16.78%

En el siguiente histograma comparativo, podemos ver cómo es que se generó este aumento antes de la implementación del sistema hasta después de la implementación del mismo:



**Figura 7: Histograma comparativo del índice de entrega a tiempo de diagnóstico**

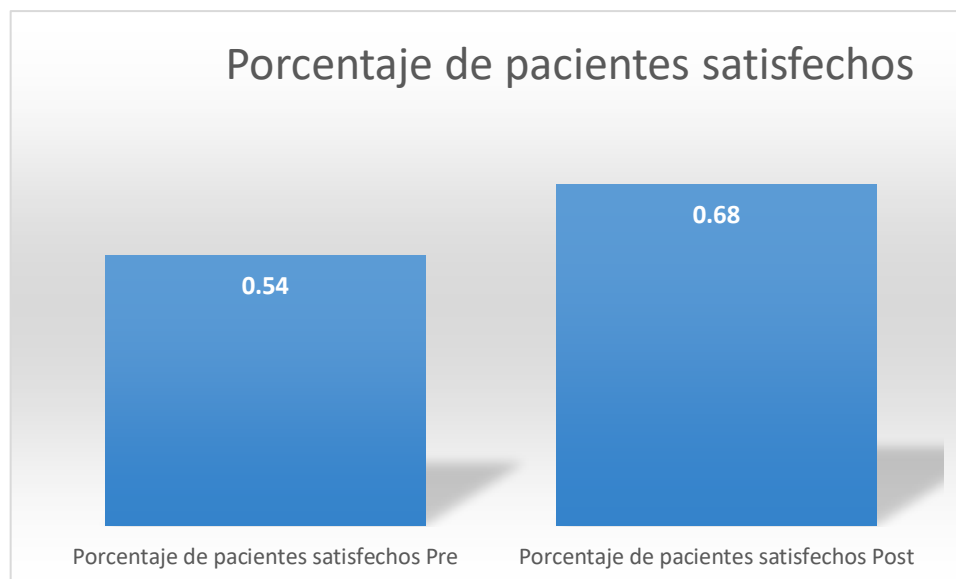
## PORCENTAJE DE PACIENTES SATISFECHOS

**Tabla 8.** Estadísticos descriptivos Porcentaje de pacientes satisfechos

Estadísticos descriptivos					
	N	Mínimo	Máximo	Media	Desv. Desviación
Satisfaccion_pre	30	27,00	90,00	53,6333	11,69581
Satisfaccion_post	30	36,00	92,00	67,8000	17,35114
N válido (por lista)	30				

En la tabla anterior se puede visualizar los resultados del análisis descriptivo del indicador porcentaje de pacientes satisfechos, en donde el promedio antes de la implementación del sistema fue de 53,63% y después de la implementación del sistema fue un 67,80%. Además de ello se visualiza los valores mínimos y máximos que se presentaron en ambos tiempos. Y también se puede visualizar el nivel de desviación, el cual se encuentra dentro de los parámetros normales ya que no existe una diferencia muy grande de los mismos, validando esta manera que el porcentaje de pacientes satisfechos aumentó un 14%

En el siguiente histograma comparativo, podemos ver cómo es que se generó este aumento antes de la implementación del sistema hasta después de la implementación del mismo:



**Figura 8: Histograma comparativo del porcentaje de pacientes satisfechos**

#### 4.2. Prueba de normalidad

Según Huarancca (2020) menciona que, para el análisis estadístico de la prueba de normalidad, se valida la muestra en caso esta sea menor a 50 individuos entonces el nivel de significancia será tomado del autor Shapiro-Wilk, de lo contrario el autor será Kolmogorov Smirnov. Adicional a ello si es que el nivel de significancia es mayor o igual a 0.05 se adopta una distribución normal o también denominada paramétrica, de lo contrario la distribución será no normal o no paramétrica

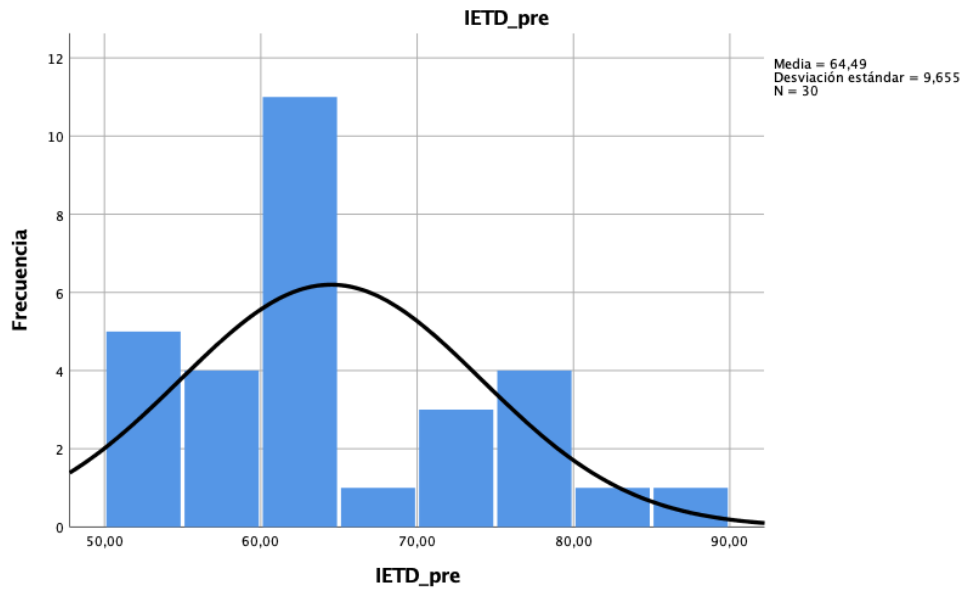
### ÍNDICE DE ENTREGA A TIEMPO DE DIAGNÓSTICO

**Tabla 9.** Prueba de normalidad índice de entrega a tiempo de diagnóstico

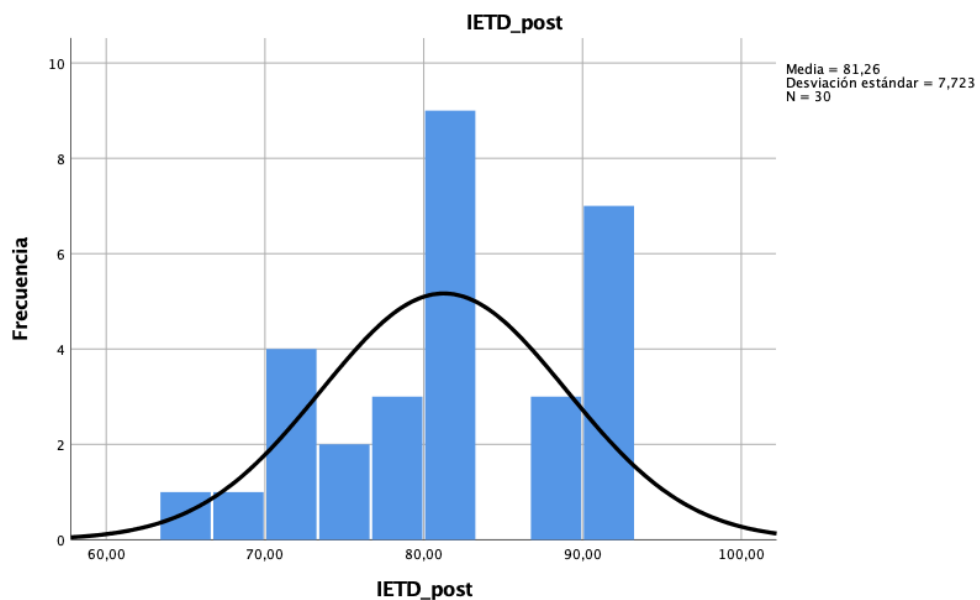
	Shapiro-Wilk		
	Estadístico	gl	Sig.
IETD_pre	0,935	5	0,067
IETD_post	0,932	5	0,056

Cómo se visualiza en la tabla anterior los niveles de significancia son mayores a 0.05 por lo tanto se adopta una distribución normal.

A continuación, se visualiza la distribución en histogramas de los resultados en ambos tiempos, antes de presentación del sistema y después del mismo:



**Figura 9: Histograma antes de la implementación del sistema del índice de entrega a tiempo de diagnóstico**



**Figura 10: Histograma después de la implementación del sistema del índice de entrega a tiempo de diagnóstico**

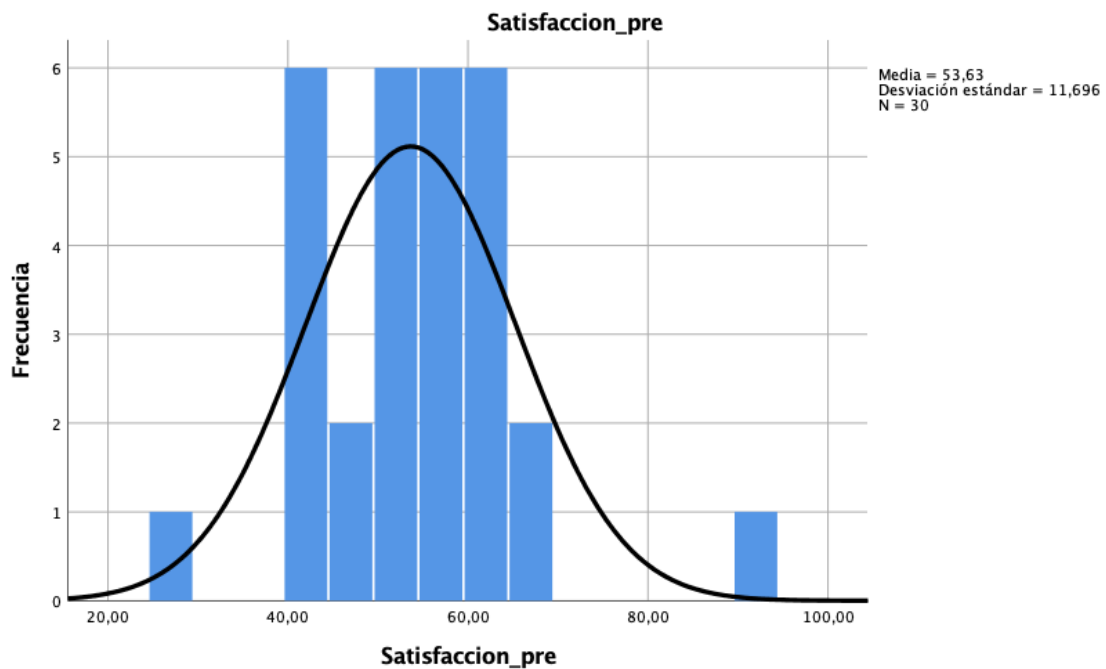
## PORCENTAJE DE PACIENTES SATISFECHOS

**Tabla 10.** Prueba de normalidad Porcentaje de pacientes satisfechos

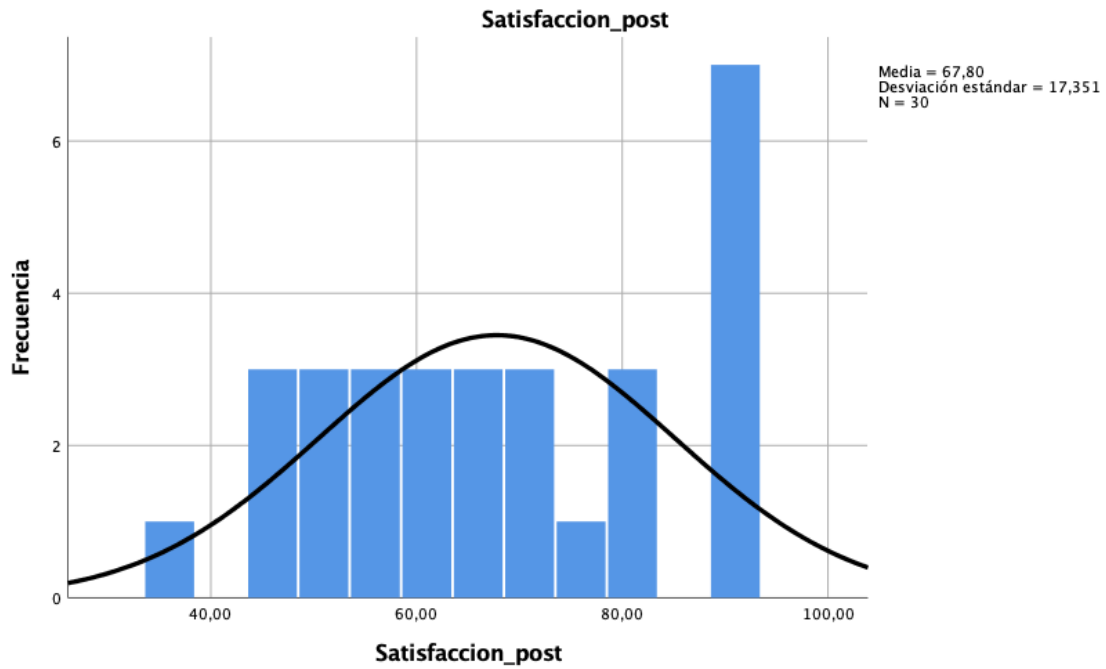
	Shapiro-Wilk		
	Estadístico	gl	Sig.
Satisfaccion_pre	0,944	5	0,114
Satisfaccion_post	0,931	5	0,053

Cómo se visualiza en la tabla anterior los niveles de significancia son mayores a 0.05 por lo tanto se adopta una distribución normal.

A continuación se visualiza la distribución en histogramas de los resultados en ambos tiempos, antes de presentación del sistema y después del mismo:



**Figura 11:** Histograma antes de la implementación del sistema del porcentaje de pacientes satisfechos



**Figura 12: Histograma después de la implementación del sistema del porcentaje de pacientes satisfechos**

#### 4.3. Prueba de hipótesis

Según Huarancca (2020) menciona que la Prueba t de Student, en estadística, es un método para probar hipótesis sobre la media de una pequeña muestra extraída de una población distribuida normalmente cuando se desconoce la desviación estándar de la población.

#### ÍNDICE DE ENTREGA A TIEMPO DE DIAGNÓSTICO

**H1:** Un sistema experto mejora el índice de entrega a tiempo de diagnóstico en el diagnóstico de enfermedades pulmonares en el policlínico Monserrat

**Dónde:**

ETD<sub>a</sub>: índice de entrega a tiempo de diagnóstico antes del sistema experto

ETD<sub>d</sub>: índice de entrega a tiempo de diagnóstico después del sistema experto

**Hipótesis H1<sub>0</sub>:** Un sistema experto no mejora el índice de entrega a tiempo de diagnóstico en el diagnóstico de enfermedades pulmonares en el policlínico Monserrat

$$H1_0: ETD_a - ETD_d \leq 0$$

$$H1_0: ETD_a > ETD_d$$

**Hipótesis H1<sub>a</sub>:** Un sistema experto mejora el índice de entrega a tiempo de diagnóstico en el diagnóstico de enfermedades pulmonares en el policlínico Monserrat

$$H1_a: ETD_a - ETD_d > 0$$

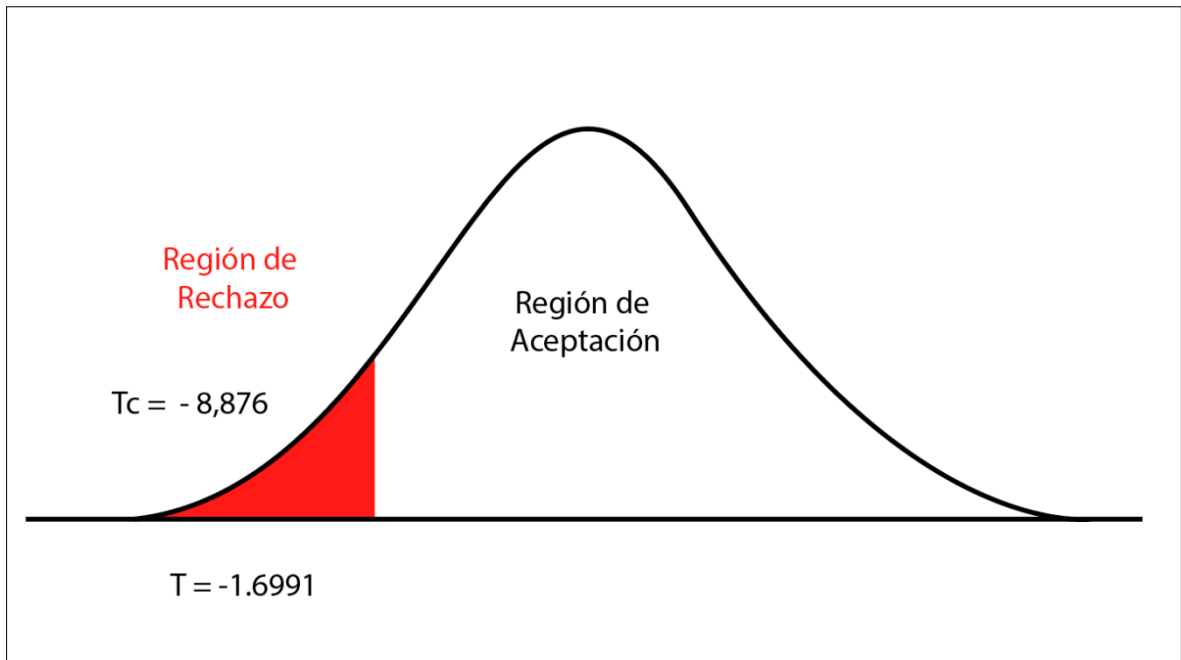
$$H1_a: ETD_d > ETD_a$$

**Tabla 11.** Prueba t-student índice de entrega a tiempo de diagnóstico

		Media	Desv. Desviación	t	gl	Sig. (bilateral)
Par 1	IETD_pre - IETD_post	- 16,7780 00	10,352584	- 8,8767 10	29	,000

En el cuadro anterior se puede visualizar que el valor de T es de -8.8765, El cual se compara con el valor de intersección de la tabla de t-student a un 95% de confianza, este valor es de -1.6991. Entonces como el primer valor es menor al valor de intersección entonces este primer valor se encuentra en la región de rechazo, por lo tanto, se rechaza la hipótesis nula y se acepta la hipótesis alterna.

La región de rechazo se puede visualizar en la siguiente figura:



**Figura 13: Región de rechazo del índice de entrega a tiempo de diagnóstico**

Al rechazar la hipótesis nula, y aceptar la alterna entonces se puede afirmar que un sistema experto mejora el índice de entrega a tiempo de diagnóstico en el diagnóstico de enfermedades pulmonares en el policlínico Monserrat.

## **PORCENTAJE DE PACIENTES SATISFECHOS**

**H2:** Un sistema experto mejora el porcentaje de pacientes satisfechos en el diagnóstico de enfermedades pulmonares en el policlínico Monserrat.

**Dónde:**

**PPS<sub>a</sub>:** Porcentaje de pacientes satisfechos antes del sistema experto

**PPS<sub>d</sub>:** Porcentaje de pacientes satisfechos después del sistema experto

**Hipótesis H2<sub>0</sub>:** Un sistema experto no mejora el porcentaje de pacientes satisfechos en el diagnóstico de enfermedades pulmonares en el policlínico Monserrat.



$$H1_0: PPS_a > PSS_d$$

$$H1_0: PPS_a - PSS_d < = 0$$

**Hipótesis H2a:** Un sistema experto mejora el porcentaje de pacientes satisfechos en el diagnóstico de enfermedades pulmonares en el policlínico Monserrat.

$$H1_a: PPS_a - PPS_d > 0$$

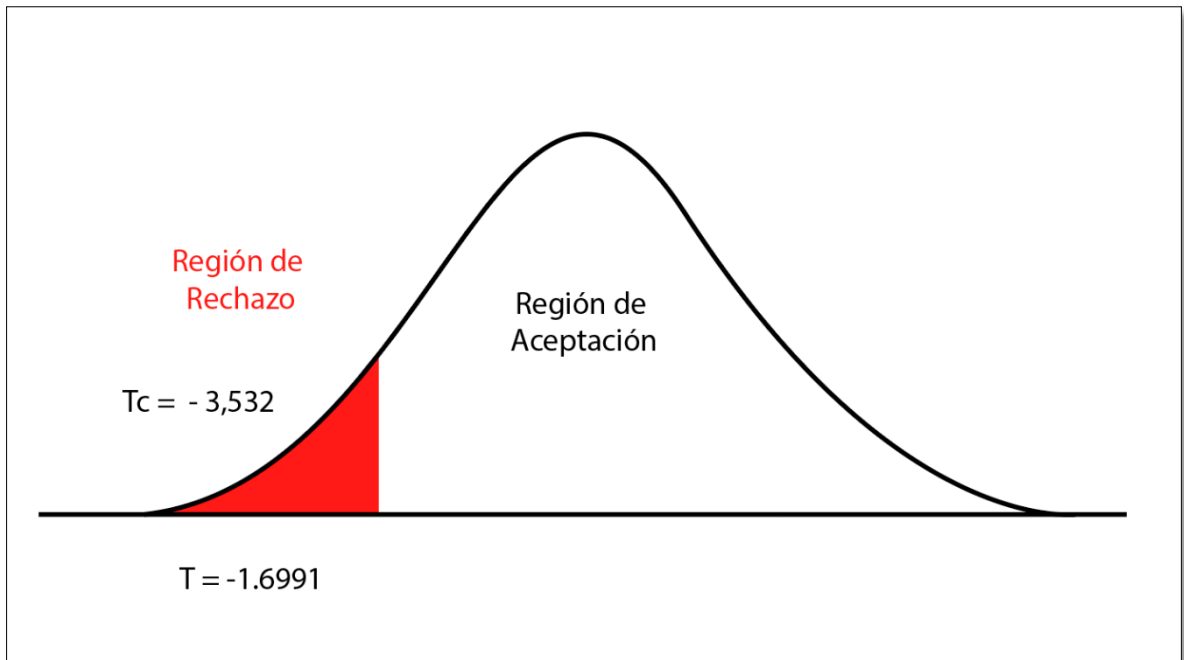
$$H1_a: PPS_d > PPS_a$$

**Tabla 12.** Prueba t-student Porcentaje de pacientes satisfechos

		Media	Desv. Desviación	t	gl	Sig. (bilateral)
Par 1	Satisfaccion_pre - Satisfaccion_post	- 14,1666 67	21,968760	- 3,5320 17	29	,001

En el cuadro anterior se puede visualizar que el valor de T es de -3,5320, El cual se compara con el valor de intersección de la tabla de t-student a un 95% de confianza, este valor es de -1.6991. Entonces como el primer valor es menor al valor de intersección entonces este primer valor se encuentra en la región de rechazo, por lo tanto, se rechaza la hipótesis nula y se acepta la hipótesis alterna.

La región de rechazo se puede visualizar en la siguiente figura:



**Figura 14: Región de rechazo del Procentaje de pacientes satisfechos**

Al rechazar la hipótesis nula, y aceptar la alterna entonces se puede afirmar que Un sistema experto mejora el porcentaje de pacientes satisfechos en el diagnóstico de enfermedades pulmonares en el policlínico Monserrat.

## V. DISCUSIÓN

Para el desarrollo del software implementado experto para el diagnóstico de enfermedades pulmonares, se llevó a cabo un estudio de sistemas expertos y su forma de trabajar, igual informa con cada uno de los antecedentes analizando de qué manera es que procesar la información y podrían generar resultados al usuario final. Para el caso de esta investigación el algoritmo que sea empleado para poder realizar diagnósticos es el algoritmo de probabilidades la cual se tiene registrada como base de conocimiento del sistema, y esta funciona en base a ingresos de información los cuales se procesan y generan posibles resultados de diagnóstico de enfermedades, en base a los síntomas, el peso, la talla y el sexo de la persona.

Al implementar este software se pudo detectar que para que el funcionamiento sea el más óptimo, los expertos deben llenar con la mayor cantidad de información y datos para que pueda generar resultados y probabilidades más exactas. Este sentido se implementaron los indicadores índice entrega a tiempo de diagnóstico, el cual permite medir que tanto tiempo una persona demora en dar un diagnóstico, en este caso se apoya en el sistema y el sistema le genera un posible diagnóstico con una posible receta, que normalmente es la más probable, Y si en caso existe algún error, entonces el experto debe intervenir para poder realizar la modificación necesaria para que en futuras consultas se tenga en cuenta esta modificación y la probabilidad aumente y así también el índice entrega a tiempo del diagnóstico mejore es decir se reduzca.

El hallazgo más importante es el hecho de que la información brindada por los expertos sea la mayor cantidad, y se contemplen la mayor cantidad de casos para de esta forma poder mejorar también el nivel de satisfacción que es el segundo indicador que se evalúa, este indicador es de suma importancia y es también utilizado en otras investigaciones que tienen relación directa con el usuario final. Si es que el diagnóstico es más exacto y rápido entonces los pacientes se sentirán más satisfechos gracias al ahorro de tiempo y una eficiente respuesta por parte del centro médico.

Ambos indicadores fueron medidos en base a porcentajes, para esto el objeto de estudio definido para ambos indicadores fue la atención. Ya que el tiempo de

evaluación fue de un mes, se sacó un promedio de las atenciones que se generaban mensualmente las cuales sean equivalentes a 1800 atenciones. De ésta se obtuvo una muestra de 317 atenciones, las cuales por conveniencia de la investigación y para que se puedan adaptar a las fichas de recolección de datos se tuvo que estratificar, en esta oportunidad se estratificó en 30, que es el valor equivalente a 30 días del mes contando de lunes a domingos.

Ya que estos indicadores permiten evaluar si es que un sistema experto puede mejorar el proceso de diagnóstico de enfermedades, se realizó una comparación entre otras investigaciones que han utilizado en los mismos indicadores, los cuales también beneficiaron a la empresa, esto se detallan a continuación:

Donde como resultados obtenidos en el desarrollo de la investigación de la presente tesis, en base al análisis descriptivo del indicador índice de entrega a tiempo de diagnóstico, en donde el promedio antes de la implementación del sistema fue de 64.48% y después de la implementación del sistema fue un 81.26%. Además de ello se visualiza los valores mínimos y máximos que se presentaron en ambos tiempos. Validando esta manera que el índice de entrega a tiempo de los diagnósticos aumentó un 16.78%.

Siendo así, en el año 2018 Franz Gupioc y Brayan Peña, desarrollaron su investigación titulada sistema experto médico para lograr la mejora del diagnóstico de pacientes con depresión en el centro médico Santa Lucía de moche. En donde utilizaron como uno de sus indicadores el tiempo promedio de diagnóstico tuvo una mejoría de un 71.12% el cual se encuentra directamente relacionado con el índice de entrega a tiempo de diagnóstico, el cual es el indicador de la actual investigación la cual tuvo un aumento de un 16.78% demostrando así que la implementación del sistema experto permitió la mejora en el índice de entrega a tiempo de diagnósticos.

Por otro lado, resultados obtenidos en el desarrollo de la investigación de la presente tesis, en base al análisis descriptivo del indicador porcentaje de pacientes satisfechos, en donde el promedio antes de la implementación del sistema fue de 53,63% y después de la implementación del sistema fue un 67,80%. Además de ello se visualiza los valores mínimos y máximos que se presentaron en ambos

tiempos. Validando esta manera que el porcentaje de pacientes satisfechos aumentó un 14%

Asimismo, en el 2018 adilla Mariluz, Santa Cruz Chamoro y Evans Jeans desarrollo de investigación titulada desarrolló un sistema experto médico para lograr el mejoramiento del diagnóstico de enfermedades gastroenterológicas en el hospital de Pampas de Tayacaja. En donde evaluaron el indicador porcentaje de pacientes satisfechos, en dónde se obtuvo un aumento del 29.2%, de la misma manera como en la actual investigación en donde este indicador también tuvo una mejora de un 14%, validando así que la implementación que la implementación de un sistema experto mejora el porcentaje de pacientes satisfechos.

En base a la observado anteriormente se puede validar que al implementar un sistema experto el cual maneja datos e información y permite automatizar de mejor forma los procesos en este caso el proceso de diagnóstico de enfermedades, se afirma que el manejo de tecnologías de información correctamente planteadas, estructuradas y con un correcto análisis pueden mejorar los procesos y generar mayor satisfacción en los usuarios finales. Para eso es muy necesario la identificación de indicadores que permitan esta medición. Para este caso se recomienda los indicadores de entregas a tiempo diagnóstico y el porcentaje de pacientes satisfechos.

## **VI. CONCLUSIONES**

La primera conclusión es que el indicador índice de entrega tiempo de diagnóstico tuvo un aumento considerable de un 16.78%. Teniendo en cuenta que al iniciar tenía un porcentaje del 64.48% y luego de la implementación del sistema experto aumentó a un 81.26%, validando esta manera que la implementación de un sistema experto mejora el índice de entrega tiempo de los diagnósticos, ya que procesa la información ingresada por los pacientes y en base a ella genera los diagnósticos y los posibles tratamientos para los pacientes.

La segunda conclusión es que, para el indicador porcentaje de pacientes satisfechos, la implementación del sistema experto también permitió la mejora de este indicador ya que en un inicio su porcentaje inicial era de 53.63% y después de su implementación un 67.8%, validando esta manera que existe un aumento del 14%. Ya que gracias a la automatización del diagnóstico y de la generación de tratamientos, el tiempo se redujo notablemente y los pacientes se vieron más satisfechas gracias a este tipo de gestión.



## **VII. RECOMENDACIONES**

Se recomienda el llenado de información y datos más detallados para diferentes casos o tipos de enfermedades, para de esta forma poder ampliar el servicio de diagnóstico y no sólo limitarse a enfermedades pulmonares.

Se recomienda el uso del sistema experto y la revisión de sus funcionalidades esperando así opiniones de mejoras para posteriormente poder implementarlas y así mejorar el proceso.

Para otros investigadores se recomienda el uso de los indicadores índice entregas a tiempo diagnóstico y porcentaje de pacientes satisfechos en caso que se desea realizar investigaciones relacionadas a sistemas expertos y al diagnóstico de enfermedades.

## REFERENCIAS

ARIAS Victor. Sistema Experto Para El Diagnóstico De Enfermedades Respiratorias Crónicas En El Distrito La Esperanza – Provincia De Trujillo. Universidad Privada del Norte. [En línea] 2018. [Fecha de consulta 20 octubre de 2021] Disponible en: <https://repositorio.upn.edu.pe/handle/11537/14972>

BERNALDO Ronel, Raymundo Lisset. Propuesta de implementación de la tecnología emergente basada en inteligencia artificial (AI) en diagnóstico médico en Hospital Cayetano Heredia. Universidad Tecnológica del Perú. [En línea] 2020 [Fecha de consulta 20 octubre de 2021] Disponible en: <https://repositorio.utp.edu.pe/handle/20.500.12867/3793>

BJORN Ole. Automation in the Virtual Testing of Mechanical Systems: Theories and Implementation Techniques. [en línea] 2018 [Fecha de consulta 26 Octubre de 2021] Disponible en: [https://books.google.com.pe/books?id=SjZIDwAAQBAJ&pg=PA16&dq=CommonKADS&hl=en&sa=X&ved=2ahUKEwiO89nQye3zAhX\\_RTABHcanCTYQ6AF6BAGKEAI#v=onepage&q=CommonKADS&f=false](https://books.google.com.pe/books?id=SjZIDwAAQBAJ&pg=PA16&dq=CommonKADS&hl=en&sa=X&ved=2ahUKEwiO89nQye3zAhX_RTABHcanCTYQ6AF6BAGKEAI#v=onepage&q=CommonKADS&f=false)

BLANCO M. Bosch R. Y Hernandez O. El padecer como diagnóstico de la enfermedad. Rev cubana med [online]. [En línea] 2019 [Fecha de consulta 20 octubre de 2021] Disponible en: [http://scielo.sld.cu/scielo.php?script=sci\\_abstract&pid=S0034-75232019000300005](http://scielo.sld.cu/scielo.php?script=sci_abstract&pid=S0034-75232019000300005)

CABELLO María. Arquitectura de Pizarrón de un sistema experto. Universidad de Colima. [En línea] 2018 [Fecha de consulta 20 octubre de 2021] Disponible en: [http://ww.ucol.mx/content/publicacionesenlinea/adjuntos/Arquitectura-de-pizarron-de-un-sistema-experto\\_449.pdf](http://ww.ucol.mx/content/publicacionesenlinea/adjuntos/Arquitectura-de-pizarron-de-un-sistema-experto_449.pdf)

CABEZAS Elaine. RECONOCIMIENTO DE PATRONES DE IMÁGENES MÉDICAS PARA ESTABLECER DIAGNÓSTICOS PREVIOS EN TRASTORNOS PULMONARES. Universidad Técnica de Ambato. [en línea] 2019 [Fecha de consulta 26 Octubre de 2021] Disponible en: [https://repositorio.uta.edu.ec/bitstream/123456789/29180/1/Tesis\\_t1533masc.pdf](https://repositorio.uta.edu.ec/bitstream/123456789/29180/1/Tesis_t1533masc.pdf)

CAPURRO D. El proceso diagnóstico. The diagnostic process [En línea] 2007 [Fecha de consulta 20 octubre de 2021] Disponible en: [https://www.scielo.cl/scielo.php?pid=S0034-98872007000400018&script=sci\\_arttext](https://www.scielo.cl/scielo.php?pid=S0034-98872007000400018&script=sci_arttext)

CARITA Fiorella. Sistema Experto Para el Diagnóstico de Paraparesia Espática Tropical Asociado al HTLV – 1. Universidad Nacional José María Arguedas. [En línea] 2019 [Fecha de consulta 20 octubre de 2021] Disponible en: [https://repositorio.unajma.edu.pe/bitstream/handle/123456789/541/Fiorella\\_Alexandra\\_Tesis\\_Bachiller\\_2019.pdf?sequence=1&isAllowed=y](https://repositorio.unajma.edu.pe/bitstream/handle/123456789/541/Fiorella_Alexandra_Tesis_Bachiller_2019.pdf?sequence=1&isAllowed=y)

CASAS, Miguel, Inferencia estadística para Economía, Editorial UNED, 2018.[ Consultado 20 de octubre de 2021]

Disponible en:

<https://books.google.com.pe/books?id=f8BjDwAAQBAJ&pg=PP1&dq=estadistica&hl=es-419&sa=X&ved=2ahUKEwiDysOa69zzAhW9JrkGHbqIBIIQ6AF6BAgCEAI#v=onepage&q=estadistica&f=false>

CORREA, Juan, Introducción a la Estadística Bayesiana, Fondo Editorial ITM, 2019.[ Consultado 21 de octubre de 2021]

Disponible en:

<https://books.google.com.pe/books?id=9wynDwAAQBAJ&pg=PR2&dq=estadistica&hl=es-419&sa=X&ved=2ahUKEwiDysOa69zzAhW9JrkGHbqIBIIQ6AF6BAgHEAI#v=onepage&q=estadistica&f=false>

DE PABLOS C. Lopez J. Romero S. Organización y transformación de los sistemas de información en la empresa. Universidad Rey Juan Carlos [en línea] 2019 [Fecha de consulta 26 Octubre de 2021] Disponible en:

<https://books.google.com.pe/books?id=hnCLDwAAQBAJ&pg=PT187&dq=sistemas+expertos&hl=en&sa=X&ved=2ahUKEwjY5sOyxO3zAhXXRTABHRmpAuI6AF6BAgGEAI#v=onepage&q=sistemas%20expertos&f=false>

DESIDERIO, Javier, Cómo hacer un perfil de proyecto de investigación científica, Editorial Palibrio 2019[ Consultado 22 de abril del 2022]

Disponible en:

<https://books.google.com.pe/books?id=QGCDwAAQBAJ&pg=PT75&dq=POBLACION+Y+MUESTRA&hl=es&sa=X&ved=2ahUKEwjCjvft97wAhWHrJUCHb2sCzQQ6AEwAnoECACQAg#v=onepage&q=POBLACION%20Y%20MUESTRA&f=false>

DIMAS A. Systematic Literature Review Of Expert System, Fuzzy Logic And Artificial Neural Network Applications. 2019. Universitas Islam Indonesia. Disponible en:

<https://dspace.uui.ac.id/bitstream/handle/123456789/16006/12522266.pdf?sequence=13&isAllowed=y>

ESCOBAR C, De Los Angeles M. Desarrollo de un software médico como apoyo al diagnóstico de enfermedades respiratorias en el consultorio médico ConsulMed. Universidad de las Fuerzas Armadas. [en línea] 2020 [Fecha de consulta 26 Octubre de 2021] Disponible en: <http://repositorio.espe.edu.ec/xmlui/handle/21000/23229>

GIRALDO, LUIS & QUINTERO, DIANA. (2015). Aplicación de la metodología Commonkads en la Gestión del Conocimiento. Revista CEA. 1. 99. 10.22430/24223182.133.

GARCIA Carlos. SISTEMA INTELIGENTE MÓVIL BASADO EN REDES NEURONALES CONVOLUCIONALES PARA EL DIAGNÓSTICO PREVIO DE ENFERMEDADES RENALES. Universidad Nacional Federico Villarreal. [en línea] 2020 [Fecha de consulta 26 Octubre de 2021] Disponible en: <http://repositorio.unfv.edu.pe/bitstream/handle/UNFV/4481/GARCIA%20CORTEGANO%20CARLOS%20ALBERTO%20-%20DOCTORADO.pdf?sequence=1&isAllowed=y>

GUPIOC Franz. Peña Brayan. Sistema Experto Médico para Mejorar el Diagnóstico de Pacientes con Depresión del CSM Santa Lucia de Moche, Año 2018. Universidad César Vallejo [en línea] 2018 [Fecha de consulta 26 Octubre de 2021] Disponible en: <https://repositorio.ucv.edu.pe/handle/20.500.12692/38474>

HOON J. Paper-Based Medical Diagnostic Devices: As a Part of Bioanalysis-Advanced Materials, Methods and devices. Springer. [en línea] 2021 [Fecha de consulta 26 Octubre de 2021] Disponible en: [https://books.google.com.pe/books?id=GxQOEAAAQBAJ&pg=PA155&dq=Medical+diagnostic&hl=es&sa=X&ved=2ahUKEwik\\_sLkx-3zAhU6VTABHTIxCOsQ6AF6BAgEEAI#v=onepage&q=Medical%20diagnostic&f=false](https://books.google.com.pe/books?id=GxQOEAAAQBAJ&pg=PA155&dq=Medical+diagnostic&hl=es&sa=X&ved=2ahUKEwik_sLkx-3zAhU6VTABHTIxCOsQ6AF6BAgEEAI#v=onepage&q=Medical%20diagnostic&f=false)

LOPEZ, Agius, Organización y transformación de los sistemas de información en la empresa [En Línea] , 2019 [ Consultado 22 de septiembre de 2021] Disponible en : <https://books.google.com.pe/books?id=hnCLDwAAQBAJ&pg=PT187&dq=sistema+experto&hl=es-419&sa=X&ved=2ahUKEwjbofXo29rzAhW5GbkGHQmiBUIQ6AF6BAgLEAI#v=onepage&q=sistema%20experto&f=false>

MAR Omar, et all. Sistema de apoyo al diagnóstico médico de COVID-19 mediante mapa cognitivo difuso. Universidad de las Ciencias Informáticas. [En línea] 2020

[Fecha de consulta 20 octubre de 2021] Disponible en:  
<http://www.revsaludpublica.sld.cu/index.php/spu/article/view/2459>

MEDRANO Hector. Diseño e implementación de un sistema experto para el diagnóstico de desnutrición en niños menores de 2 años en el área de pediatría del centro de salud Tupac Amaru – Chaupimarca. Universidad Nacional Daniel Alcides Carrión. [En línea] 2020 [Fecha de consulta 20 octubre de 2021] Disponible en:  
<http://repositorio.undac.edu.pe/handle/undac/1898>

MINISTERIO DE SALUD. Minsa brinda pautas para diferenciar la COVID-19 de otras enfermedades respiratorias [En línea] 2021 [ Consultado 21 de Octubre de 2021] Disponible en: <https://www.gob.pe/institucion/minsa/noticias/498885-minsa-brinda-pautas-para-diferenciar-la-covid-19-de-otras-enfermedades-respiratorias>

MONTERO, Lorenzo, Estadística descriptiva, Fondo Editorial THOMSOM [en línea] 2015.[ Consultado 21 de octubre de 2021]  
Disponible en: [https://books.google.com.pe/books?id=D6sj2d0xTgUC&printsec=frontcover&dq=estadística+descriptiva&hl=es-419&sa=X&redir\\_esc=y#v=onepage&q=estadística%20descriptiva&f=false](https://books.google.com.pe/books?id=D6sj2d0xTgUC&printsec=frontcover&dq=estadística+descriptiva&hl=es-419&sa=X&redir_esc=y#v=onepage&q=estadística%20descriptiva&f=false)

MIRON Adrian. Sistema de ayuda al diagnóstico médico basado en Deep Learning. Universidad autónoma de Madrid [en línea] 2019 [Fecha de consulta 26 Octubre de 2021] Disponible en:  
[https://repositorio.uam.es/bitstream/handle/10486/688782/mirón\\_cao%20\\_adrián\\_tfm.pdf?sequence=1](https://repositorio.uam.es/bitstream/handle/10486/688782/mirón_cao%20_adrián_tfm.pdf?sequence=1)

OECD. Panorama estadístico de la OCDE 2015-2016: Economía, medio ambiente y sociedad. ISBN: 9789264258402. 2016.

OMS. Chronic respiratory diseases. [en línea] 2021 [Consulta: 26 noviembre 2021] Disponible en: [https://www.who.int/health-topics/chronic-respiratory-diseases#tab=tab\\_1](https://www.who.int/health-topics/chronic-respiratory-diseases#tab=tab_1)

OPS Organización Panamericana de la Salud. La Carga de las Enfermedades Respiratorias Crónicas. [En línea] 2021 [ Consultado 21 de Octubre de 2021] Disponible en: <https://www.paho.org/es/enfermedades-no-transmisibles-salud-mental/portal-datos-enfermedades-no-transmisibles-salud-4>

ORTEGÓN E. Machicao V. Complejidad, inteligencia artificial y evolución en la gestión pública: Retos y oportunidades. Universidad Continental. [en línea] 2020 [Fecha de consulta 26 Octubre de 2021] Disponible en:  
<https://books.google.com.pe/books?id=K57SDwAAQBAJ&pg=PT68&dq=sistemas>

+expertos&hl=en&sa=X&ved=2ahUKEwiG-aWVxO3zAhVqSjABHVHwAt8Q6AF6BAgLEAI#v=onepage&q=sistemas%20expertos&f=false

PADILLA M., Santa Cruz C. Evans J. DESARROLLO DE UN SISTEMA EXPERTO MÉDICO PARA MEJORAR EL DIAGNOSTICO DEL PACIENTE EN ENFERMEDADES GASTROENTEROLOGICAS EN EL HOSPITAL DE PAMPAS DE TAYACAJA. Universidad nacional de Huancavelica [en línea] 2018 [Fecha de consulta 26 Octubre de 2021] Disponible en: <https://repositorio.unh.edu.pe/handle/UNH/2532>

PEÑA, Alvarado, La era de la transformación digital de las organizaciones, Editorial FUDA [En Línea] , 2018 [ Consultado 22 de septiembre de 2021] Disponible en : <https://books.google.com.pe/books?id=U9Q7EAAAQBAJ&pg=PA124&dq=sistema+experto&hl=es-419&sa=X&ved=2ahUKEwiq6v-75trzAhUsDrkGHS8yCyA4ChDoAXoECAQQAg#v=onepage&q=sistema%20experto&f=false>

PERTEGAL, Luisa, Alfabetización digital e informacional, Gedisa Editorial [En Línea] , 2015 [ Consultado 21 de octubre de 2021] <https://books.google.com.pe/books?id=Al3eDwAAQBAJ&pg=PP346&dq=sistemas+expertos&hl=es-419&sa=X&ved=2ahUKEwiP957bq9zzAhXtlZUCHeF1CqAQ6AF6BAgGEAI#v=onepage&q=sistemas%20expertos&f=false>

PIATTINI, Mario, Fábricas de Software: Experiencias, Tecnologías y Organización. 2ª Ed., Editorial RA-MA [En Línea] , 2015 [ Consultado 21 de octubre de 2021] [https://books.google.com.pe/books?id=Wl6fDwAAQBAJ&printsec=frontcover&dq=desarrollo+de+software&hl=es-419&sa=X&redir\\_esc=y#v=onepage&q=desarrollo%20de%20software&f=false](https://books.google.com.pe/books?id=Wl6fDwAAQBAJ&printsec=frontcover&dq=desarrollo+de+software&hl=es-419&sa=X&redir_esc=y#v=onepage&q=desarrollo%20de%20software&f=false)

PINAR del Rio. Estudio y selección de las técnicas de Inteligencia Artificial para el diagnóstico de enfermedades. Rev Ciencias Médicas vol.22 no.3 [en línea] 2018 [Fecha de consulta 26 Octubre de 2021] Disponible en: [http://scielo.sld.cu/scielo.php?script=sci\\_arttext&pid=S1561-31942018000300014](http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1561-31942018000300014)

SAINZ Leisi, et, all. Sistemas de expertos desarrollados en el mundo para la detección, el diagnóstico y el tratamiento del cáncer. Centro Provincial de Genética Médica. Pinar del río. [En línea] 2020 [Fecha de consulta 20 octubre de 2021] Disponible en: <https://www.medigraphic.com/cgi-bin/new/resumen.cgi?IDARTICULO=101295>

SALAZAR, Ludwing Probabilidad y Estadística para Bachilleratos Tecnológicos, Editorial Patria Educación , 2020.[ Consultado 21 de octubre de 2021] Disponible en:

<https://books.google.com.pe/books?id=mo55DwAAQBAJ&printsec=frontcover&dq=estadística&hl=es-419&sa=X&ved=2ahUKEwjLyYTZ7NzzAhW3lLkGHbGIAOs4ChDoAXoECAkQAg#v=onepage&q&f=false>

SCHANEN-BERGOT. El diagnóstico de las enfermedades neuromusculares: [en línea] 2019 [Consulta: 20 octubre 2021]. Disponible en: [http://www.asemgalicia.com/wp-content/uploads/Diagnostico-y-nfermedades-Neuromusculares\\_Saber-y-Entender\\_Informe-8.pdf](http://www.asemgalicia.com/wp-content/uploads/Diagnostico-y-nfermedades-Neuromusculares_Saber-y-Entender_Informe-8.pdf)

SERRANO Juan. Prototipo De Aplicación Móvil Para La Identificación De Mazorcas De Cacao Enfermas Haciendo Uso De Visión Por Computadora Y Aprendizaje De Máquina. Universidad Autónoma de Bucaramanga. [En línea] 2020 [Fecha de consulta 20 octubre de 2021] Disponible en: [https://repository.unab.edu.co/bitstream/handle/20.500.12749/13367/2020\\_Tesis\\_Juan\\_Sebastian\\_Serrano\\_Arenas.pdf?sequence=1&isAllowed=y](https://repository.unab.edu.co/bitstream/handle/20.500.12749/13367/2020_Tesis_Juan_Sebastian_Serrano_Arenas.pdf?sequence=1&isAllowed=y)

SOCIEDAD PERUANA DE NEUMOLOGÍA. Recursos COVID-19 [en línea] 2020 [Fecha de consulta 26 Octubre de 2021] Disponible en: <http://www.spneumologia.org.pe/index.php/cientificas/covid-19>

TAYPE W. Chicas L, De la Cruz L. Amado J. Tiempo de espera para atención médica urgente en un hospital terciario después de implementar un programa de mejora de procesos. An. Fac. med. vol.80 no.4 Lima oct./dic 2019 [en línea] 2019 [Fecha de consulta 26 Octubre de 2021] Disponible en: [http://www.scielo.org.pe/scielo.php?script=sci\\_arttext&pid=S1025-55832019000400005](http://www.scielo.org.pe/scielo.php?script=sci_arttext&pid=S1025-55832019000400005)

TUNDIDOR , Alberto Cadena de suministro 4.0, Marge Books [En Línea] , 2020[ Consultado [ 22 de abril del 2022] Disponible en : <https://books.google.com.pe/books?id=LjB8DwAAQBAJ&printsec=frontcover&dq=tecnologías+en+las+empresas&hl=es-419&sa=X&ved=2ahUKEwjVxtqjrj3AhWvJrkGHc-pAywQ6AF6BAgDEAI#v=onepage&q&f=false>

UTPL. Universidad Técnica Particular de Loja. ¿Qué es el diagnóstico clínico y cuál es su importancia? [En línea] 2018 [Fecha de consulta 20 octubre de 2021] Disponible en: <https://noticias.utpl.edu.ec/que-es-el-diagnostico-clinico-y-cual-es-su-importancia>



VASCONCELOS, Santillán, Introducción a la Computación, Editoria Patia Educación [En Línea] , 2019[ Consultado 22 de septiembre de 2021]

Disponible en:

<https://books.google.com.pe/books?id=tI55DwAAQBAJ&pg=PA175&dq=sistema+experto&hl=es-419&sa=X&ved=2ahUKEwjnhdzw29rzAhWzK7kGHQ4YCQEQ6AF6BAglEAI#v=onepage&q=sistema%20experto&f=false>

## **ANEXOS**

### Anexo 1: Matriz de consistencia

Problemas	Objetivos	Hipótesis	Variables	Variable Dependiente				Métodos
<b>Principal</b>	<b>General</b>	<b>General</b>	<b>Independiente</b>					<b>Tipo de investigación</b> Aplicada
¿De qué forma un sistema experto influye en el diagnóstico de enfermedades pulmonares en el policlínico Monserrat?	Determinar de qué forma un sistema experto influye en el diagnóstico de enfermedades pulmonares en el policlínico Monserrat	Un sistema experto mejora el diagnóstico de enfermedades pulmonares en el policlínico Monserrat	Sistema experto					
<b>Específicos</b>	<b>Específicos</b>	<b>Específicos</b>	<b>Dependiente</b>	<b>Dimensiones</b>	<b>Indicadores</b>	<b>Instrumento</b>	<b>Formula</b>	<b>Diseño de Investigación</b> Experimental, Pre experimental
¿De qué forma un sistema experto influye en el índice de entrega a tiempo de diagnóstico en el diagnóstico de enfermedades pulmonares en el policlínico Monserrat?	Determinar de qué forma un sistema experto influye en el índice de entrega a tiempo de diagnóstico en el diagnóstico de enfermedades pulmonares en el policlínico Monserrat	Un sistema experto mejora el índice de entrega a tiempo de diagnóstico en el diagnóstico de enfermedades pulmonares en el policlínico Monserrat	Diagnóstico de enfermedades pulmonares	Verificación del diagnóstico	índice de entrega a tiempo de diagnóstico	Ficha	$IETD = \frac{NDET}{NDE} \times 100$	<b>Población</b> 1800 atenciones
¿De qué forma un sistema experto influye en el porcentaje de pacientes satisfechos en el diagnóstico de enfermedades pulmonares en el policlínico Monserrat?	Determinar de qué forma un sistema experto influye en el porcentaje de pacientes satisfechos en el diagnóstico de enfermedades pulmonares en el policlínico Monserrat.	Un sistema experto mejora el porcentaje de pacientes satisfechos en el diagnóstico de enfermedades pulmonares en el policlínico Monserrat.			Porcentaje de pacientes satisfechos	Ficha	$PPS = \frac{NPS}{TPA} \times 100$	<b>Muestra</b> 317 atenciones agrupadas en 30

## Anexo 2: Operacionalización de las variables

Tipo	Variable	Definición Conceptual	Definición Operacional	Dimensión	Indicador	Escala de Medición
Independiente	Sistema experto	López (2019), Un sistema experto es un programa desarrollado para computadoras que permite solucionar problemas generales tratando de emular el comportamiento humano por medio del conocimiento computacional con inteligencia artificial y un dominio	Un sistema experto, es aquella aplicación tecnológica que permitirá mejorar el proceso de diagnóstico de enfermedades, por medio de una base de conocimiento, el cual permitirá brindar las hipótesis de las posibles enfermedades que pueda tener el paciente, en base a los síntomas detectados.			
Dependiente	Diagnóstico de enfermedades pulmonares	Para Schanen-Bergot (2017) el diagnóstico es un proceso por el cual a través de una consulta especializada se obtiene la confirmación en la enfermedad o discapacidad del paciente, dependiendo del tipo de paciente los diagnósticos pueden darse en un tiempo más prolongado.	El proceso de diagnóstico de enfermedades se realizará por medio del sistema experto, el cual servirá de apoyo ya que este posee una serie de datos los cuales serán ingresados por los expertos, para poder automatizar y agilizar el proceso.	Verificación del diagnóstico	Índice de entrega a tiempo de diagnóstico	Razón
					Índice de entrega a tiempo de diagnóstico	Razón

### Anexo 3: Indicadores de diagnóstico de enfermedades pulmonares

DIMENSIÓN	INDICADOR	DESCRIPCIÓN	TÉCNICA	INSTRUMENTO	UNIDAD DE MEDIDA	FÓRMULA
Verificación del diagnóstico	Índice de entrega a tiempo de diagnóstico	Guzmán et al. (2015) menciona que la acción de cumplir los plazos de entrega de los resultados responde a la calidad que el paciente o médico perciba, ya que al brindar los resultados en los plazos adecuados puede tener gran implicancia en el cuidado del paciente.	Fichaje	Ficha	Porcentaje	$IETD = \frac{NDET}{NDE} \times 100$ <p>Dónde IETD representa al Índice de entrega a tiempo de Diagnóstico; NDET representa al: Número de diagnósticos Entregados a Tiempo y el NDE es el: Número de Diagnósticos Entregados</p>
	Porcentaje de pacientes satisfechos	Arenal (2020) este indicador permite la medición de la satisfacción de los pacientes atendidos, respecto al servicio que se le ha brindado por parte de la institución médica, permite medir la calidad de la atención en general que se les da a los pacientes.	Fichaje	Ficha	Porcentaje	$PPS = \frac{NPS}{TPA} \times 100$ <p>Dónde PPS representa al Porcentaje de pacientes satisfechos; NPS representa al: Número de pacientes satisfechos y TPA representa al: Total de pacientes atendidos</p>

**Anexo 4: Instrumento Ficha de Registro Índice de entrega a tiempo de Diagnóstico del mes de septiembre**

Ficha de Registro				
<b>Investigador</b>	NIZAMA ANTUNEZ, GIANMARCO ROGER	<b>Tipo de Prueba</b>		Pre test
<b>Empresa investigada</b>	Policlínico Monserrat			
<b>Motivo de Investigación</b>	Índice de entrega a tiempo de Diagnóstico			
<b>Fecha Inicio</b>	01 sept	<b>Fecha fin</b>	30 sept	
<b>Variable</b>	<b>Indicador</b>	<b>Medida</b>	<b>Fórmula</b>	
Diagnóstico de enfermedades pulmonares	Índice de entrega a tiempo de Diagnóstico	Porcentaje	IETD= (NDET /NDE) x100	
<b>Item</b>	<b>Fecha</b>	<b>Número de diagnósticos Entregados a Tiempo</b>	<b>Número de Diagnósticos Entregados</b>	<b>Índice de entrega a tiempo de Diagnóstico</b>
1	01 sept	6	10	60,00
2	02 sept	7	9	77,78
3	3 sept	7	11	63,64
4	4 sept	8	10	80,00
5	5 sept	7	10	70,00
6	6 sept	7	9	77,78
7	7 sept	6	12	50,00
8	8 sept	7	11	63,64
9	9 sept	7	9	77,78
10	10 sept	8	9	88,89
11	11 sept	7	10	70,00
12	12 sept	6	11	54,55
13	13 sept	7	12	58,33
14	14 sept	6	12	50,00
15	15 sept	7	11	63,64
16	16 sept	7	12	58,33
17	17 sept	7	11	63,64
18	18 sept	6	10	60,00
19	19 sept	7	10	70,00
20	20 sept	7	11	63,64
21	21 sept	7	12	58,33
22	22 sept	6	12	50,00
23	23 sept	7	12	58,33

24	24 sept	6	10	60,00
25	25 sept	7	11	63,64
26	26 sept	6	10	60,00
27	27 sept	7	11	63,64
28	28 sept	6	11	54,55
29	29 sept	6	9	66,67
30	30 sept	7	9	77,78

317

64,48

**Anexo 5: Instrumento Ficha de Registro Índice de entrega a tiempo de Diagnóstico del mes de octubre**

Ficha de Registro				
<b>Investigador</b>	NIZAMA ANTUNEZ, GIANMARCO ROGER		<b>Tipo de Prueba</b>	Re test
<b>Empresa investigada</b>	Policlínico Monserrat			
<b>Motivo de Investigación</b>	Índice de entrega a tiempo de Diagnóstico			
<b>Fecha Inicio</b>	01 octubre	<b>Fecha fin</b>	30 octubre	
<b>Variable</b>	<b>Indicador</b>	<b>Medida</b>	<b>Fórmula</b>	
Diagnóstico de enfermedades pulmonares	Índice de entrega a tiempo de Diagnóstico	Porcentaje	IETD= (NDET /NDE) x100	
<b>Item</b>	<b>Fecha</b>	<b>Número de diagnósticos Entregados a Tiempo</b>	<b>Número de Diagnósticos Entregados</b>	<b>Índice de entrega a tiempo de Diagnóstico</b>
1	01-oct	6	10	60,00
2	02-oct	6	9	66,67
3	03-oct	7	10	70,00
4	04-oct	7	10	70,00
5	05-oct	7	9	77,78
6	06-oct	7	9	77,78
7	07-oct	7	11	63,64
8	08-oct	7	11	63,64
9	09-oct	8	9	88,89
10	10-oct	8	10	80,00
11	11-oct	6	10	60,00
12	12-oct	6	11	54,55
13	13-oct	7	11	63,64
14	14-oct	6	12	50,00
15	15-oct	7	11	63,64
16	16-oct	6	12	50,00
17	17-oct	7	10	70,00
18	18-oct	6	10	60,00
19	19-oct	6	9	66,67
20	20-oct	7	12	58,33
21	21-oct	7	11	63,64
22	22-oct	8	12	66,67



23	23-oct	7	12	58,33
24	24-oct	6	11	54,55
25	25-oct	8	11	72,73
26	26-oct	6	10	60,00
27	27-oct	7	12	58,33
28	28-oct	7	11	63,64
29	29-oct	6	10	60,00
30	30-oct	7	11	63,64

317

64,56

**Anexo 6: Instrumento Ficha de Registro Índice de entrega a tiempo de Diagnóstico del mes de marzo**

Ficha de Registro				
<b>Investigador</b>	NIZAMA ANTUNEZ, GIANMARCO ROGER	<b>Tipo de Prueba</b>	Pos test	
<b>Empresa investigada</b>	Policlínico Monserrat			
<b>Motivo de Investigación</b>	Índice de entrega a tiempo de Diagnóstico			
<b>Fecha Inicio</b>	01 marzo	<b>Fecha fin</b>	30 marzo	
<b>Variable</b>	<b>Indicador</b>	<b>Medida</b>	<b>Fórmula</b>	
Diagnóstico de enfermedades pulmonares	Índice de entrega a tiempo de Diagnóstico	Porcentaje	IETD= (NDET /NDE) x100	
<b>Item</b>	<b>Fecha</b>	<b>Número de diagnósticos Entregados a Tiempo</b>	<b>Número de Diagnósticos Entregados</b>	<b>Índice de entrega a tiempo de Diagnóstico</b>
1	01-mar	7	11	63,64
2	02-mar	9	10	90,00
3	03-mar	8	10	80,00
4	04-mar	8	9	88,89
5	05-mar	7	9	77,78
6	06-mar	7	9	77,78
7	07-mar	8	11	72,73
8	08-mar	8	11	72,73
9	09-mar	7	9	77,78
10	10-mar	8	10	80,00
11	11-mar	9	11	81,82
12	12-mar	10	11	90,91
13	13-mar	8	9	88,89
14	14-mar	8	12	66,67
15	15-mar	8	11	72,73
16	16-mar	10	12	83,33
17	17-mar	9	10	90,00
18	18-mar	9	11	81,82
19	19-mar	8	9	88,89
20	20-mar	11	12	91,67
21	21-mar	9	11	81,82
22	22-mar	9	12	75,00
23	23-mar	9	11	81,82

24	24-mar	9	11	81,82
25	25-mar	8	10	80,00
26	26-mar	9	10	90,00
27	27-mar	8	11	72,73
28	28-mar	9	12	75,00
29	29-mar	9	10	90,00
30	30-mar	11	12	91,67

317

81,26

**Anexo 7: Instrumento Ficha de Registro Porcentaje de pacientes satisfechos del mes de septiembre**

Ficha de Registro					
<b>Investigador</b>	NIZAMA ANTUNEZ, GIANMARCO ROGER		<b>Tipo de Prueba</b>	Pre test	
<b>Empresa investigada</b>	Policlínico Monserrat				
<b>Motivo de Investigación</b>	Porcentaje de pacientes satisfechos				
<b>Fecha Inicio</b>	01 octubre		<b>Fecha fin</b>	30 septiembre	
<b>Variable</b>	<b>Indicador</b>	<b>Medida</b>	<b>Fórmula</b>		
Diagnóstico de enfermedades pulmonares	Porcentaje de pacientes satisfechos	Porcentaje	PPS= NPS/TPA x100		
<b>Item</b>	<b>Fecha</b>	<b>Número de pacientes satisfechos</b>	<b>Total de pacientes atendidos</b>	<b>Porcentaje de pacientes satisfechos</b>	
1	1 sept	9	10	0,90	
2	2 sept	4	9	0,44	
3	3 sept	5	11	0,45	
4	4 sept	4	10	0,40	
5	5 sept	4	10	0,40	
6	6 sept	4	9	0,44	
7	7 sept	8	12	0,67	
8	8 sept	3	11	0,27	
9	9 sept	4	9	0,44	
10	10 sept	5	9	0,56	
11	11 sept	5	10	0,50	
12	12 sept	6	11	0,55	
13	13 sept	5	12	0,42	
14	14 sept	6	12	0,50	
15	15 sept	7	11	0,64	
16	16 sept	6	12	0,50	
17	17 sept	7	11	0,64	
18	18 sept	6	10	0,60	
19	19 sept	5	10	0,50	
20	20 sept	6	11	0,55	
21	21 sept	7	12	0,58	
22	22 sept	6	12	0,50	

23	23 sept	7	12	0,58
24	24 sept	5	10	0,50
25	25 sept	7	11	0,64
26	26 sept	6	10	0,60
27	27 sept	7	11	0,64
28	28 sept	5	11	0,45
29	29 sept	5	9	0,56
30	30 sept	6	9	0,67

317

0,54

**Anexo 8: Instrumento Ficha de Registro Porcentaje de pacientes satisfechos del mes de octubre**

Ficha de Registro				
<b>Investigador</b>	NIZAMA ANTUNEZ, GIANMARCO ROGER	<b>Tipo de Prueba</b>	Re test	
<b>Empresa investigada</b>	Policlínico Monserrat			
<b>Motivo de Investigación</b>	Porcentaje de pacientes satisfechos			
<b>Fecha Inicio</b>	01 octubre	<b>Fecha fin</b>	30 octubre	
<b>Variable</b>	<b>Indicador</b>	<b>Medida</b>	<b>Fórmula</b>	
Diagnóstico de enfermedades pulmonares	Porcentaje de pacientes satisfechos	Porcentaje	PPS= NPS/TPA x100	
<b>Item</b>	<b>Fecha</b>	<b>Número de pacientes satisfechos</b>	<b>Total, de pacientes atendidos</b>	<b>Porcentaje de pacientes satisfechos</b>
1	01-oct	4	10	0,40
2	02-oct	4	9	0,44
3	03-oct	4	10	0,40
4	04-oct	4	10	0,40
5	05-oct	4	9	0,44
6	06-oct	5	9	0,56
7	07-oct	5	11	0,45
8	08-oct	3	11	0,27
9	09-oct	5	9	0,56
10	10-oct	5	10	0,50
11	11-oct	4	10	0,40
12	12-oct	6	11	0,55
13	13-oct	5	11	0,45
14	14-oct	5	12	0,42
15	15-oct	7	11	0,64
16	16-oct	6	12	0,50
17	17-oct	6	10	0,60
18	18-oct	6	10	0,60
19	19-oct	5	9	0,56
20	20-oct	6	12	0,50
21	21-oct	6	11	0,55
22	22-oct	6	12	0,50

23	23-oct	7	12	0,58
24	24-oct	4	11	0,36
25	25-oct	7	11	0,64
26	26-oct	5	10	0,50
27	27-oct	7	12	0,58
28	28-oct	5	11	0,45
29	29-oct	4	10	0,40
30	30-oct	6	11	0,55

317

0,49

**Anexo 9: Instrumento Ficha de Registro Porcentaje de pacientes satisfechos del mes de marzo**

Ficha de Registro				
<b>Investigador</b>	NIZAMA ANTUNEZ, GIANMARCO ROGER	<b>Tipo de Prueba</b>	Pre test	
<b>Empresa investigada</b>	Policlínico Monserrat			
<b>Motivo de Investigación</b>	Porcentaje de pacientes satisfechos			
<b>Fecha Inicio</b>	01 marzo	<b>Fecha fin</b>	30 marzo	
<b>Variable</b>	<b>Indicador</b>	<b>Medida</b>	<b>Fórmula</b>	
Diagnóstico de enfermedades pulmonares	Porcentaje de pacientes satisfechos	Porcentaje	PPS= NPS/TPA x100	
<b>Item</b>	<b>Fecha</b>	<b>Número de pacientes satisfechos</b>	<b>Total de pacientes atendidos</b>	<b>Porcentaje de pacientes satisfechos</b>
1	01-mar	5	11	0,45
2	02-mar	7	10	0,70
3	03-mar	6	10	0,60
4	04-mar	5	9	0,56
5	05-mar	4	9	0,44
6	06-mar	8	9	0,89
7	07-mar	6	11	0,55
8	08-mar	10	11	0,91
9	09-mar	6	9	0,67
10	10-mar	7	10	0,70
11	11-mar	9	11	0,82
12	12-mar	6	11	0,55
13	13-mar	4	9	0,44
14	14-mar	6	12	0,50
15	15-mar	10	11	0,91
16	16-mar	11	12	0,92
17	17-mar	5	10	0,50
18	18-mar	4	11	0,36
19	19-mar	6	9	0,67
20	20-mar	9	12	0,75
21	21-mar	7	11	0,64
22	22-mar	11	12	0,92
23	23-mar	9	11	0,82
24	24-mar	8	11	0,73



25	25-mar	6	10	0,60
26	26-mar	9	10	0,90
27	27-mar	10	11	0,91
28	28-mar	6	12	0,50
29	29-mar	6	10	0,60
30	30-mar	10	12	0,83

317

0,68

## Anexo 10: Validación del Instrumento

### VALIDACION DEL INSTRUMENTO

#### I. DATOS GENERALES

Apellidos y Nombres del Experto: Liendo Arévalo Milner David

Título y/o Grado Académico:

Doctor ( )    Magister ( x )    Ingeniero ( )    Licenciado ( )

Universidad que labora:

Universidad César Vallejo

Fecha:

23/11/21

**TESIS: Sistema experto para diagnóstico de enfermedades pulmonares en el policlínico  
Monserrat**

**Autores: Nizama Antunez, Gianmarco Roger**

**Deficiente (0-20%)    Regular(21-50%)    Bueno(51-70%)    Muy Bueno(71-80%)    Excelente(81-100%)**  
Mediante la evaluación de expertos usted tiene la facultad de calificar la tabla de validación del instrumento involucradas mediante una serie de indicadores con puntuaciones especificadas en la tabla, con la valoración de 0% - 100%. Asimismo, se exhorta a las sugerencias de cambio de ítems que crea pertinente, con la finalidad de mejorar la coherencia de los indicadores para su valoración.

#### II. ASPECTOS DE VALIDACIÓN

INDICADOR	CRITERIO	VALORACIÓN				
		0-20%	21-50%	51-70%	71-80%	81-100%
CLARIDAD	Es formulado con lenguaje apropiado.				80%	
OBJETIVIDAD	Esta expresado en conducta observable.				80%	
ACTUALIDAD	Es adecuado el avance, la ciencia y tecnología.				80%	
ORGANIZACIÓN	Existe una organización lógica.				80%	
SUFICIENCIA	Comprende los aspectos de cantidad y calidad.				80%	
INTENCIONALIDA	Adecuado para valorar los aspectos del sistema metodológico y científico.				80%	
CONSISTENCIA	Está basado en aspectos teóricos y científicos.				80%	
COHERENCIA	En los datos respecto al indicador.				80%	
METODOLOGÍA	Responde al propósito de investigación.				80%	
PERTENENCIA	El instrumento es adecuado al tipo de investigación.				80%	
	TOTAL					

#### III. PROMEDIO DE VALIDACIÓN

#### IV. OPCIÓN DE APLICABILIDAD

- ( ) El instrumento puede ser aplicado, tal como está elaborado
- ( ) El instrumento debe ser mejorado antes de ser aplicado

FIRMA DEL EXPERTO



## VALIDACION DEL INSTRUMENTO

### I. DATOS GENERALES

Apellidos y Nombres del Experto: ARADIEL CASTAÑEDA, HILARIO

Título y/o Grado Académico:

Doctor (X)    Magister (X)    Ingeniero ( )    Licenciado ( )

Universidad que labora:

Universidad César Vallejo

Fecha:

23/11/21

**TESIS: Sistema experto para diagnóstico de enfermedades pulmonares en el policlínico  
Monserrat**

**Autores: Nizama Antunez, Gianmarco Roger**

**Deficiente (0-20%) Regular (21-50%) Bueno (51-70%) Muy Bueno (71-80%) Excelente (81-100%)**  
Mediante la evaluación de expertos usted tiene la facultad de calificar la tabla de validación del instrumento involucradas mediante una serie de indicadores con puntuaciones especificadas en la tabla, con la valoración de 0% - 100%. Asimismo, se exhorta a las sugerencias de cambio de ítems que crea pertinente, con la finalidad de mejorar la coherencia de los indicadores para su valoración.

### II. ASPECTOS DE VALIDACIÓN

INDICADOR	CRITERIO	VALORACIÓN				
		0-20%	21-50%	51-70%	71-80%	81-100%
CLARIDAD	Es formulado con lenguaje apropiado.					85%
OBJETIVIDAD	Esta expresado en conducta observable.					85%
ACTUALIDAD	Es adecuado el avance, la ciencia y tecnología.					85%
ORGANIZACIÓN	Existe una organización lógica.					85%
SUFICIENCIA	Comprende los aspectos de cantidad y calidad.					85%
INTENCIONALIDA	Adecuado para valorar los aspectos del sistema metodológico y científico.					85%
CONSISTENCIA	Está basado en aspectos teóricos y científicos.					85%
COHERENCIA	En los datos respecto al indicador.					85%
METODOLOGÍA	Responde al propósito de investigación.					85%
PERTENENCIA	El instrumento es adecuado al tipo de investigación.					85%
	TOTAL					85%

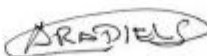
### III. PROMEDIO DE VALIDACIÓN

### IV. OPCIÓN DE APLICABILIDAD

( X ) El instrumento puede ser aplicado, tal como está elaborado

( ) El instrumento debe ser mejorado antes de ser aplicado

FIRMA DEL EXPERTO



## Anexo 11: Sistema

### Inicio de sesión



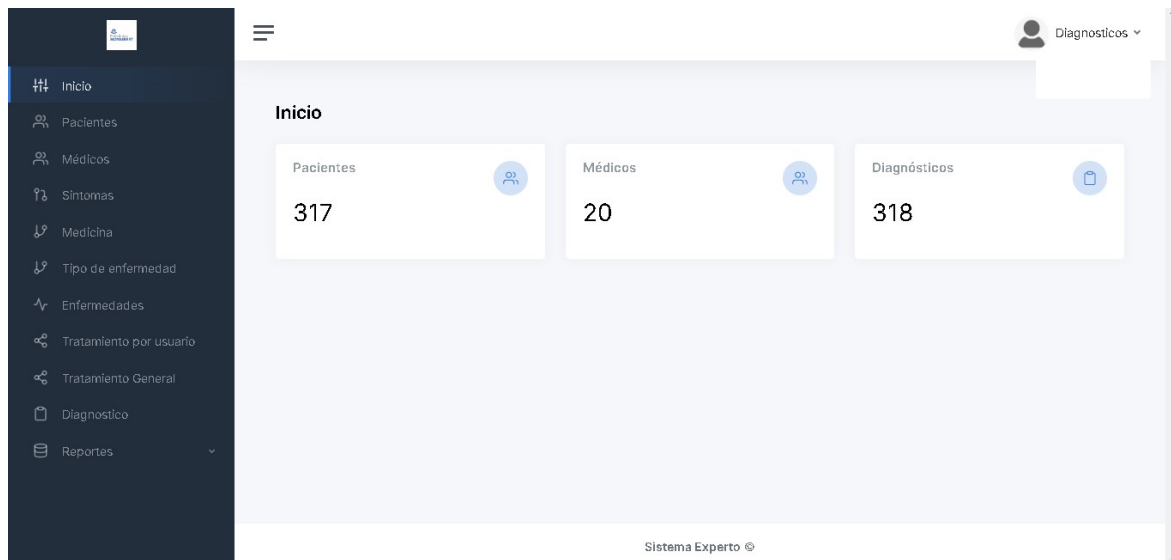


**Bienvenido a tu panel administrativo de Sistema Experto**

Administra, controla y organiza tu aplicación.

[INICIAR SESIÓN](#)

### Dashboard del Sistema



**Inicio**

Pacientes 317	Médicos 20	Diagnósticos 318
------------------	---------------	---------------------

Sistema Experto ©

Diagnósticos ▾

- Inicio
- Pacientes
- Médicos
- Síntomas
- Medicina
- Tipo de enfermedad
- Enfermedades
- Tratamiento por usuario
- Tratamiento General
- Diagnostico
- Reportes

## Registro de Pacientes

**Pacientes** Agregar Paciente

EXCEL PDF BUSCAR:

#	DNI	Nombres y Apellidos	Teléfono	Fecha de nacimiento	Acciones
1	72584632	Sandra Lorena Barahona Zegarra	952014588	28-11-2004	
2	72658410	Anfitribe Emilia Vega Gutierrez	963025014	12-12-2000	
3	72650419	America Paulina Sotelo Bazán	952985199	17-05-1997	
4	72540021	Cristian Manuel Terreros Mesa	975841252	17-02-2004	
5	72015483	Anitani Flavia Asai Miranda	960205748	28-02-1994	
6	72054243	Yasuri Alexa Cova Casas	958258844	30-05-2003	
7	72412502	Yamilet Yahaira Sliuce Suxe	920054723	31-10-2004	

## Registro de Médicos

**Médicos** Agregar Médico

EXCEL PDF BUSCAR:

#	DNI	Nombres y Apellidos	Teléfono	Especialidad	Acciones
1	09874214	Yired Felicit Estelo Garrido	988742514	Apnea del sueño	
2	09632563	Johan Renato Huamán Oclia	901285699	Asma bronquial	
3	09582145	Valdiviano Emiliano Montoya Prado	952001487	Neumonía	
4	09542155	Jonas Gabriel Quevedo Urieta	952147820	Covid	
5	09524120	Liliana Blanca Pulgar Coloma	985238856	Cáncer de pulmón	
6	09524781	Kimberly Aurora Jasmínez Céspedes	995200145	Neumotorax	
7	09582145	Humberto Aaron Fernández Damosa	962514670	Fibrosis quística	

## Registro de Síntomas

**Síntomas** Agregar Síntoma

Tipo de Enfermedad: TODOS

EXCEL PDF BUSCAR:

#	Nombre	Tipo	Tipo de Enfermedad	Acciones
1	Sudor nocturno	Si/No	PULMONAR	
2	Estornudos	Opciones	PULMONAR	
3	Ojos	Opciones	PULMONAR	
4	Falta de apetito	Opciones	PULMONAR	
5	Dolor corporal	Opciones	PULMONAR	
6	Acropaquia (ensanchamiento y redondeo de los dedos)	Si/No	PULMONAR	

## Registro de Medicina

**Medicina** Agregar Medicina

EXCEL PDF BUSCAR:

#	Nombre	Acciones
1	Dexametasona	
2	Paracetamol x500	

2 de 2 registros

Anterior 1 Siguiente

Sistema Experto ©

## Registro Tipo de Enfermedades

**Tipo de enfermedad** Agregar Tipo de enfermedad

EXCEL PDF BUSCAR:

#	Nombre	Descripción	Acciones
1	PULMONAR	Enfermedad pulmonar se refiere a trastornos que afectan los pulmones, los órganos que nos permiten respirar. Las enfermedades pulmonares son una gran preocupación para las mujeres. La cantidad de mujeres en los EE. UU. diagnosticadas con una enfermedad pulmonar continúa en aumento. También, más mujeres están muriendo por enfermedades pulmonares.	

1 de 1 registros

Anterior 1 Siguiente

Sistema Experto

## Registro de Enfermedades

**Enfermedades** Agregar Enfermedad

EXCEL PDF BUSCAR:

#	Nombre	Tipo de Enfermedad	Puntaje	Acciones
1	Influenza	PULMONAR	14	
2	Neumonía	PULMONAR	18	
3	Asma	PULMONAR	10	
4	Fibrosis	PULMONAR	20	
5	Tuberculosis	PULMONAR	20	
6	Bronquitis	PULMONAR	12	
7	Gripe general	PULMONAR	10	

## Registro de Tratamiento por Usuario

The screenshot shows the 'Tratamientos' section of the application. On the left is a dark sidebar with a menu containing: Inicio, Pacientes, Médicos, Síntomas, Medicina, Tipo de enfermedad, Enfermedades, **Tratamiento por usuario** (highlighted), Tratamiento General, Diagnóstico, and Reportes. The main content area has a header with a user profile icon and 'Diagnósticos'. Below the header is a search bar labeled 'Ingrese DNI del paciente'. A table with the following columns is displayed: '#', 'Enfermedad', 'Fecha', and 'Acciones'. The table is currently empty. At the bottom, the text 'Sistema Experto ©' is visible.

## Registro de Tratamiento General

The screenshot shows the 'Tratamiento General' section. The sidebar is identical to the previous screenshot, with 'Tratamiento General' highlighted. The main content area has a header with a user profile icon and 'Diagnósticos'. Below the header is a 'Nuevo Tratamiento' button. A filter section includes a dropdown menu for 'Enfermedad' set to 'Todos'. Below this are two buttons: 'EXCEL' and 'PDF'. To the right is a search field labeled 'BUSCAR:'. A table with columns: '#', 'Enfermedad', 'Edad', 'Peso', and 'Acciones' is shown. The table contains the text 'Ningún dato disponible en esta tabla'. Below the table, it says 'No se encontraron registros' and has 'Anterior' and 'Siguiete' buttons. At the bottom, the text 'Sistema Experto ©' is visible.



## Registro de Diagnóstico

## Reportes

## Anexo 12: Entrevista al cliente

Investigador : Gianmarco Roger Nizama Antunez  
Entrevistado : Miguel Pérez  
Cargo del Entrevistado : Encargado de área  
Fecha de Entrevista : 20 Octubre 2021

### Preguntas:

1. Nombre de la empresa y ubicación.  
Policlínico Monserrat, Ubicada en Carabayllo
2. Sector empresarial de la organización  
Área de salud
3. Actividades principales del Policlínico  
El policlínico brinda el servicio de medicina general y pediátrica, psicología, odontología, laboratorio y ahora pruebas COVID
4. Explique el proceso en donde se tienen problemas

El policlínico inicia su proceso de atención al cliente o al paciente, por medio del registro del mismo y creación de la historia, luego de esto se genera una cita para que el doctor pueda atender y generar el diagnóstico de la enfermedad que tiene el paciente, para luego poder generar un tratamiento y las recetas respectivas. Con la situación actual que se viene dando por el coronavirus el policlínico ha empezado a tener problemas en la atención de los pacientes y en la generación de diagnósticos, ya que esta enfermedad está generando efectos secundarios, secuelas y tensión en los pacientes, lo que genera una gran cantidad de pacientes muchas veces desesperados por ser atendidos o por saber el diagnóstico que se tiene. Esto está generando cuellos de botella, y falta de satisfacción a los pacientes por el tiempo de

espera no sólo en la atención sino también en la generación de diagnósticos y resultados de los exámenes médicos, Y ahora con el tema del coronavirus, es un poco complicado mantener el distanciamiento de los pacientes que quedan en la espera de una atención, y a veces todos los problemas se juntan y dar una solución es complicada ya que a veces se debe poner prioridad a las emergencias y los pacientes tienen que esperar mucho más tiempo.

5. ¿Desde hace cuanto tiempo se presentan estos problemas?

Desde inicios de la pandemia los problemas han empezado a ser bastante fuertes.

6. ¿Tiene un estimado en dinero, de cuanto se pierde a diario por estos problemas?

No se puede estimar con exactitud el dinero perdido, pero si la gran falta de satisfacción por una gran parte de los pacientes.

7. ¿Qué medidas utiliza para evitar o solucionar estos problemas?

Actualmente lo que se trata de hacer es mantener un registro en los pacientes como historial y brindárselo a los doctores para que puedan generar mejores diagnósticos y la atención sea más fluida.

8. ¿Autoriza las visitas o entrevistas con las demás personas que intervienen en el proceso de producción en la empresa?

Sí

---

Firma y Sello

Miguel Perez

## **Anexo 13: Carta de presentación de la empresa**



### **El que suscribe en Policlínico Monerrat**

Consta que:

El Sr. Nizama Antunez, Gianmarco Roger con DNI 72409553, se encuentra desarrollando su tesis en nuestra empresa, titulada Sistema experto para diagnóstico de enfermedades pulmonares en el policlínico Monserrat, según los requerimientos especificados, para ayudar a que este proceso se optimice.

El sistema experto brindará el apoyo para la detección de enfermedades pulmonares, las cuales están en aumento por la pandemia mundial.

Se emite esta constancia para los fines que el alumno crea conveniente

Miguel Perez  
Encargado de Area

Lima, 25 de octubre de 2021

Av. Concondorcauqui 474 Urb. Tungsuca, Carabayllo

## Anexo 14: Turnitin

Feedback Studio - Google Chrome  
ev.turnitin.com/app/carta/es/?u=1107251081&lang=es&s=1&o=1846658765&student\_user=1

feedback studio GIANMARCO ROGER NIZAMA ANTUNEZ | Sistema experto para diagnóstico de enfermedades pulmonares en el policlinic...

**UNIVERSIDAD CÉSAR VALLEJO**  
FACULTAD DE INGENIERÍA Y ARQUITECTURA  
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

Sistema experto para diagnóstico de enfermedades pulmonares en el policlínico Monserrat

TESIS PARA OBTENER EL TÍTULO PROFESIONAL DE:  
LICENCIADO EN INGENIERÍA DE SISTEMAS

**AUTOR:**  
Nizama Antunez, Gianmarco Roger (0000-0001-9787-8060)

**ASESOR:**  
Mgtr. Liendo Arévalo Milner David, (0000-0002-7865-351X)

LÍNEA DE INVESTIGACIÓN:

**Resumen de coincidencias**

**22 %**

Se están viendo fuentes estándar

Ver fuentes en inglés (Beta)

Coincidencias

1	repositorio.ucv.edu.pe Fuente de Internet	11 %	>
2	Entregado a Universida... Trabajo del estudiante	4 %	>
3	hdl.handle.net Fuente de Internet	1 %	>
4	www.psycologia.com Fuente de Internet	1 %	>
5	repositorio.undao.edu... Fuente de Internet	1 %	>
6	Entregado a Colegio La...	<1 %	>

## **Anexo 15: Constancia de traducción de abstract**

This document has been translated by the Translation and Interpreting Service of Cesar Vallejo University and it has been revised by the native speaker of English: Mark Stables.



*Ana Gonzales Castañeda*

Dr. Ana Gonzales Castañeda  
Professor of the School of Languages

## Anexo 16: Recibo del pago traducción de abstract



UNIVERSIDAD CÉSAR VALLEJO S.A.C.  
R.U.C. 20164113532  
AV. LARCO 1770 - URB. SAN ANDRÉS STA ETAPA -  
VÍCTOR LARCO - TRUJILLO - TRUJILLO  
UCV CAMPUS TRUJILLO  
BOLETA DE VENTA ELECTRÓNICA  
BA16 - Nº 1002539

Código : 7001201206  
Nombres : NIZAMA ANTUNEZ GIANMARCO ROGER  
Unidad : UCV TRUJILLO  
Programa : EXTERNOS

Cant. x Prec. Unit.  
Descripción

1.00 x 50.00		
ABSTRACT PREGRADO TEMAS		50.00
DISTINTOS A EDUCACION-COSTO POR		
PAG		
<hr/>		
TOTAL		50.00
EXONERADA	S/	0.00
INALECTA	S/	0.00
GRAVADA	S/	42.37
I.G.V.	S/	7.63
TOTAL	S/	50.00

Emisión : 23/06/2022 Venc. : 23/06/2022  
Estado : CANCELADO T.C. : 0.00  
VENTA CONTADO

SEUSS SEUSS SEUSS ADMIN (SEUSS)

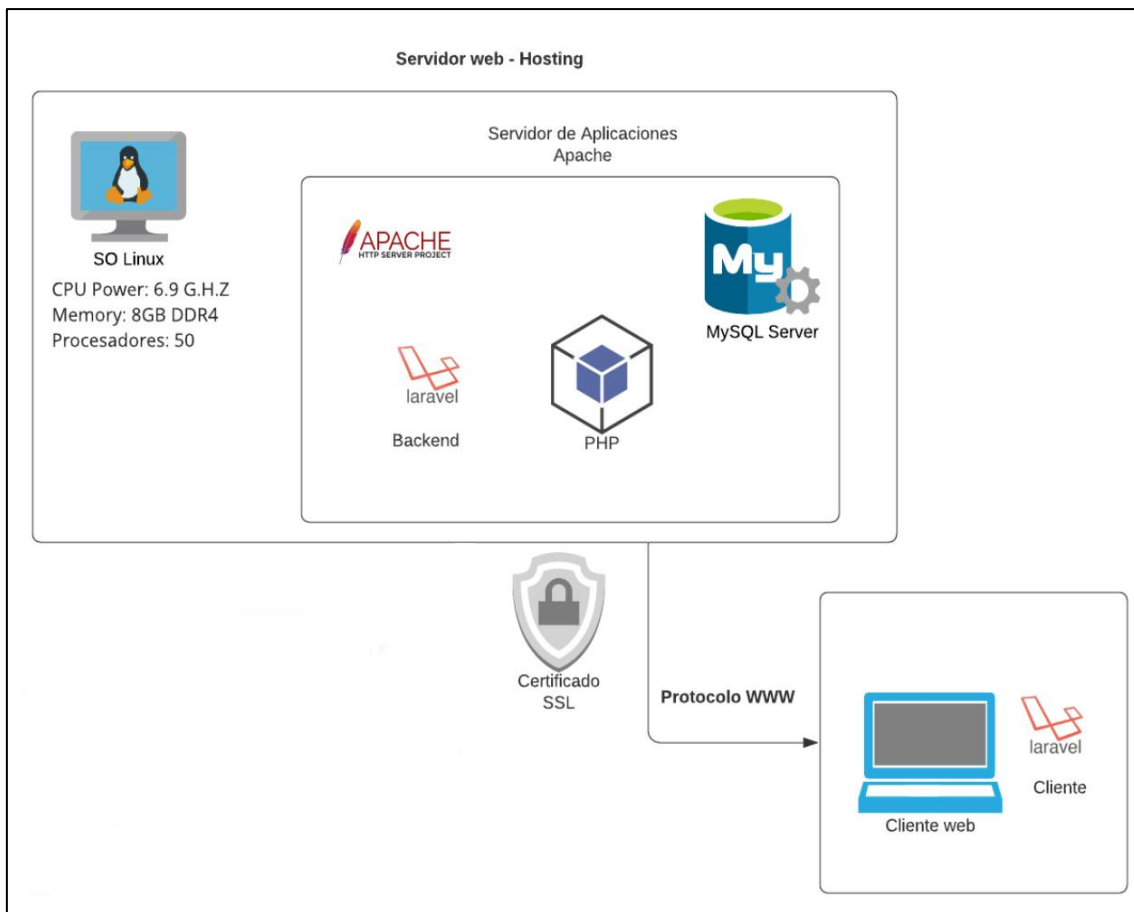
"Los derechos de crédito que mantiene la Universidad César Vallejo S.A.C., representados por el presente comprobante de pago, han sido transferidos en dominio fiduciario a la Fiduciaria S.A."

23/06/2022 10:29:48 p.m.

Representación impresa del comprobante de venta electrónico,



## Anexo 17: Arquitectura del software





## **Anexo 18: Desarrollo de metodología de software Scrum**

En este documento se logrará observar a detalle la implementación de la metodología de desarrollo de software SCRUM en la investigación titulada “Sistema experto para diagnóstico de enfermedades pulmonares en el policlínico Monserrat”.

Esta propuesta está basada referente al análisis de entregables funcionales, aquellos que son llamados como Sprints, de igual manera se asegura que estos entregables será 100% funcionales y darán pase a la implementación de módulos para el uso requerido esto sin tener el 100% del software completo.

### **Alcance:**

Según lo que sea analizado, a continuación, se detallan los objetivos del software:

- Desarrollar un sistema web que permita gestionar los diagnósticos de enfermedades de pacientes, recolectando sus síntomas y generando un diagnóstico confiable.
- Así mismo generando tratamientos requeridos para cada enfermedad.
- Desarrollar un sistema web la cual permita gestionar la lista de pacientes, médicos, medicinas y enfermedades
- Mejorar el tiempo de entrega de los diagnósticos de enfermedades pulmonares.
- Desarrollar una aplicación móvil que permita registrar los datos del paciente como también sus síntomas y generar tratamientos requeridos.

## Roles:

Así mismo en el cuadro siguiente se puede observar a detalle los roles para esta investigación.

**Tabla\_ 1: Nombre y Roles del Proyecto**

<b>ROL</b>	<b>NOMBRE</b>
Scrum Master	Luz Gomez Taype
Team Member	Gianmarco Roger Nizama Antunez
Product Owner	Miguel Rios

**FUENTE:** Elaboración Propia

## 1. Planificación: Historias de usuario

**Tabla\_ 2: Historia de Usuario 1**

<b>Historia de Usuario</b>	
<b>Número: 1</b>	<b>Usuario:</b> Todos
<b>Nombre Historia:</b> Inicio de sesión	<b>Tiempo Estimado:</b> 3 días
<b>Iteración: 1</b>	<b>Prioridad:</b> Alta
<b>Programador responsable:</b>	
<b>Descripción:</b> El sistema permitirá el inicio de sesión de dos tipos de usuarios, admin y super admin.	

Fuente: Elaboración Propia

**Tabla\_ 3: Historia de Usuario 2**

<b>Historia de Usuario</b>	
<b>Número: 2</b>	<b>Usuario:</b> Todos
<b>Nombre Historia:</b> Dashboard	<b>Tiempo Estimado:</b> 2 días
<b>Iteración: 1</b>	<b>Prioridad:</b> Baja
<b>Programador responsable:</b>	
<b>Descripción:</b> El sistema brindará un resumen de reportes a los usuarios.	

Fuente: Elaboración Propia

**Tabla\_ 4: Historia de Usuario 3**

<b>Historia de Usuario</b>	
<b>Número: 3</b>	<b>Usuario:</b> Administrador
<b>Nombre Historia:</b> Gestión de pacientes - Lista	<b>Tiempo Estimado:</b> 4 días
<b>Iteración: 1</b>	<b>Prioridad:</b> Alta
<b>Programador responsable:</b>	
<b>Descripción:</b> El sistema permitirá gestionar la lista de los pacientes.	

Fuente: Elaboración Propia

**Tabla\_ 5: Historia de Usuario 4**

<b>Historia de Usuario</b>	
<b>Número: 4</b>	<b>Usuario:</b> Administrador
<b>Nombre Historia:</b> Gestión de pacientes - Registro	<b>Tiempo Estimado:</b> 4 días
<b>Iteración: 2</b>	<b>Prioridad:</b> Alta
<b>Programador responsable:</b>	
<b>Descripción:</b> El sistema permitirá el registro de los datos del paciente, DNI, nombres, apellidos, teléfono, correo, fecha de nacimiento, talla y peso.	

Fuente: Elaboración Propia

**Tabla\_ 6: Historia de Usuario 5**

<b>Historia de Usuario</b>	
<b>Número: 5</b>	<b>Usuario:</b> Administrador
<b>Nombre Historia:</b> Gestión de médicos - Lista	<b>Tiempo Estimado:</b> 4 días
<b>Iteración: 2</b>	<b>Prioridad:</b> Alta
<b>Programador responsable:</b>	
<b>Descripción:</b> El sistema permitirá gestionar la lista de todos los médicos registrados.	

Fuente: Elaboración Propia

**Tabla\_ 7: Historia de Usuario 6**

<b>Historia de Usuario</b>	
<b>Número: 6</b>	<b>Usuario:</b> Administrador
<b>Nombre Historia:</b> Gestión de médicos - Registro	<b>Tiempo Estimado:</b> 4 días
<b>Iteración: 3</b>	<b>Prioridad:</b> Alta
<b>Programador responsable:</b>	
<b>Descripción:</b> El sistema permitirá registrar los datos de cada médico, DNI, nombres, apellidos, teléfono, correo, fecha de nacimiento y su especialidad.	

Fuente: Elaboración Propia

**Tabla\_ 8: Historia de Usuario 7**

<b>Historia de Usuario</b>	
<b>Número: 7</b>	<b>Usuario:</b> Administrador
<b>Nombre Historia:</b> Gestión Tipo de enfermedad	<b>Tiempo Estimado:</b> 5 días
<b>Iteración: 3</b>	<b>Prioridad:</b> Alta
<b>Programador responsable:</b>	
<b>Descripción:</b> El sistema permitirá gestionar el tipo de enfermedad, el nombre y la descripción.	

Fuente: Elaboración Propia

**Tabla\_ 9: Historia de Usuario 8**

<b>Historia de Usuario</b>	
<b>Número: 8</b>	<b>Usuario:</b> Usuario
<b>Nombre Historia:</b> Gestión de síntomas	<b>Tiempo Estimado:</b> 5 días
<b>Iteración:</b> 4	<b>Prioridad:</b> Alta
<b>Programador responsable:</b>	
<b>Descripción:</b> El sistema web permitirá gestionar los síntomas donde se registrarán específicamente tres tipos de síntomas “opciones”, si/no y valor numérico.	

Fuente: Elaboración Propia

**Tabla\_ 10: Historia de Usuario 9**

<b>Historia de Usuario</b>	
<b>Número: 9</b>	<b>Usuario:</b> Todos
<b>Nombre Historia:</b> Registro de enfermedades	<b>Tiempo Estimado:</b> 5 días
<b>Iteración:</b> 4	<b>Prioridad:</b> Alta
<b>Programador responsable:</b>	
<b>Descripción:</b> El sistema web permitirá registrar enfermedades con sus respectivos síntomas en la cual cada síntoma tendrá un puntaje y en base a ello se medirá que enfermedades puede tener el paciente.	

Fuente: Elaboración Propia

**Tabla\_ 11: Historia de Usuario 10**

<b>Historia de Usuario</b>	
<b>Número: 10</b>	<b>Usuario:</b> Todos
<b>Nombre Historia:</b> Asignación de síntomas para las enfermedades	<b>Tiempo Estimado:</b> 6 días
<b>Iteración: 5</b>	<b>Prioridad:</b> Muy Alta
<b>Programador responsable:</b>	
<b>Descripción:</b> El sistema web permitirá asignar los síntomas para las enfermedades esto según a los tipos que se especificaron, "opciones", si/no y valor numérico.	

Fuente: Elaboración Propia

**Tabla\_ 12: Historia de Usuario 11**

<b>Historia de Usuario</b>	
<b>Número: 11</b>	<b>Usuario:</b> Todos
<b>Nombre Historia:</b> Registro de diagnósticos	<b>Tiempo Estimado:</b> 4 días
<b>Iteración: 5</b>	<b>Prioridad:</b> Muy Alta
<b>Programador responsable:</b>	
<b>Descripción:</b> El sistema web permite registrar diagnósticos de la cual se puede tener como resultados varias enfermedades. Así mismo se deberá evaluar cada enfermedad en base a síntomas.	

Fuente: Elaboración Propia

**Tabla\_ 13: Historia de Usuario 12**

<b>Historia de Usuario</b>	
<b>Número: 12</b>	<b>Usuario:</b> Todos
<b>Nombre Historia:</b> Resultados del procesamiento	<b>Tiempo Estimado:</b> 5 días
<b>Iteración:</b> 5	<b>Prioridad:</b> Muy Alta
<b>Programador responsable:</b>	
<b>Descripción:</b> El sistema web permitirá mostrar la probabilidad por cada enfermedad a la cual se podrá asignar un tratamiento por cada enfermedad detectada.	

Fuente: Elaboración Propia

**Tabla\_ 14: Historia de Usuario 13**

<b>Historia de Usuario</b>	
<b>Número: 13</b>	<b>Usuario:</b> Todos
<b>Nombre Historia:</b> Gestión de tratamientos	<b>Tiempo Estimado:</b> 5 días
<b>Iteración:</b> 5	<b>Prioridad:</b> Muy Alta
<b>Programador responsable:</b>	
<b>Descripción:</b> El sistema permitirá mostrar los tratamientos del paciente, las cuales podrán ser varias, sin embargo, en distintas fechas. Estos tratamientos deberán ser como recetas respecto al peso y talla de los pacientes.	

Fuente: Elaboración Propia



**Tabla\_ 15: Historia de Usuario 14**

<b>Historia de Usuario</b>	
<b>Número: 14</b>	<b>Usuario:</b> Todos
<b>Nombre Historia:</b> Gestión de medicinas	<b>Tiempo Estimado:</b> 4 días
<b>Iteración: 5</b>	<b>Prioridad:</b> Alta
<b>Programador responsable:</b>	
<b>Descripción:</b> El sistema permitirá gestionar la lista de medicinas.	

Fuente: Elaboración Propia

**Tabla\_ 16: Historia de Usuario 15**

<b>Historia de Usuario</b>	
<b>Número: 15</b>	<b>Usuario:</b> Todos
<b>Nombre Historia:</b> Registro de tratamientos	<b>Tiempo Estimado:</b> 4 días
<b>Iteración: 5</b>	<b>Prioridad:</b> Alta
<b>Programador responsable:</b>	
<b>Descripción:</b> El sistema permitirá registrar los tratamientos para cada enfermedad de igual manera se cargan las medicinas por defecto.	

Fuente: Elaboración Propia

**Tabla\_ 17: Historia de Usuario 16**

<b>Historia de Usuario</b>	
<b>Número: 16</b>	<b>Usuario:</b> Todos
<b>Nombre Historia:</b> Registros de tratamientos generales	<b>Tiempo Estimado:</b> 4 días
<b>Iteración:</b> 5	<b>Prioridad:</b> Alta
<b>Programador responsable:</b>	
<b>Descripción:</b> El sistema permitirá registrar tratamientos generales, respecto a la edad y peso de las personas, para de esta manera se le asigne automáticamente.	

Fuente: Elaboración Propia

**Tabla\_ 18: Historia de Usuario 17**

<b>Historia de Usuario</b>	
<b>Número: 17</b>	<b>Usuario:</b> Todos
<b>Nombre Historia:</b> Gestión de síntomas paciente	<b>Tiempo Estimado:</b> 4 días
<b>Iteración:</b> 4	<b>Prioridad:</b> Alta
<b>Programador responsable:</b>	
<b>Descripción:</b> El sistema web permitirá gestionar y mostrar los síntomas de cada paciente.	

Fuente: Elaboración Propia

**Tabla\_ 19: Historia de Usuario 18**

<b>Historia de Usuario</b>	
<b>Número: 18</b>	<b>Usuario:</b> Todos
<b>Nombre Historia:</b> Gestión de historia clínica	<b>Tiempo Estimado:</b> 4 días
<b>Iteración:</b> 4	<b>Prioridad:</b> Alta
<b>Programador responsable:</b>	
<b>Descripción:</b> El sistema web permitirá gestionar y mostrar la historia clínica de cada paciente.	

Fuente: Elaboración Propia

**Tabla\_ 20: Product Backlog**

ITEM	H.U	Nombre de Historia	Iteración	Tiempo Estimado	Prioridad
1	HU1	Inicio de sesión	1	3	ALTA
2	HU2	Dashboard	1	2	BAJA
3	HU3	Gestión de pacientes – Lista	1	4	ALTA
4	HU4	Gestión de pacientes – Registro	1	4	ALTA
5	HU5	Gestión de médicos – Lista	2	4	ALTA
6	HU6	Gestión de médicos – Registro	2	4	ALTA
7	HU7	Gestión Tipo de enfermedad	2	5	ALTA
8	HU8	Gestión de síntomas	2	5	ALTA
9	HU9	Registro de enfermedades	3	5	ALTA
10	HU10	Asignación de síntomas para las enfermedades	3	6	MUY ALTA
11	HU11	Registro de diagnósticos	3	4	MUY ALTA
12	HU12	Resultados del procesamiento	3	5	MUY ALTA
13	HU13	Gestión de tratamientos	4	5	MUY ALTA
14	HU14	Gestión de medicinas	4	4	ALTA
15	HU15	Registro de tratamientos	4	4	ALTA
16	HU16	Registros de tratamientos generales	4	4	ALTA
17	HU17	Gestión de síntomas pacientes	4	4	ALTA
18	HU18	Gestión de historia clínica	4	4	ALTA

**Fuente:** Elaboración Propia

**Tabla\_ 21: Requerimientos Funcionales**

<b>Código</b>	<b>Requerimiento Funcional</b>	<b>Prioridad</b>	<b>Estimación en días</b>	<b>Tiempo Real</b>
<b>RF1</b>	El sistema permitirá el inicio de sesión de dos tipos de usuarios, admin y super admin.	1	3	3
<b>RF2</b>	El sistema brindará un resumen de reportes a los usuarios.	1	2	2
<b>RF3</b>	El sistema permite gestionar la lista de los pacientes.	1	4	4
<b>RF4</b>	El sistema permitirá el registro de los datos del paciente, dni, nombres, apellidos, teléfono, correo, fecha de nacimiento, talla y peso.	1	4	4
<b>RF5</b>	El sistema permitirá gestionar la lista de todos los médicos registrados.	2	4	4
<b>RF6</b>	El sistema permitirá registrar los datos de cada médico, DNI, nombres, apellidos, teléfono, correo, fecha de nacimiento y su especialidad.	2	4	4
<b>RF7</b>	El sistema permitirá gestionar el tipo de enfermedad, el nombre y la descripción.	2	5	5
<b>RF8</b>	El sistema web permitirá gestionar los síntomas donde se registrarán específicamente tres tipos de síntomas "opciones", si/no y valor numérico.	2	5	5
<b>RF9</b>	El sistema web permitirá registrar enfermedades con sus respectivos síntomas en la cual cada síntoma tendrá un puntaje y en base a ello se medirá que enfermedades puede tener el paciente.	3	5	5
<b>RF10</b>	El sistema web permitirá asignar los síntomas para las enfermedades esto según a los tipos que se especificaron, "opciones", si/no y valor numérico.	3	6	6
<b>RF11</b>	El sistema web permite registrar diagnósticos de la cual se puede tener como resultados varias enfermedades. Así mismo se deberá evaluar cada enfermedad en base a síntomas.	3	4	4
<b>RF12</b>	El sistema web permitirá mostrar la probabilidad por cada enfermedad a la cual se podrá asignar un tratamiento por cada	3	5	5

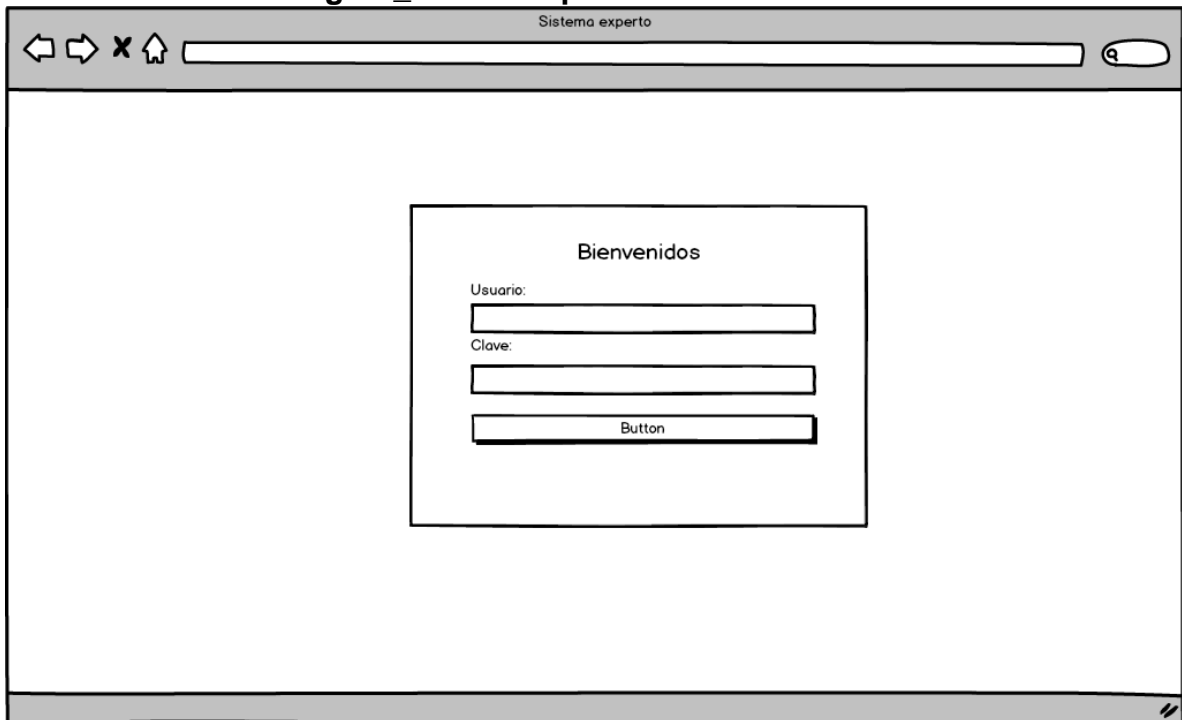
	enfermedad detectada.			
<b>RF13</b>	El sistema permitirá mostrar los tratamientos del paciente, las cuales podrán ser varias, sin embargo, en distintas fechas. Estos tratamientos deberán ser como recetas respecto al peso y talla de los pacientes.	4	5	5
<b>RF14</b>	El sistema permitirá gestionar la lista de medicinas.	4	4	4
<b>RF15</b>	El sistema permitirá registrar los tratamientos para cada enfermedad de igual manera se cargan las medicinas por defecto.	4	4	4
<b>RF16</b>	El sistema permitirá registrar tratamientos generales, respecto a la edad y peso de las personas, para de esta manera se le asigne automáticamente.	4	4	4
<b>RF17</b>	El sistema web permitirá gestionar y mostrar los síntomas de cada paciente.	4	4	4
<b>RF18</b>	El sistema web permitirá gestionar y mostrar la historia clínica de cada paciente.	4	4	4
<b>RF19</b>	El sistema móvil permitirá el inicio de sesión de dos tipos de usuarios, admin y super admin.	5	2	2
<b>RF20</b>	El sistema móvil permitirá gestionar los diagnósticos de cada paciente.	5	5	5
<b>RF21</b>	El sistema móvil permitirá gestionar los síntomas de cada paciente.	5	6	6
<b>RF22</b>	El sistema móvil mostrará el módulo perfil a todos los usuarios, para que se pueda gestionar los respectivos datos.	5	3	3

**Tabla\_ 22: Requerimientos No Funcionales**

<b>Código</b>	<b>Tipo</b>	<b>Requerimiento No Funcional</b>
<b>RNF1</b>	<b>Usabilidad</b>	El sistema debe ser intuitivo para que de esta manera los usuarios aprendan a manejar el uso del sistema de forma rápida y sencilla.
		El sistema debe estar desarrollado con un diseño orientado a la línea gráfica de la empresa.
<b>RNF2</b>	<b>Fiabilidad</b>	El sistema debe proporcionar confianza y seguridad de que toda la información que sea manejado, será analizada por la empresa.
		El sistema debe ser competente al soportar ataques externos.
<b>RNF3</b>	<b>Rendimiento</b>	El sistema debe tener la capacidad de rapidez y soporte, para gestionar la gran cantidad de usuarios que además van conectados entre sí.
<b>RNF4</b>	<b>Disponibilidad</b>	El sistema debe estar disponible las 24 horas para evitar problema en la conectividad de los usuarios desde distintos lugares en todo momento.
<b>RNF5</b>	<b>Soporte</b>	El sistema debe tener la sencillas de análisis y comprensión de código para la gestión de soporte de manera sencillo y rápido.
<b>RNF6</b>	<b>Seguridad</b>	El sistema debe brindar un nivel de seguridad lo suficientemente eficaz para diferenciar funcionalidades de cada perfil de los usuarios, y evitar robo de información de distintos tipos.

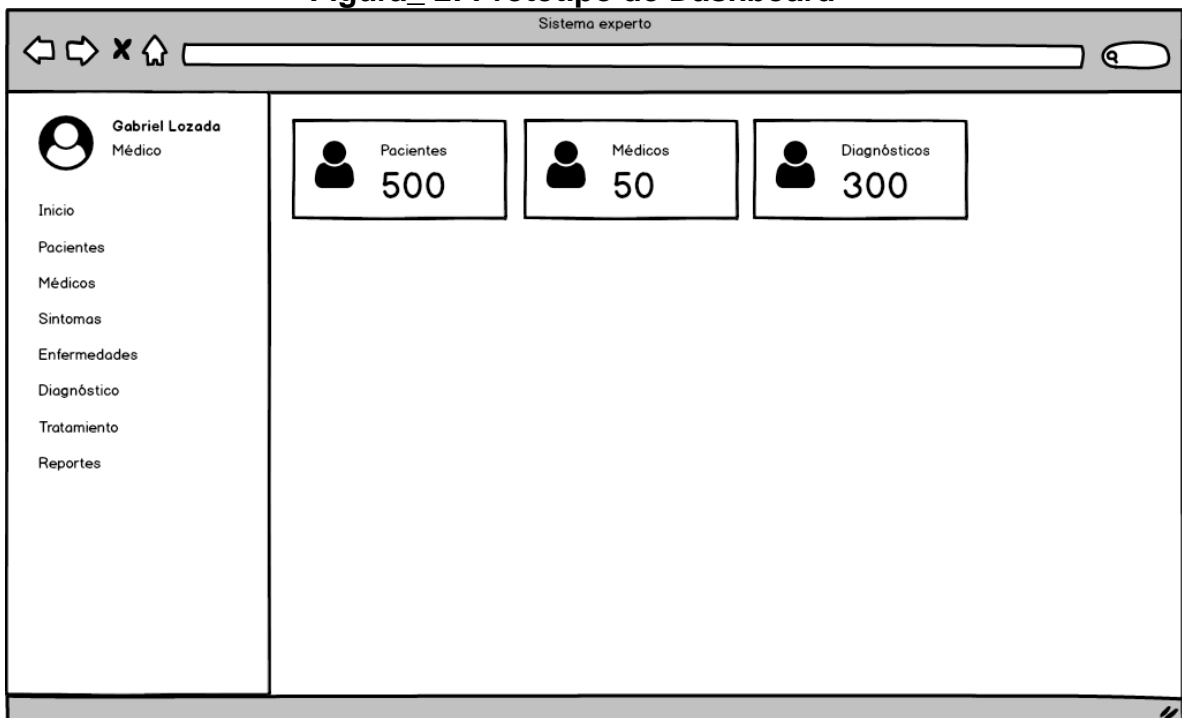
## Mockups o interfáz de usuario

Figura\_ 1: Prototipo Inicio de sesión



Prototipo de inicio de sesión para un sistema experto. La interfaz muestra un navegador con el título "Sistema experto". El contenido principal es un formulario de login centrado con el título "Bienvenidos". El formulario incluye campos para "Usuario:" y "Clave:", y un botón "Button".

Figura\_ 2: Prototipo de Dashboard



Prototipo de dashboard para un sistema experto. La interfaz muestra un navegador con el título "Sistema experto". El dashboard incluye un perfil de usuario "Gabriel Lozada Médico" con un menú de navegación lateral que contiene: Inicio, Pacientes, Médicos, Síntomas, Enfermedades, Diagnóstico, Tratamiento y Reportes. El contenido principal muestra tres tarjetas de resumen: "Pacientes" con 500, "Médicos" con 50, y "Diagnósticos" con 300.



**Figura\_ 3: Prototipo de Gestión de pacientes - Lista**

Sistema experto

Gabriel Lozada  
Médico

Inicio  
Pacientes  
Médicos  
Síntomas  
Enfermedades  
Diagnóstico  
Tratamiento

Pacientes

Nuevo Registro

Item	Nombres	Telefono	Edad	Acciones
1	Luis Lopez	98765322	30 años	
2	Rosa Ruiz	98733322	31 años	

**Figura\_ 4: Prototipo de Gestión de pacientes - Registro**

Sistema experto

Gabriel Lozada  
Médico

Inicio  
Pacientes  
Médicos  
Síntomas  
Enfermedades  
Diagnóstico  
Tratamiento

Pacientes

DNI:

Nombres:

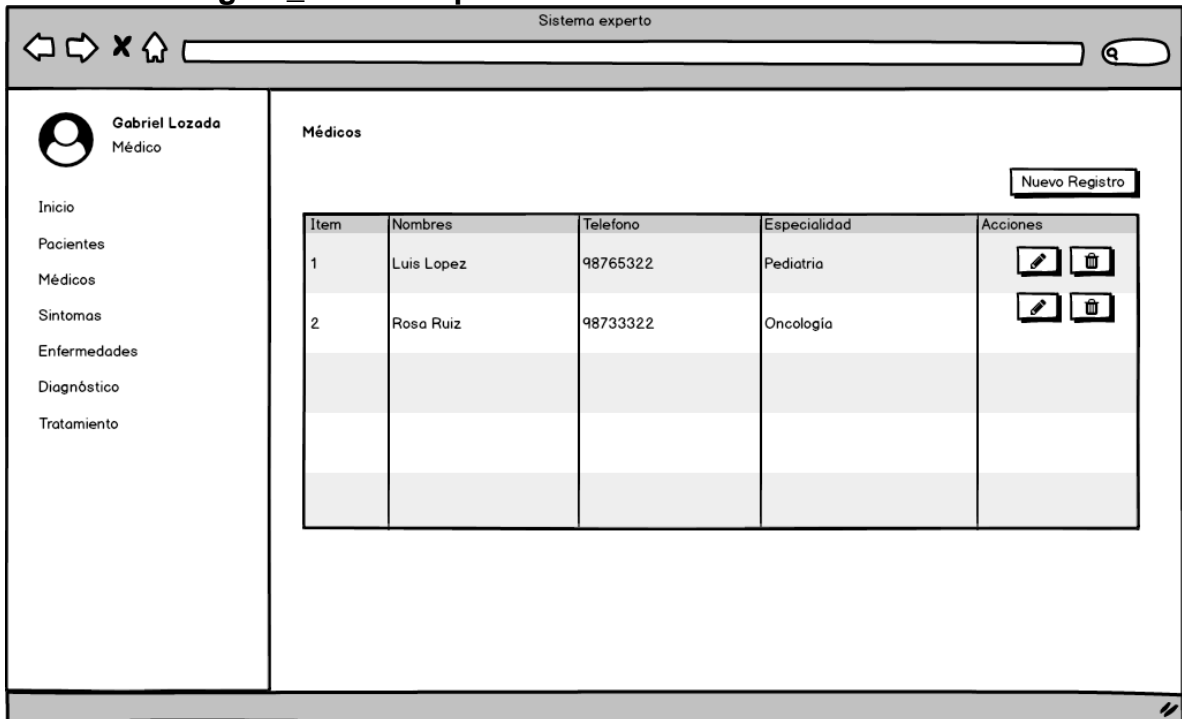
Apellidos:

Teléfono:

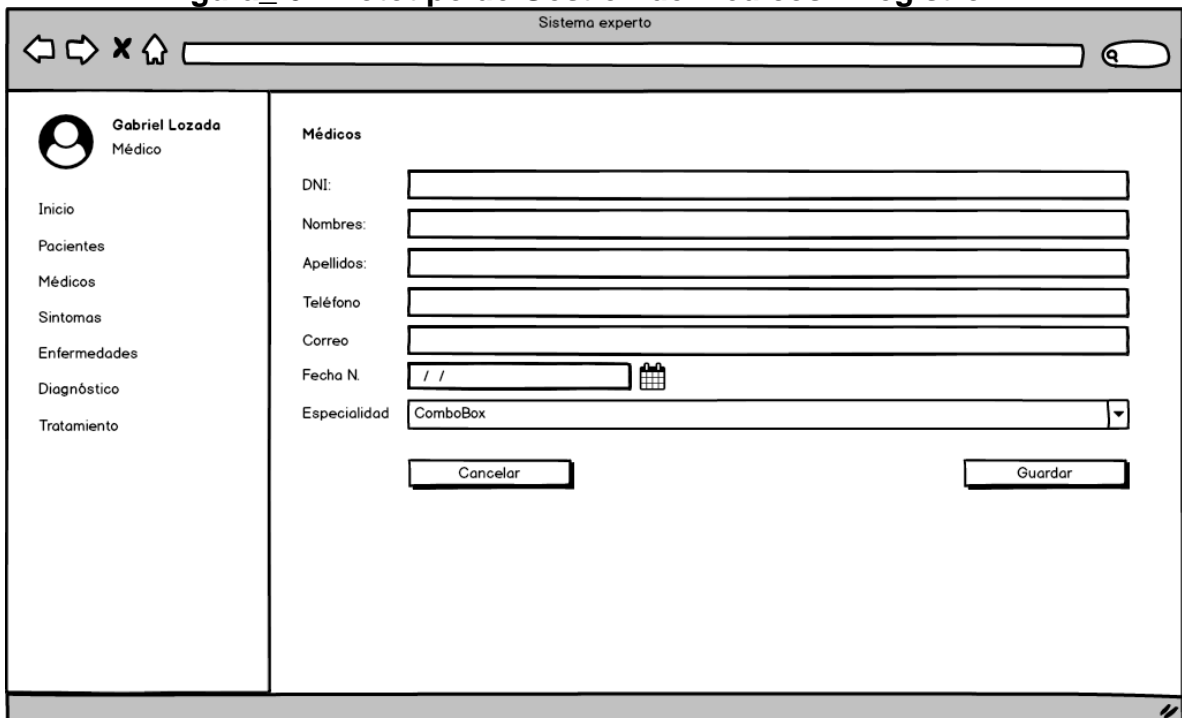
Correo:

Fecha N.  Talla:  Peso:

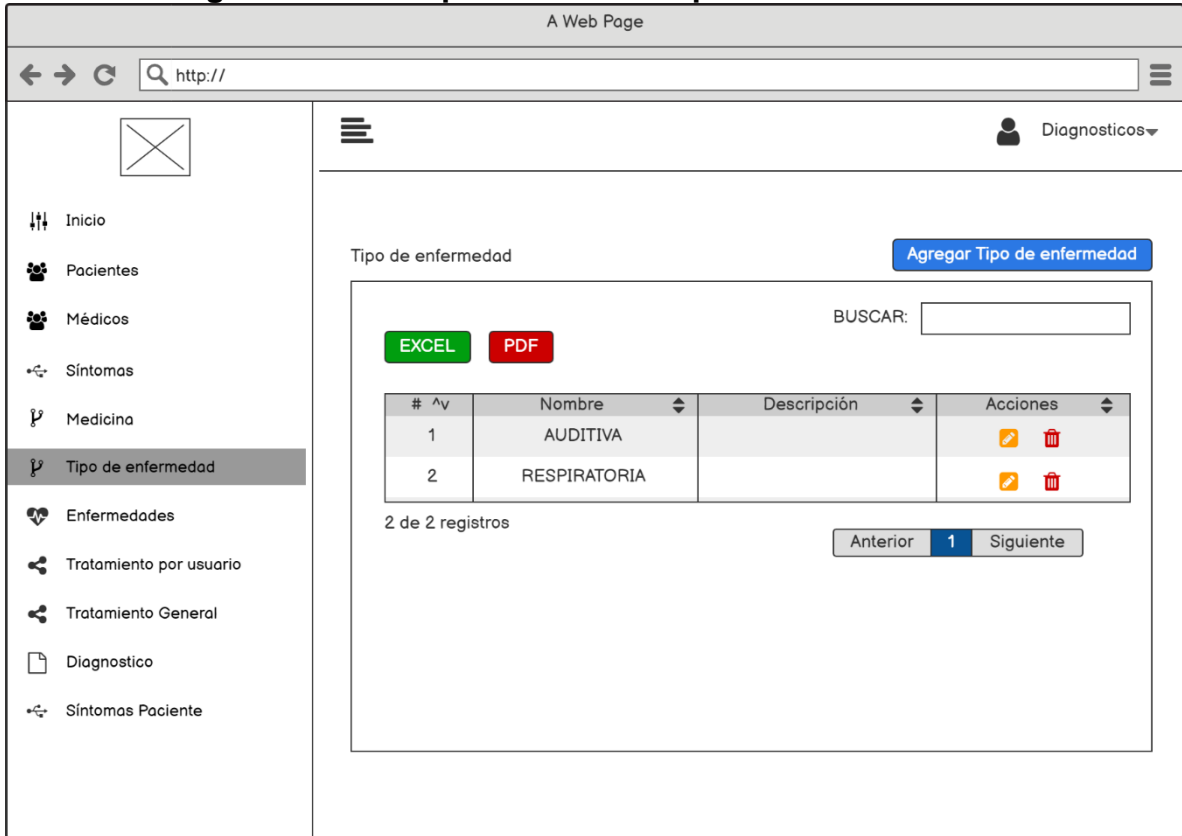
Figura\_ 5: Prototipo de Gestión de médicos - Lista



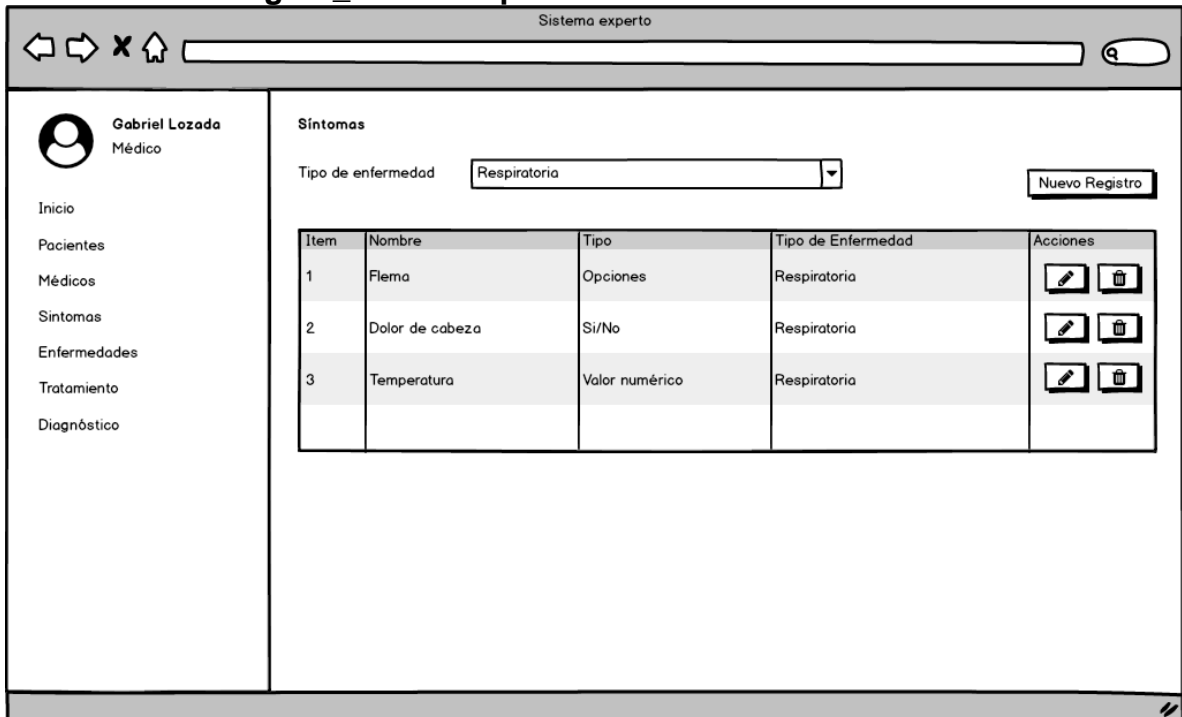
Figura\_ 6: Prototipo de Gestión de médicos - Registro



**Figura\_ 7: Prototipo de Gestión Tipo de enfermedad**



**Figura\_ 8: Prototipo de Gestión de síntomas**



## Síntomas Opciones

Sistema experto

Gabriel Lozada  
Médico

Inicio  
Pacientes  
Médicos  
Síntomas  
Enfermedades  
Tratamiento  
Diagnóstico

**Registro de Síntomas**

Tipo enfermedad: Respiratoria

Tipo: Opciones

Nombre: Flema

Items:  Amarrilla  Transparente  Verde

ingrese separado por comas

Cancelar Guardar

## Síntomas - Si/no

Sistema experto

Gabriel Lozada  
Médico

Inicio  
Pacientes  
Médicos  
Síntomas  
Enfermedades  
Tratamiento  
Diagnóstico

**Registro de Síntomas**

Tipo enfermedad: Respiratoria

Tipo: Si/NO

Nombre: Dolor de Cabeza

Cancelar Guardar

## Síntomas - Respiratoria

Sistema experto

Gabriel Lozada  
Médico

Inicio  
Pacientes  
Médicos  
Síntomas  
Enfermedades  
Tratamiento  
Diagnóstico

Registro de Síntomas

Tipo enfermedad: Respiratoria

Tipo: Valor numérico

Nombre: Temperatura

Cancelar Guardar

Figura\_ 9: Prototipo de Registro de enfermedades







Sistema experto

Gabriel Lozada  
Médico

Inicio  
Pacientes  
Médicos  
Síntomas  
Enfermedades  
Tratamiento  
Diagnóstico

Registro de Enfermedades

Nuevo Registro

Item	Nombre	Acciones
1	Covid 19	Síntomas  
2	Hepatitis	Síntomas  
3	Gripe	Síntomas  

Sistema experto

← → × 🏠

🔍

**Gabriel Lozada**  
Médico

Inicio

Pacientes

Médicos

Sintomas

Enfermedades

Tratamiento

Diagnóstico

### Registro de Enfermedades

Tipo enfermedad:

Nombre:

Descripción:

Puntaje de confirmación de enfermedad:  Si la evaluación es mayor o igual a esta, el paciente tendrá una probabilidad del 100% de tener esta enfermedad

**Figura\_ 10: Prototipo de Asignación de síntomas para las enfermedades**

Sistema experto

← → × 🏠

🔍

**Gabriel Lozada**  
Médico

Inicio

Pacientes

Médicos

Sintomas

Enfermedades

Tratamiento

Diagnóstico

### Sintomas para: Covid 19

Item	Nombre	Tipo	Accion
1	Flema	Opciones	<input type="button" value="✎"/> <input type="button" value="🗑"/>
2	Dolor de cabeza	si/no	<input type="button" value="✎"/> <input type="button" value="🗑"/>
3	Temperatura	valor	<input type="button" value="✎"/> <input type="button" value="🗑"/>

**Asignación de puntos por opciones**

Sistema experto

← → × 🏠

🔍

**Gabriel Lozada**  
Médico

Inicio

Pacientes

Médicos

Sintomas

Enfermedades

Tratamiento

Diagnóstico

### Sintomas

Agregar Sintomas:

Sistema experto

← → × ↶

🔍

**Gabriel Lozada**  
Médico

Inicio

Pacientes

Médicos

Sintomas

Enfermedades

Tratamiento

Diagnóstico

Sintomas

Agregar Sintomas:

Item	Nombre	Puntaje
1	Amarilla	<input style="width: 50px;" type="text" value="10"/>
2	Transparente	<input style="width: 50px;" type="text" value="4"/>
3	Verde	<input style="width: 50px;" type="text" value="0"/>

### Asignación de puntos si/no

Sistema experto

← → × ↶

🔍

**Gabriel Lozada**  
Médico

Inicio

Pacientes

Médicos

Sintomas

Enfermedades

Tratamiento

Diagnóstico

Sintomas

Agregar Sintomas:

Item	Nombre	Puntaje
1	Si	<input style="width: 50px;" type="text" value="10"/>
2	No	<input style="width: 50px;" type="text" value="1"/>

### Asignación de puntos por valores

Sistema experto

← → × ↶

🔍

**Gabriel Lozada**  
Médico

Inicio

Pacientes

Médicos

Sintomas

Enfermedades

Tratamiento

Diagnóstico

Sintomas

Agregar Sintomas:

Rango  y  puntaje

Item	Nombre	Puntaje	Acciones
1	entre 37 a 39	<input style="width: 50px;" type="text" value="8"/>	<input style="width: 30px;" type="button" value="🗑"/>
2	entre 39 a 40	<input style="width: 50px;" type="text" value="10"/>	<input style="width: 30px;" type="button" value="🗑"/>
3	entre 40 a 45	<input style="width: 50px;" type="text" value="15"/>	<input style="width: 30px;" type="button" value="🗑"/>

**Figura\_ 11: Prototipo de Registro de diagnósticos**

Sistema experto

← → × 🏠

🔍

**Gabriel Lozada**  
Médico

Inicio

Pacientes

Médicos

Sintomas

Enfermedades

Tratamiento

Diagnóstico

**Diagnóstico**

🔍 Ingrese DNI del paciente Buscar

Paciente: Juan Lopez Perez Nuevo Diagnóstico

Item	Fecha	Doctor	Acciones
1	01 Nov de 2021	Perez Albela	<span style="border: 1px solid #ccc; padding: 2px;">+ Tratamiento</span>
2	20 Nov de 2021	Dr. House	<span style="border: 1px solid #ccc; padding: 2px;">+ Tratamiento</span>

Sistema experto

← → × 🏠

🔍

**Gabriel Lozada**  
Médico

Inicio

Pacientes

Médicos

Sintomas

Enfermedades

Tratamiento

Diagnóstico

**Paciente: Juan Lopez Perez**

Tipo enfermedad: Respiratoria

Selecciones los sintomas del paciente, si no aplica deje en blanco

1. Flema: Amarilla

2. Dolor de cabeza:  Si  No

3. Temeperatura: 39

Procesar

**Figura\_ 12: Prototipo de Resultados del procesamiento**

Sistema experto

← → × 🏠

🔍

**Gabriel Lozada**  
Médico

Inicio

Pacientes

Médicos

Sintomas

Enfermedades

Tratamiento

Diagnóstico

**Resultados de análisis**

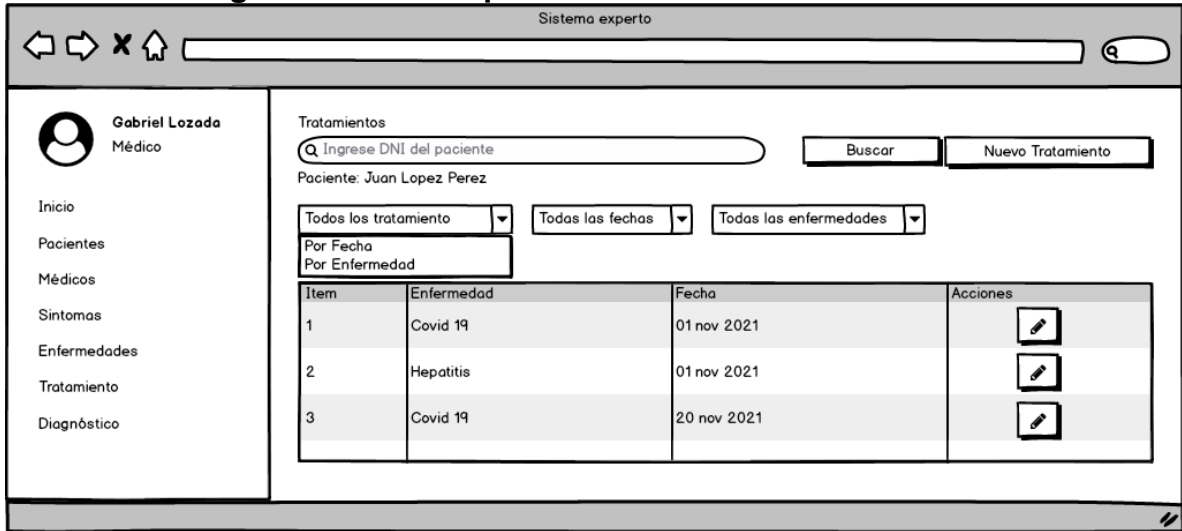
Paciente: Juan Lopez Perez

Acontinuación se detallan las posibles enfermedades que posee el paciente

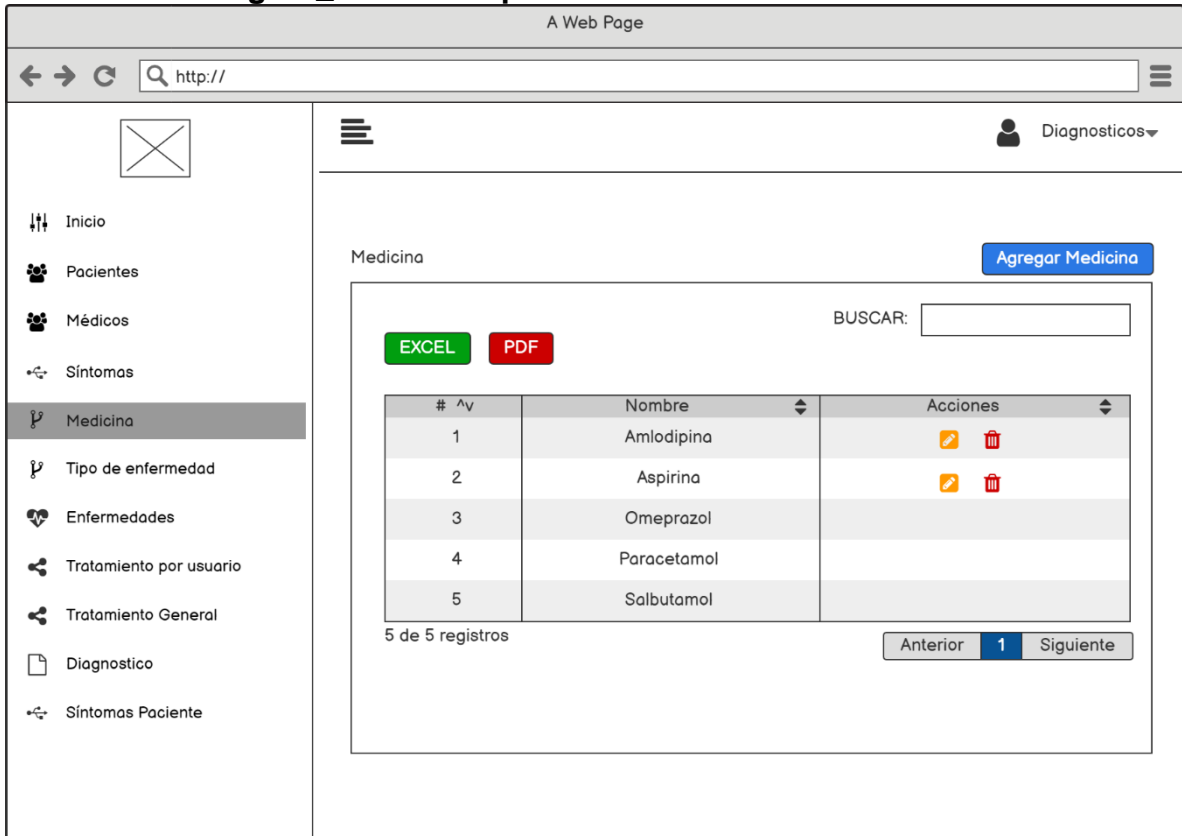
Item	Enfermedad	Probabilidad	Tratamiento
1	Covid	99%	<span style="border: 1px solid #ccc; padding: 2px;">+ Agregar</span>
2	Hepatitis	12%	<span style="border: 1px solid #ccc; padding: 2px;">+ Agregar</span>
3	Resfriado	80%	<span style="border: 1px solid #ccc; padding: 2px;">+ Agregar</span>
4	Lumbalgia	4%	<span style="border: 1px solid #ccc; padding: 2px;">+ Agregar</span>



**Figura\_ 13: Prototipo de Gestión de tratamientos**



**Figura\_ 14: Prototipo de Gestión de medicinas**



Figura\_ 15: Prototipo de Registro de tratamientos

Sistema experto

← → × ↶

🔍

**Gabriel Lozada**  
Médico

Inicio

Pacientes

Médicos

Sintomas

Enfermedades

Tratamiento

Diagnóstico

[Lista de tratamiento](#) > Registro de tratamiento

Paciente: Juan Lopez Perez

Enfermedad:  Fecha:

AGREGAR MEDICINA

Medicina:

Frecuencia:

Cantidad Total:

Detalles:

item	Medicina	Frecuencia	Cantidad	Acciones
1	Paracetamol 500gb	1 tableta cada 5 horas	10	<input type="button" value="✎"/> <input type="button" value="🗑"/>
2	Desametaxona 500m	1 cucharada cada 8 horas	10	<input type="button" value="✎"/> <input type="button" value="🗑"/>
3	Amoxicilina	1 Inyectada cada 3 días	5	<input type="button" value="✎"/> <input type="button" value="🗑"/>

Recomendaciones:

Figura\_ 16: Prototipo de Registros de tratamientos generales

Sistema experto

← → × ↶

🔍

**Gabriel Lozada**  
Médico

Inicio

Pacientes

Médicos

Sintomas

Enfermedades

Tratamiento

Diagnóstico

Tratamiento General

Enfermedad:

item	Enfermedad	Edad	Peso	Acciones
1	Covid 19	18 a 40 años	50 a 60 kilos	<input type="button" value="✎"/> <input type="button" value="🗑"/>
2	Covid 19	40 a 45 años	50 a 60 kilos	<input type="button" value="✎"/> <input type="button" value="🗑"/>
				<input type="button" value="✎"/> <input type="button" value="🗑"/>

Sistema experto

← → ✕ 🏠

🔍

**Gabriel Lozada**  
Médico

Inicio

Pacientes

Médicos

Sintomas

Enfermedades

Tratamiento

Diagnóstico

### Tratamiento General

Enfermedad:

Rango de edad:

Rango de peso:

**AGREGAR MEDICINA**

Medicina:

Frecuencia:

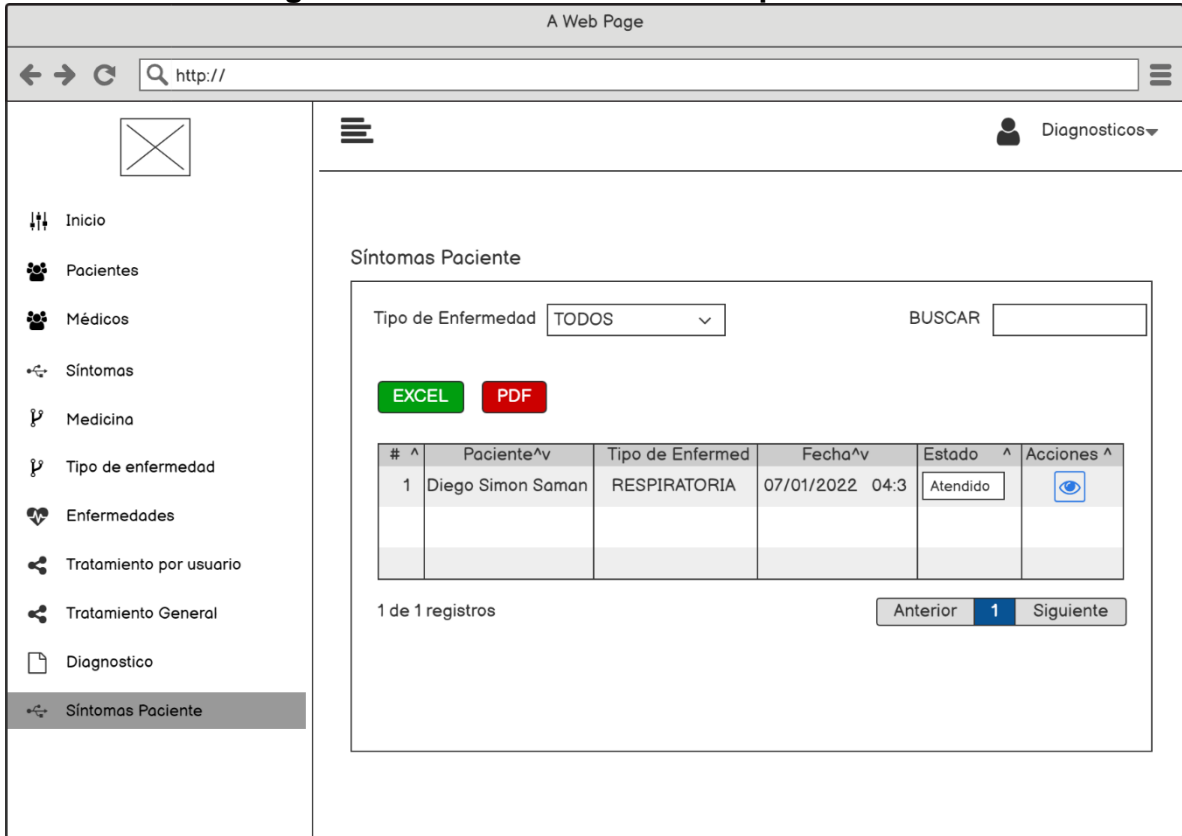
Cantidad Total:

Detalles:

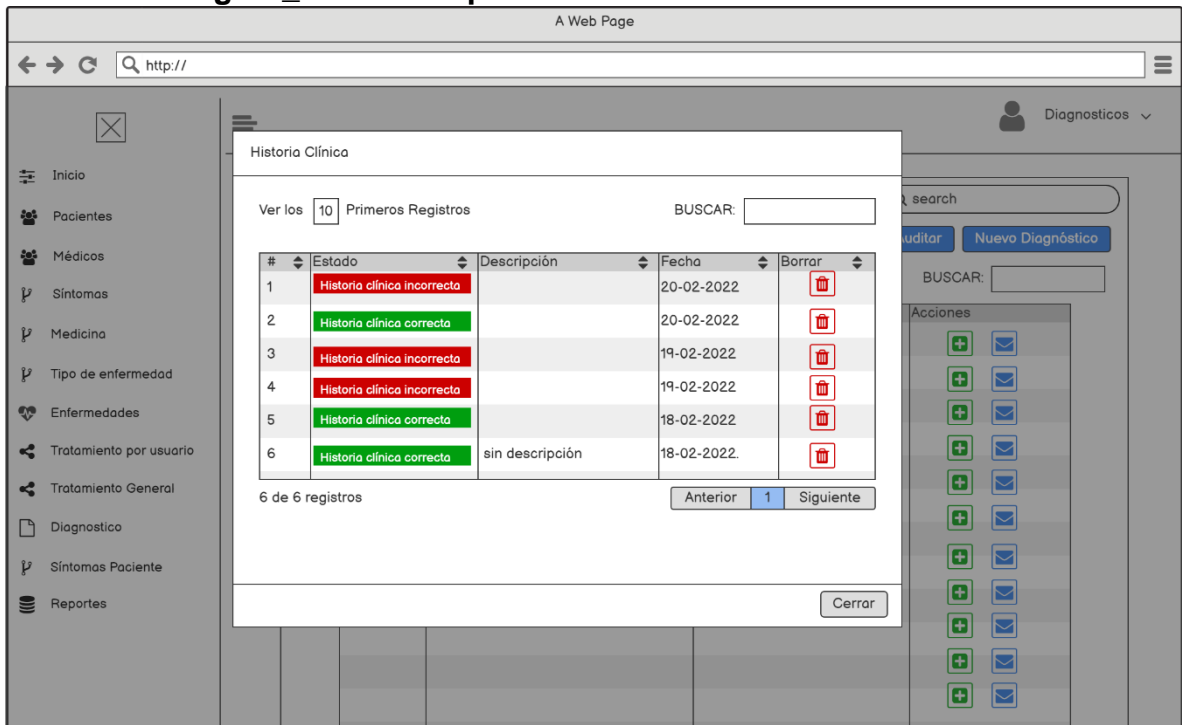
item	Medicina	Frecuencia	Cantidad	Acciones
1	Paracetamol 500gb	1 tableta cada 5 horas	10	<input type="button" value="✎"/> <input type="button" value="🗑"/>
2	Desametaxona 500m	1 cucharada cada 8 horas	10	<input type="button" value="✎"/> <input type="button" value="🗑"/>
3	Amoxicilina	1 Inyectada cada 3 días	5	<input type="button" value="✎"/> <input type="button" value="🗑"/>

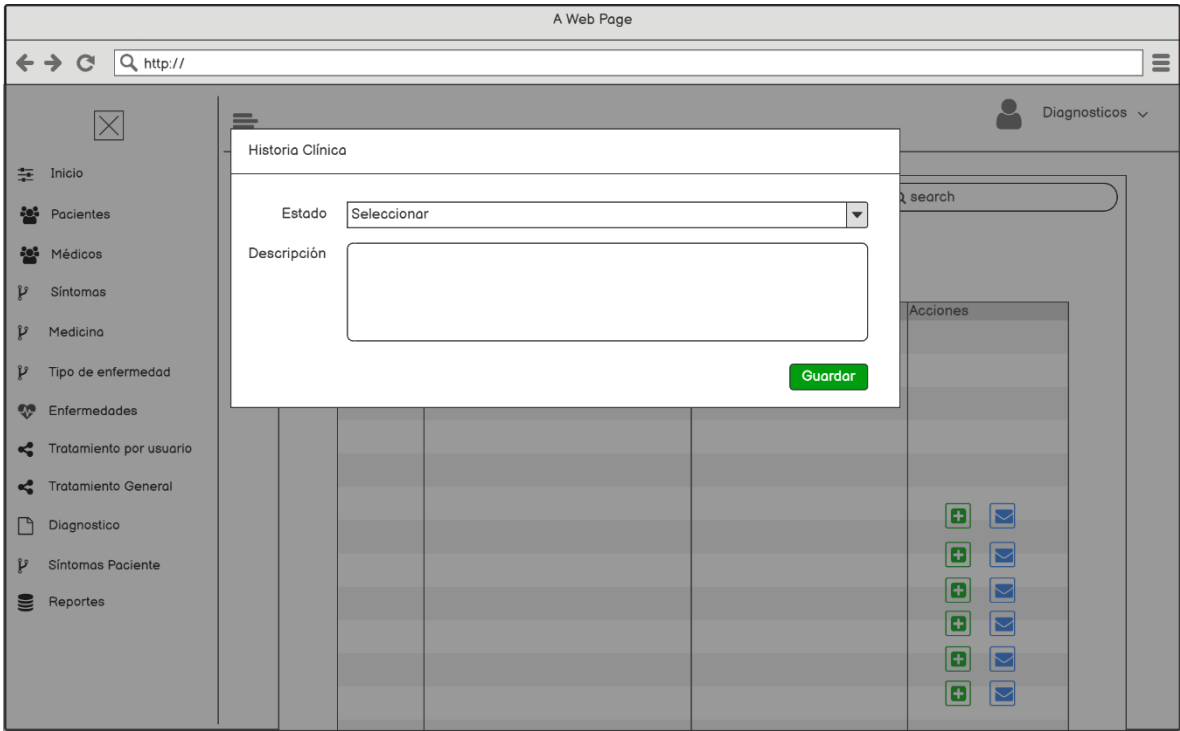
Recomendaciones:

**Figura\_ 17: Gestión de síntomas paciente**



**Figura\_ 18: Prototipo de Gestión de historia clínica**





## PLANIFICACIÓN DEL SPRINT N°1

Siendo las 4 pm del día 4 de octubre del 2021, se reúne en el Policlínico Monserrat.

Presentes:

ROL	NOMBRE
Scrum Master	Luz Gomez Taype
Team Member	Gianmarco Roger Nizama Antunez
Product Owner	Miguel Rios

El gerente del Policlínico Monserrat, ejecutó la presentación de todos los requerimientos obtenidos, de igual manera indica los que son de mayor prioridad.

Luego de ser analizada los requerimientos presentados por el gerente del Policlínico Monserrat. La señora **Luz Gomez** aclara ciertas dudas y da su compromiso para el cumplimiento de los requerimientos expuestos en el Sprint 1.

Así mismo los presentes imparten su aprobación respecto a lo expuesto esto específicamente en el Sprint 1, señalando que la fecha de entrega respecto a este Sprint será el 16 de octubre del 2021.


## EJECUCIÓN DEL SPRINT 1

Anterior a la fase del diseño se necesita saber exactamente el funcionamiento del sistema, conocer y entender cada parte de ella, dicho de una manera diferente el análisis respectivo a lo que se requiere esto de acuerdo a las historias de usuario.

**RF1: El sistema permitirá el inicio de sesión de dos tipos de usuarios, admin y super admin.**

### DISEÑO

#### PROTOTIPO

De la misma manera a continuación se muestra el prototipo desarrollado para este requerimiento, la cual fue presentado mediante el producto owner para poder continuar con la parte del desarrollo, el programa en el que se desarrollo fue el Balsamiq mockups.

**Figura\_ 19: Prototipo Inicio de sesión**

El prototipo muestra una ventana de navegador con el título "Sistema experto". En el centro de la página hay un formulario de inicio de sesión con el título "Bienvenidos". El formulario contiene tres campos de entrada: "Usuario:", "Clave:" y un botón etiquetado como "Button".

## DESARROLLO

En las siguientes figuras se pueden observar la parte del desarrollo del sistema, la cual va comenzando por el controlador, modelo y vista.

Figura\_ 20: Controlador de inicio de sesión

```
1 <?php
2
3 namespace App\Http\Controllers\Auth;
4
5 use App\Http\Controllers\Controller;
6 use App\Providers\RouteServiceProvider;
7 use Illuminate\Foundation\Auth\AuthenticatesUsers;
8
9 class LoginController extends Controller
10 {
11     /**
12      |-----
13      | Login Controller
14      |-----
15      |
16      | This controller handles authenticating users for the application and
17      | redirecting them to your home screen. The controller uses a trait
18      | to conveniently provide its functionality to your applications.
19      |
20      */
21
22     use AuthenticatesUsers;
23
24     /**
25      * Where to redirect users after login.
26      *
27      * @var string
28      */
29     protected $redirectTo = RouteServiceProvider::HOME;
30
31     /**
32      * Create a new controller instance.
33      *
34      * @return void
35      */
36     public function __construct()
37     {
38         $this->middleware('guest')->except('logout');
39     }
40 }
41
```



Figura\_ 21: Modelo de inicio de sesión

```
User.php x
1 k?php
2
3 namespace App\Models;
4
5 use Illuminate\Contracts\Auth\MustVerifyEmail;
6 use Illuminate\Database\Eloquent\SoftDeletes;
7 use Illuminate\Database\Eloquent\Factories\HasFactory;
8 use Illuminate\Foundation\Auth\User as Authenticatable;
9 use Illuminate\Notifications\Notifiable;
10
11 class User extends Authenticatable
12 {
13     use HasFactory, Notifiable, SoftDeletes;
14
15     protected $dates = ['deleted_at'];
16
17     protected $fillable = [
18         'id_perfil',
19         'id_especialidad',
20         'nombres',
21         'apellidos',
22         'dni',
23         'telefono',
24         'email',
25         'password',
26         'fecha_nacimiento',
27         'talla',
28         'peso'
29     ];
30
31     function perfil(){
32         return $this->belongsTo('App\Models\Perfil','id_perfil','id');
33     }
34
35     function especialidad(){
36         return $this->belongsTo('App\Models\Especialidad','id_especialidad','id');
37     }
38
39 }
40
```

Figura\_ 22: Vista de inicio de sesión

```
login.blade.php
1 <!doctype html>
2 <html lang="en">
3
4 <head>
5 <title>Login Sistema Experto</title>
6 <!-- Required meta tags -->
7 <meta charset="utf-8">
8 <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
9 <link rel="shortcut icon" href="{{ asset('assets/img/logo.png') }}" />
10 <!-- CSRF Token -->
11 <meta name="csrf-token" content="{{ csrf_token() }}">
12 <!-- Bootstrap -->
13 <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css" integrity="
sha384-Vkoo8x4CGSO3+Hhvxv8T/QSPaxtkktu6ug5TOENV6gBIFeWPGFN9MuhOF23Q9Ifjh" crossorigin="anonymous">
14 <!-- Fonts -->
15 <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@200;300;400;500;600;700;800;900&display=swap" rel
"stylesheet">
16 <style>
17
18 </style>
19 </head>
20
21 <body>
22 @include('sweetalert::alert')
23 <main class="login-p d-block d-md-flex">
24 <div class="img d-none d-md-block">
25 <!-- 
26 
27 </div>
28 <div class="form-log">
29 <div class="cont-form">
30 <div class="text-center pb-5">
31 
32 </div>
33 <div class="text-plane">
34 <div class="welcome">
35 <h1 class="d-none d-md-block">
36 Bienvenido a tu panel administrativo de
37 <span>
38 <a href="https://instafact.pe" target="_blank">Sistema Experto</a>
39 </span>
40 </h1>
41 <p class="descrip--s my-3">Administra, controla y organiza tu aplicación.</p>
42 </div>
43 </div>
44 {!! Form::open(['route' => 'login']) !!}
45 <div class="el-form-item el-form-item--feedback is-required is-no-asterisk el-form-item--medium">
46 <div class="el-form-item_content">
47 <div class="el-input el-input--medium el-input--prefix">
48 <input type="email" name="email" autocomplete="true" placeholder="Dirección de correo" clas
="el-input_inner" value="{{ old('email') }}" autofocus required>
49 <span class="el-input_prefix">
50 <i class="eva-hover">
51 <svg xmlns="http://www.w3.org/2000/svg" width="20px" height="24" viewBox="0 0 24 24
fill="#DCDFE6" class="eva eva-animation eva-icon-hover-undefined"><g data-name="
Layer 2"><g data-name="email"><rect width="24" height="24" opacity="0"></rect><path
d="M19 4H5a3 3 0 0 3 3v10a3 3 0 0 3 3h14a3 3 0 0 3 3v7a3 3 0 0 3 3zm-.67
2.12 10.75 5.67 6zM19 18H5a1 1 0 0 1-1-1V7.25L7.4 5.55a1 1 0 0 .62 1 1 0 0
-16-.2L20 7.25V17a1 1 0 0 1-1 1z"></path></g></g></svg>
52 </i>
53 </span>
54 </div>
55 </div>
56 </div>
57 </div>
```

```

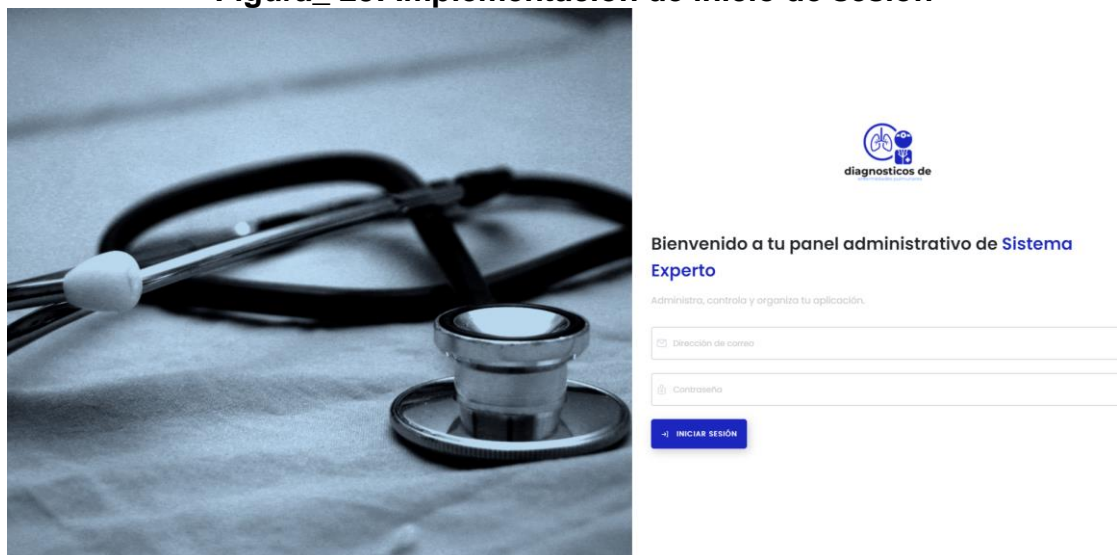
login.blade.php
246
247
248 <div class="el-form-item el-form-item--feedback is-required is-no-asterisk el-form-item--medium">
249 <div class="el-form-item__content">
250 <div class="el-input el-input--medium el-input--prefix">
251 <input type="password" name="password" autocomplete="off" placeholder="Contraseña" class="
252 el-input__inner" required>
253 <span class="el-input__prefix">
254 <i class="eva-hover">
255 <svg xmlns="http://www.w3.org/2000/svg" width="20px" height="24" viewBox="0 0 24 24"
256 fill="#000000" class="eva eva-animation eva-icon-hover-undefined"><g data-name="
257 Layer 2"><g data-name="lock"><rect width="24" height="24" opacity="0"></rect><path d
258 ="M17 8h-1v6.11a4 4 0 1 0-8 0v8h7a3 3 0 0 0-3 3v8a3 3 0 0 0 3 3h10a3 3 0 0 0
259 3-3v-8a3 3 0 0 0-3-3z" data-bbox="17 8 24 24"/></g></g></svg>
260 </i>
261 </span>
262 </div>
263 </div>
264 <div class="boton">
265 <button type="submit" class="btn btn-primary">
266 <i class="eva-hover mr-2" slot="prefix"><svg xmlns="http://www.w3.org/2000/svg" width="15px"
267 height="15px" viewBox="0 0 24 24" fill="#000000" class="eva eva-animation eva-icon-hover-undefined">
268 <g data-name="Layer 2"><g data-name="log-in"><rect width="24" height="24" transform="
269 rotate(-90 12 12)" opacity="0"></rect><path d="M19 4h-2a1 1 0 0 0 2h1v12h-1a1 1 0 0 0 2h2a1
270 1 0 0 1-1v5a1 1 0 0 1-1z" data-bbox="19 4 24 24"/></path><path d="M11.8 7.4a1 1 0 0 1-1.6 1.2l12 11H4a1 1 0 0 0
271 2h0.09l-1.72 2.44a1 1 0 0 0 .24 1.4 1 1 0 0 0 .58 1.8 1 1 0 0 0 .81-.42l2.82-4a1 1 0 0 0 0-1.18"
272 ></path></g></g></svg></i>
273 <span>
274 Iniciar Sesión
275 </span>
276 </button>
277 </div>
278 @error('email')
279 <div class="form-group pt-3 text-center">
280 {{ $message }}
281 </div>
282 @enderror
283 {!! Form::close() !!}
284 </div>
285 </main>
286 </body>
287 </html>

```

## IMPLEMENTACIÓN

En la siguiente figura se presenta la interfaz de este requerimiento, la cual fue definida anteriormente para proceder con el desarrollo.

Figura\_ 23: Implementación de inicio de sesión



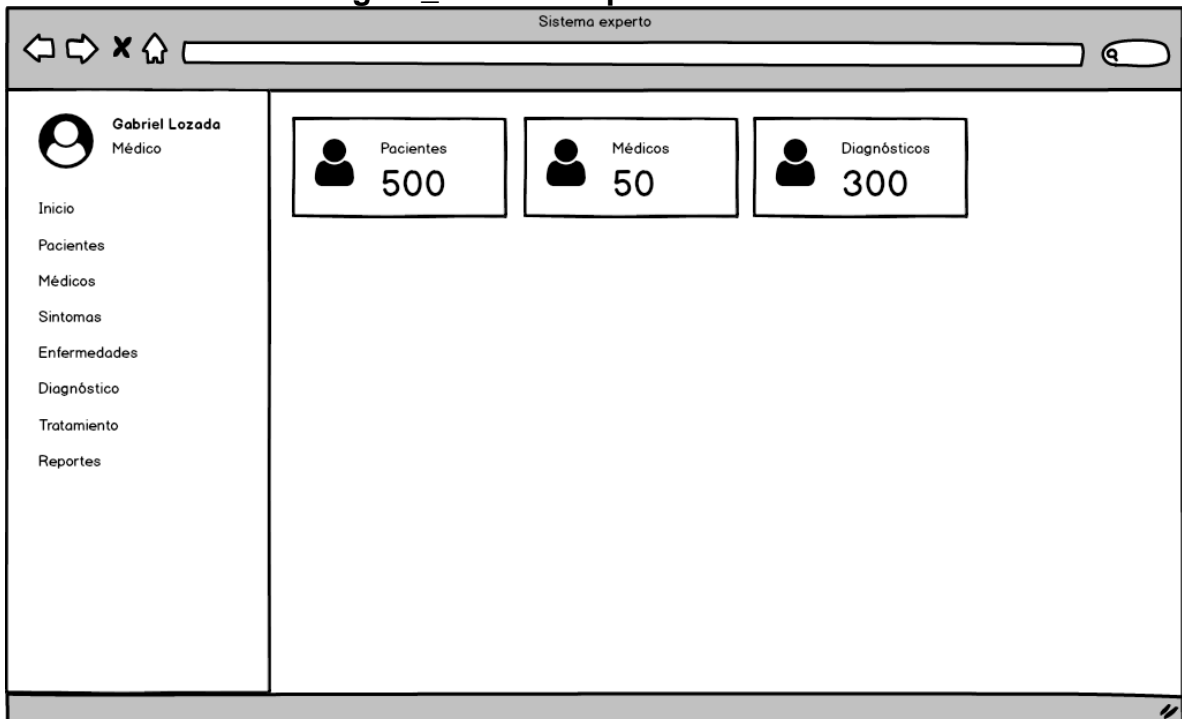
RF2: El sistema brindará un resumen de reportes a los usuarios.

## DISEÑO

## PROTOTIPO

De la misma manera a continuación se muestra el prototipo desarrollado para este requerimiento, la cual fue presentado mediante el producto owner para poder continuar con la parte del desarrollo, el programa en el que se desarrollo fue el Balsamiq mockups.

**Figura\_ 24: Prototipo Dashboard**



## DESARROLLO

En las siguientes figuras se pueden observar la parte del desarrollo del sistema, la cual va comenzando por el controlador, modelo y vista.

Figura\_ 25: Controlador de Dashboard

```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use App\Models\User;
7 use App\Models\Diagnostico;
8 use App\Models\Pedido;
9 use Illuminate\Support\Facades\Auth;
10
11 class HomeController extends Controller
12 {
13     /**
14      * Create a new controller instance.
15      *
16      * @return void
17      */
18     public function __construct()
19     {
20         $this->middleware('auth');
21     }
22
23     /**
24      * Show the application dashboard.
25      *
26      * @return \Illuminate\Contracts\Support\Renderable
27      */
28     public function index(Request $request)
29     {
30         if(Auth::user()->id_perfil != 2){
31             $pacientes = User::where('id_perfil',2)->count();
32             $medicos = User::where('id_perfil',3)->count();
33             $diagnosticos = Diagnostico::count();
34             return view('home',compact('pacientes','medicos','diagnosticos'));
35         }else{
36             Auth::logout();
37             $request->session()->invalidate();
38             $request->session()->regenerateToken();
39             toast('Acceso denegado','error')->timerProgressBar(3000);
40             return redirect(route('login'));
41         }
42     }
43 }
44
```

Figura\_ 26: Modelo de Dashboard

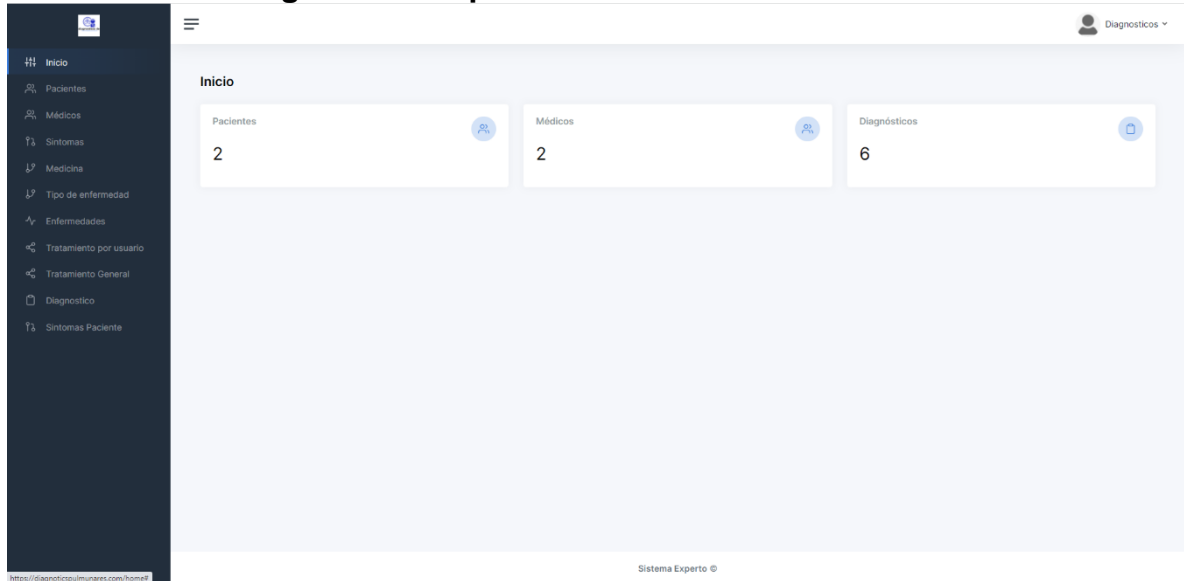
Figura\_ 27: Vista de Dashboard

```
home.blade.php
1 @extends('layouts.app')
2 @section('css')
3 <style type="text/css">
4     a:hover{
5         text-decoration: unset;
6     }
7 </style>
8 @endsection
9 @section('content')
10 <div class="row mb-2 mb-xl-3">
11     <div class="col-auto d-none d-sm-block">
12         <h3><strong>Inicio</strong></h3>
13     </div>
14 </div>
15 <div class="row">
16     <div class="col-sm-4">
17         <div class="card">
18             <a href="{{ route('user.paciente.index') }}">
19                 <div class="card-body">
20                     <div class="row">
21                         <div class="col mt-0">
22                             <h5 class="card-title">Pacientes</h5>
23                         </div>
24
25                         <div class="col-auto">
26                             <div class="stat text-primary">
27                                 <i class="align-middle data-feather="users"></i>
28                             </div>
29                         </div>
30                     </div>
31                     <h1 class="mt-1 mb-3">{{ $pacientes }}</h1>
32                 </div>
33             </a>
34         </div>
35     </div>
36     <div class="col-sm-4">
37         <div class="card">
38             <a href="{{ route('user.medico.index') }}">
39                 <div class="card-body">
40                     <div class="row">
41                         <div class="col mt-0">
42                             <h5 class="card-title">Médicos</h5>
43                         </div>
44
45                         <div class="col-auto">
46                             <div class="stat text-primary">
47                                 <i class="align-middle data-feather="users"></i>
48                             </div>
49                         </div>
50                     </div>
51                     <h1 class="mt-1 mb-3">{{ $medicos }}</h1>
52                 </div>
53             </a>
54         </div>
55     </div>
56     <div class="col-sm-4">
57         <div class="card">
58             <a href="{{ route('diagnostico.index') }}">
59                 <div class="card-body">
60                     <div class="row">
61                         <div class="col mt-0">
62                             <h5 class="card-title">Diagnósticos</h5>
63                         </div>
64
65                         <div class="col-auto">
66                             <div class="stat text-primary">
67                                 <i class="align-middle data-feather="clipboard"></i>
68                             </div>
69                         </div>
70                     </div>
71                     <h1 class="mt-1 mb-3">{{ $diagnosticos }}</h1>
72                 </div>
73             </a>
74         </div>
75     </div>
76 </div>
```

## IMPLEMENTACIÓN

En la siguiente figura se presenta la interfaz de este requerimiento, la cual fue definida anteriormente para proceder con el desarrollo.

**Figura\_ 28: Implementación de Dashboard**



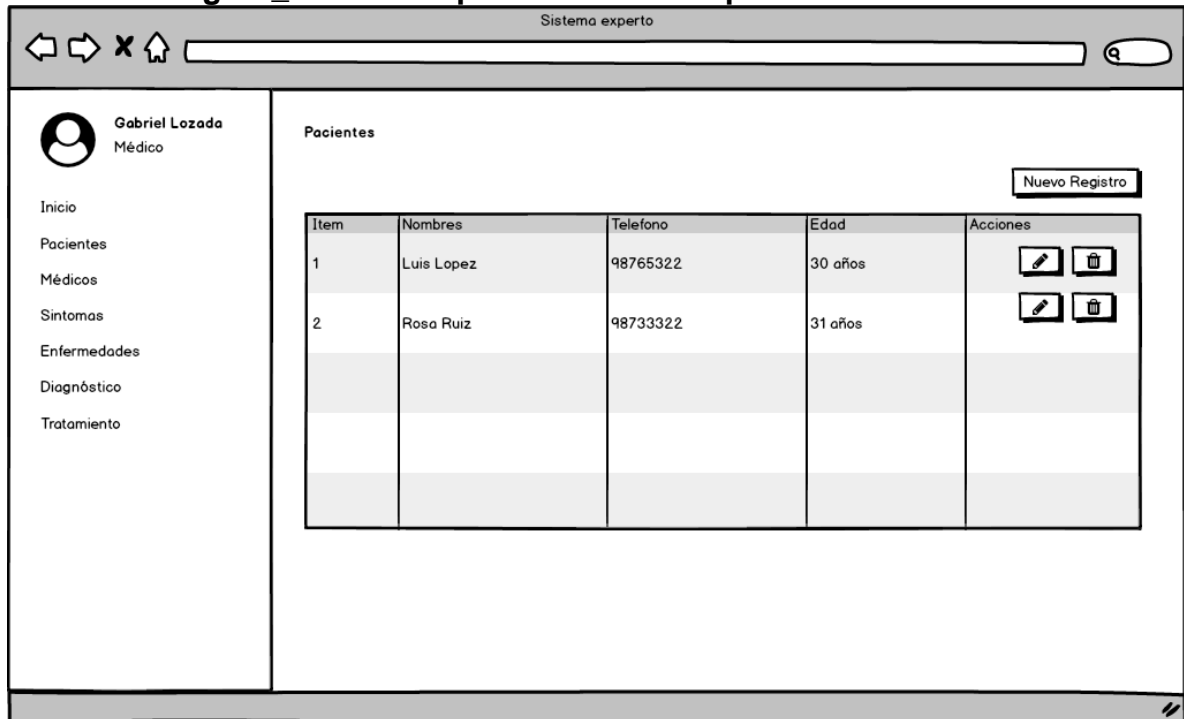
**RF3: El sistema permite gestionar la lista de los pacientes.**

## **DISEÑO**

### **PROTOTIPO**

De la misma manera a continuación se muestra el prototipo desarrollado para este requerimiento, la cual fue presentado mediante el producto owner para poder continuar con la parte del desarrollo, el programa en el que se desarrollo fue el Balsamiq mockups.

Figura\_ 29: Prototipo de Gestión de pacientes - Lista



## DESARROLLO

En las siguientes figuras se pueden observar la parte del desarrollo del sistema, la cual va comenzando por el controlador, modelo y vista.



Figura\_ 30: Controlador de Gestión de pacientes - Lista

```
UserController.php x
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use App\Models\User;
7 use App\Models\EnfermedadTratamiento;
8 use App\Models\Especialidad;
9 use Illuminate\Support\Facades\Auth;
10 use Alert;
11
12 class UserController extends Controller
13 {
14     public function indexPaciente()
15     {
16         return view('paciente.index');
17     }
18
19     public function indexMedico()
20     {
21         return view('medico.index');
22     }
23
24     public function listaPaciente()
25     {
26         $users = User::where('id_perfil',2)->orderByDesc('created_at')->get();
27
28         $data = collect();
29         foreach ($users as $key => $item) {
30             $route_edit = route('user.paciente.edit',$item->id);
31             $acciones="";
32             if($item->id != 1){
33                 $acciones .= '<a href="'.$route_edit.'" class="btn btn-warning" data-toggle="tooltip"
34                     title="Editar"><i class="fas fa-pen"></i></a>
35                 <button type="button" class="btn btn-danger" data-toggle="tooltip" title="Eliminar"
36                     onclick="deleteRegister('.$item->id.')"><i class="fas fa-trash-alt"></i></button>
37                 ';
38             }else{
39                 $acciones .= '<a href="'.$route_edit.'" class="btn btn-warning" data-toggle="tooltip"
40                     title="Editar"><i class="fas fa-pen"></i></a>';
41             }
42
43             $data_users = array(
44                 ($key+1),
45                 $item->dni,
46                 $item->nombres." ".$item->apellidos,
47                 $item->telefono,
48                 date('d-m-Y',strtotime($item->fecha_nacimiento)),
49                 $acciones
50             );
51             $data->push($data_users);
52         }
53         $results = array(
54             "draw" => 0,
55             "recordsTotal" => count($data),
56             "recordsFiltered" => count($data),
57             "data" => $data
58         );
59         return response()->json($results);
60     }
61 }
```

```

58 public function listaMedico()
59 {
60     $users = User::where('id_perfil',3)->orderByDesc('created_at')->get();
61
62     $data = collect();
63     foreach ($users as $key => $item) {
64         $route_edit = route('user.medico.edit',$item->id);
65         $acciones = "";
66         if($item->id != 1){
67             $acciones .= '<a href="'.$route_edit.'" class="btn btn-warning" data-toggle="tooltip"
68                 title="Editar"><i class="fas fa-pen"></i></a>
69                 <button type="button" class="btn btn-danger" data-toggle="tooltip" title="Eliminar"
70                 onclick="deleteRegister('.$item->id.')"><i class="fas fa-trash-alt"></i></button>
71             ';
72         }else{
73             $acciones .= '<a href="'.$route_edit.'" class="btn btn-warning" data-toggle="tooltip"
74                 title="Editar"><i class="fas fa-pen"></i></a>';
75         }
76         $data_users = array(
77             ($key),
78             $item->dni,
79             $item->nombres. " " . $item->apellidos,
80             $item->telefono,
81             $item->especialidad->nombre,
82             $acciones
83         );
84         $data->push($data_users);
85     }
86     $results = array(
87         "draw" => 0,
88         "recordsTotal" => count($data),
89         "recordsFiltered" => count($data),
90         "data" => $data
91     );
92     return response()->json($results);
93 }
94 public function createPaciente()
95 {
96     return view('paciente.create');
97 }
98
99 public function createMedico()
100 {
101     $lista_especialidad = Especialidad::all()->pluck('nombre','id')->prepend('Seleccionar','');
102     return view('medico.create',compact('lista_especialidad'));
103 }
104
105 public function storePaciente(Request $request)
106 {
107     $input = $request->all();
108     $input["id_perfil"] = 2;
109     $input["password"] = bcrypt($input['dni']);
110     User::create($input);
111     Alert::success('Usuario agregado', 'Registro guardado correctamente')->autoclose(5000);
112     return redirect(route('user.paciente.index'));
113 }
114
115 public function storeMedico(Request $request)
116 {
117     $input = $request->all();
118     $input["id_perfil"] = 3;
119     $input["password"] = bcrypt($input['dni']);
120     User::create($input);
121     Alert::success('Usuario agregado', 'Registro guardado correctamente')->autoclose(5000);
122     return redirect(route('user.medico.index'));
123 }

```

```

124 public function editPaciente($id)
125 {
126     $user = User::find($id);
127     return view('paciente.edit', compact('user'));
128 }
129
130 public function editMedico($id)
131 {
132     $user = User::find($id);
133     $lista_especialidad = Especialidad::all()->pluck('nombre', 'id')->prepend('seleccionar', '');
134     return view('medico.edit', compact('user', 'lista_especialidad'));
135 }
136
137 public function updatePaciente($id, Request $request)
138 {
139     $input = $request->all();
140     if (empty($input['password'])) {
141         $input['password'] = bcrypt($input['password']);
142     }else{
143         unset($input['password']);
144     }
145
146     unset($input['_method']);
147     unset($input['_token']);
148     User::where('id', $id)->update($input);
149     Alert::success('Usuario actualizado', 'Registro actualizado correctamente')->autoclose(5000);
150     return redirect(route('user.paciente.index'));
151 }
152
153 public function updateMedico($id, Request $request)
154 {
155     $input = $request->all();
156     if (empty($input['password'])) {
157         $input['password'] = bcrypt($input['password']);
158     }else{
159         unset($input['password']);
160     }
161
162     unset($input['_method']);
163     unset($input['_token']);
164     User::where('id', $id)->update($input);
165     Alert::success('Usuario actualizado', 'Registro actualizado correctamente')->autoclose(5000);
166     return redirect(route('user.medico.index'));
167 }
168
169 public function eliminar($id)
170 {
171     $result = true;
172     $message = null;
173     $enfermedad_tratamiento = EnfermedadTratamiento::where('id_usuario', $id)->first();
174     if($enfermedad_tratamiento){
175         $result = false;
176         $message = "Usuario en uso";
177     }else{
178         User::where('id', $id)->delete();
179         $message = "Usuario eliminado correctamente";
180     }
181     return response()->json(
182         array(
183             'result' => $result,
184             'message' => $message
185         )
186     );
187 }
188 }
189 }

```

Figura\_ 31: Modelo de Gestión de pacientes - Lista

```

1 k?php
2
3 namespace App\Models;
4
5 use Illuminate\Contracts\Auth\MustVerifyEmail;
6 use Illuminate\Database\Eloquent\SoftDeletes;
7 use Illuminate\Database\Eloquent\Factories\HasFactory;
8 use Illuminate\Foundation\Auth\User as Authenticatable;
9 use Illuminate\Notifications\Notifiable;
10
11 class User extends Authenticatable
12 {
13     use HasFactory, Notifiable, SoftDeletes;
14
15     protected $dates = ['deleted_at'];
16
17     protected $fillable = [
18         'id_perfil',
19         'id_especialidad',
20         'nombres',
21         'apellidos',
22         'dni',
23         'telefono',
24         'email',
25         'password',
26         'fecha_nacimiento',
27         'talla',
28         'peso'
29     ];
30
31     function perfil(){
32         return $this->belongsTo('App\Models\Perfil', 'id_perfil', 'id');
33     }
34
35     function especialidad(){
36         return $this->belongsTo('App\Models\Especialidad', 'id_especialidad', 'id');
37     }
38 }
39 }
40 }

```

Figura\_ 32: Vista de Gestión de pacientes - Lista

```

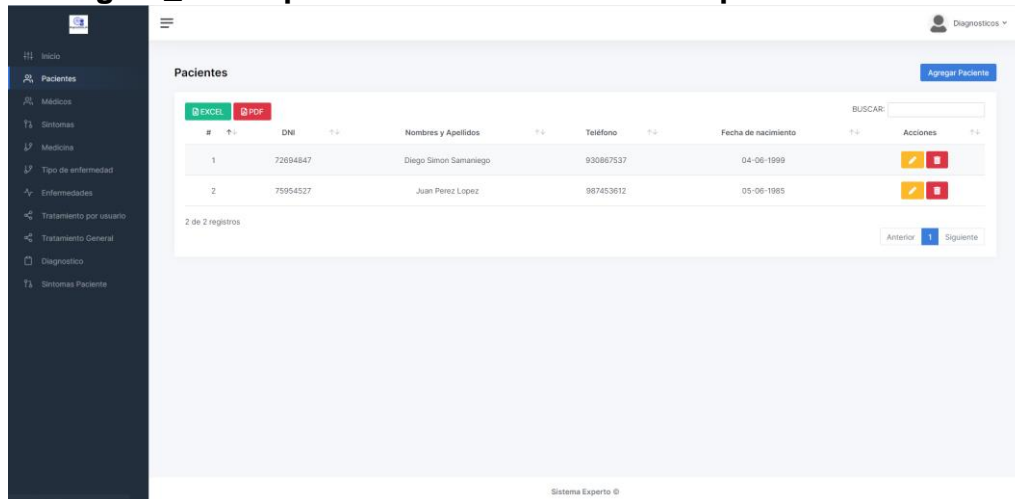
indexblade.php
1 @extends('layouts.app')
2 @section('content')
3 @include('sweetalert::alert')
4 <div class="row mb-2 mb-xl-3">
5 <div class="col-auto d-none d-sm-block">
6 <h3><strong>Pacientes</strong></h3>
7 </div>
8
9 <div class="col-auto ms-auto text-end mt-n1">
10 <a href="{{ route('user.paciente.create') }}" class="btn btn-primary">Agregar Paciente</a>
11 </div>
12 </div>
13 <div class="card">
14 <div class="card-body">
15 <table id="pacientes" class="table table-striped text-center" style="width:100%">
16 <thead>
17 <tr>
18 <th>#</th>
19 <th>DNI</th>
20 <th>Nombres y Apellidos</th>
21 <th>Teléfono</th>
22 <th>Fecha de nacimiento</th>
23 <th class="no-export">Acciones</th>
24 </tr>
25 </thead>
26 </table>
27 </div>
28 </div>
29 @endsection
30
31 @section('js')
32 <script>
33
34 var table = $('#pacientes').DataTable( {
35     "language": {
36         "lengthMenu": "Ver los MENU Primeros Registros",
37         "info": " _END_ de TOTAL registros",
38         "infoEmpty": "No se encontraron registros",
39         "infoFiltered": "(Filtrado de _MAX_ total entradas)",
40         "loadingRecords": "Cargando...",
41         "processing": "Procesando...",
42         "sSearch": "BUSCAR:",
43         "sZeroRecords": "No se encontraron resultados",
44         "sEmptyTable": "Ningún dato disponible en esta tabla",
45         "oPaginate": {
46             "sFirst": "Primero",
47             "sLast": "Último",
48             "sNext": "Siguiente",
49             "sPrevious": "Anterior"
50         },
51         "fnInfoCallback": null
52     },
53     "aProcessing": true,
54     "asServerSide": true,
55     "destroy": true,
56     "ajax": {
57         "method": "GET",
58         "url": "{{ route('user.paciente.lista') }}"
59     },
60     "iDisplayLength": 10,
61     dom: 'Bfrtip',
62     buttons: [
63         {
64             extends: 'excelHtml5',
65             title: 'Sistema experto - pacientes'
66         },
67         {
68             exportOptions: { columns: ":not(.no-export)" }
69         },
70         {
71             extend: 'pdfHtml5',
72             title: 'Sistema experto - pacientes',
73             orientation: 'landscape',
74             className: 'btn btn-danger',
75             text: '<i class="far fa-file-pdf"></i> PDF',
76             exportOptions: { columns: ":not(.no-export)" }
77         },
78     ],
79 });
80
81 function deleteRegister(id){
82     Swal.fire({
83         title: 'Esta seguro de eliminar este registro?',
84         text: "",
85         icon: 'warning',
86         showCancelButton: true,
87         confirmButtonColor: '#3085d6',
88         cancelButtonColor: '#d33',
89         confirmButtonText: 'Confirmar',
90         cancelButtonText: 'Cancelar'
91     }).then((result) => {
92         if (result.value) {
93             $.ajax({
94                 url: "{{URL::to('/user')}}"/ + "/" + id,
95                 type: 'DELETE'
96             })
97             .done(function(res) {
98                 if(res.result){
99                     table.ajax.reload();
100                     Swal.fire(res.message, "", "success");
101                 }else{
102                     Swal.fire(res.message, "", "error");
103                 }
104             });
105         }
106     })
107 }
108 </script>
109 @endsection

```

## IMPLEMENTACIÓN

En la siguiente figura se presenta la interfaz de este requerimiento, la cual fue definida anteriormente para proceder con el desarrollo.

**Figura\_ 33: Implementación de Gestión de pacientes - Lista**



The screenshot displays a web application interface for patient management. On the left is a dark sidebar with a menu containing items like 'Inicio', 'Pacientes', 'Médicos', 'Síntomas', 'Medicina', 'Tipo de enfermedad', 'Enfermedades', 'Tratamiento por caso', 'Tratamiento General', 'Diagnostico', and 'Síntomas Paciente'. The main content area is titled 'Pacientes' and features a table with columns for '#', 'DNI', 'Nombres y Apellidos', 'Teléfono', 'Fecha de nacimiento', and 'Acciones'. There are also buttons for 'EXCEL' and 'PDF' at the top left of the table, and a search bar labeled 'BUSCAR:' at the top right. A blue button 'Agregar Paciente' is located in the top right corner of the main area. The table contains two rows of patient data. At the bottom of the table, it indicates '2 de 2 registros' and has navigation buttons for 'Anterior', '1', and 'Siguiente'. The footer of the page reads 'Sistema Experto ©'.

#	DNI	Nombres y Apellidos	Teléfono	Fecha de nacimiento	Acciones
1	72984847	Diego Simon Samaniego	930887537	04-06-1999	[Iconos de acciones]
2	75954527	Juan Perez Lopez	987453012	05-06-1985	[Iconos de acciones]

**RF4: El sistema permitirá el registro de los datos del paciente, dni, nombres, apellidos, teléfono, correo, fecha de nacimiento, talla y peso.**

## DISEÑO

### PROTOTIPO

De la misma manera a continuación se muestra el prototipo desarrollado para este requerimiento, la cual fue presentado mediante el producto owner para poder continuar con la parte del desarrollo, el programa en el que se desarrollo fue el Balsamiq mockups.

Figura\_ 34: Prototipo Gestión de pacientes – Registro

El prototipo muestra una ventana de navegador con el título "Sistema experto". En la parte superior izquierda hay iconos de navegación (atrás, adelante, cerrar, inicio) y una barra de búsqueda. A la izquierda se encuentra un menú de navegación con un perfil de usuario "Gabriel Lozada Médico" y opciones como Inicio, Pacientes, Médicos, Síntomas, Enfermedades, Diagnóstico y Tratamiento. El área principal, titulada "Pacientes", contiene un formulario con los siguientes campos: DNI, Nombres, Apellidos, Teléfono, Correo, Fecha N. (con un icono de calendario), Talla y Peso. Al final del formulario hay dos botones: "Cancelar" y "Guardar".

## DESARROLLO

En las siguientes figuras se pueden observar la parte del desarrollo del sistema, la cual va comenzando por el controlador, modelo y vista.

Figura\_ 35: Controlador de Gestión de pacientes - Registro

```
UserController.php x
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use App\Models\User;
7 use App\Models\EnfermedadTratamiento;
8 use App\Models\Especialidad;
9 use Illuminate\Support\Facades\Auth;
10 use Alert;
11
12 class UserController extends Controller
13 {
14     public function indexPaciente()
15     {
16         return view('paciente.index');
17     }
18
19     public function indexMedico()
20     {
21         return view('medico.index');
22     }
23
24     public function listaPaciente()
25     {
26         $users = User::where('id_perfil',2)->orderByDesc('created_at')->get();
27
28         $data = collect();
29         foreach ($users as $key => $item) {
30             $route_edit = route('user.paciente.edit',$item->id);
31             $acciones="";
32             if($item->id != 1){
33                 $acciones .= '<a href="'.$route_edit.'" class="btn btn-warning" data-toggle="tooltip"
34                 title="Editar"><i class="fas fa-pen"></i></a>
35                 <button type="button" class="btn btn-danger" data-toggle="tooltip" title="Eliminar"
36                 onclick="deleteRegister('.$item->id.')"><i class="fas fa-trash-alt"></i></button>
37                 ';
38             }else{
39                 $acciones .= '<a href="'.$route_edit.'" class="btn btn-warning" data-toggle="tooltip"
40                 title="Editar"><i class="fas fa-pen"></i></a>';
41             }
42
43             $data_users = array(
44                 ($key+1),
45                 $item->dni,
46                 $item->nombres." ".$item->apellidos,
47                 $item->telefono,
48                 date('d-m-Y',strtotime($item->fecha_nacimiento)),
49                 $acciones
50             );
51             $data->push($data_users);
52         }
53         $results = array(
54             "draw" => 0,
55             "recordsTotal" => count($data),
56             "recordsFiltered" => count($data),
57             "data" => $data
58         );
59         return response()->json($results);
60     }
61 }
```

```

58 public function listaMedico()
59 {
60     $users = User::where('id_perfil',3)->orderByDesc('created_at')->get();
61
62     $data = collect();
63     foreach ($users as $key => $item) {
64         $route_edit = route('user.medico.edit',$item->id);
65         $acciones = "";
66         if($item->id != 1){
67             $acciones .= '<a href="'.$route_edit.'" class="btn btn-warning" data-toggle="tooltip"
68                 title="Editar"><i class="fas fa-pen"></i></a>
69             <button type="button" class="btn btn-danger" data-toggle="tooltip" title="Eliminar"
70                 onclick="deleteRegister('.$item->id.')"><i class="fas fa-trash-alt"></i></button>
71             ';
72         }else{
73             $acciones .= '<a href="'.$route_edit.'" class="btn btn-warning" data-toggle="tooltip"
74                 title="Editar"><i class="fas fa-pen"></i></a>';
75         }
76         $data_users = array(
77             ($key),
78             $item->dni,
79             $item->nombres. " " . $item->apellidos,
80             $item->telefono,
81             $item->especialidad->nombre,
82             $acciones
83         );
84         $data->push($data_users);
85     }
86     $results = array(
87         "draw" => 0,
88         "recordsTotal" => count($data),
89         "recordsFiltered" => count($data),
90         "data" => $data
91     );
92     return response()->json($results);
93 }
94 public function createPaciente()
95 {
96     return view('paciente.create');
97 }
98
99 public function createMedico()
100 {
101     $lista_especialidad = Especialidad::all()->pluck('nombre','id')->prepend('Seleccionar,');
102     return view('medico.create',compact('lista_especialidad'));
103 }
104
105 public function storePaciente(Request $request)
106 {
107     $input = $request->all();
108     $input["id_perfil"] = 2;
109     $input["password"] = bcrypt($input['dni']);
110     User::create($input);
111     Alert::success('Usuario agregado', 'Registro guardado correctamente')->autoclose(5000);
112     return redirect(route('user.paciente.index'));
113 }
114
115 public function storeMedico(Request $request)
116 {
117     $input = $request->all();
118     $input["id_perfil"] = 3;
119     $input["password"] = bcrypt($input['dni']);
120     User::create($input);
121     Alert::success('Usuario agregado', 'Registro guardado correctamente')->autoclose(5000);
122     return redirect(route('user.medico.index'));
123 }

```



```

124 public function editPaciente($id)
125 {
126     $user = User::find($id);
127     return view('paciente.edit', compact('user'));
128 }
129
130 public function editMedico($id)
131 {
132     $user = User::find($id);
133     $lista_especialidad = Especialidad::all()->pluck('nombre', 'id')->xprepend('seleccionar', '');
134     return view('medico.edit', compact('user', 'lista_especialidad'));
135 }
136
137 public function updatePaciente($id, Request $request)
138 {
139     $input = $request->all();
140     if (empty($input['password'])) {
141         $input['password'] = bcrypt($input['password']);
142     }else{
143         unset($input['password']);
144     }
145
146     unset($input['_method']);
147     unset($input['_token']);
148     User::where('id', $id)->update($input);
149     Alert::success('Usuario actualizado', 'Registro actualizado correctamente')->autoclose(5000);
150     return redirect(route('user.paciente.index'));
151 }
152
153 public function updateMedico($id, Request $request)
154 {
155     $input = $request->all();
156     if (empty($input['password'])) {
157         $input['password'] = bcrypt($input['password']);
158     }else{
159         unset($input['password']);
160     }
161
162     unset($input['_method']);
163     unset($input['_token']);
164     User::where('id', $id)->update($input);
165     Alert::success('Usuario actualizado', 'Registro actualizado correctamente')->autoclose(5000);
166     return redirect(route('user.medico.index'));
167 }
168
169 public function eliminar($id)
170 {
171     $result = true;
172     $message = null;
173     $enfermedad_tratamiento = EnfermedadTratamiento::where('id_usuario', $id)->first();
174     if($enfermedad_tratamiento){
175         $result = false;
176         $message = "Usuario en uso";
177     }else{
178         User::where('id', $id)->delete();
179         $message = "Usuario eliminado correctamente";
180     }
181     return response()->json(
182         array(
183             'result' => $result,
184             'message' => $message
185         )
186     );
187 }
188 }
189 }

```

Figura\_ 36: Modelo de Gestión de pacientes - Registro

```

1 k?php
2
3 namespace App\Models;
4
5 use Illuminate\Contracts\Auth\MustVerifyEmail;
6 use Illuminate\Database\Eloquent\SoftDeletes;
7 use Illuminate\Database\Eloquent\Factories\HasFactory;
8 use Illuminate\Foundation\Auth\User as Authenticatable;
9 use Illuminate\Notifications\Notifiable;
10
11 class User extends Authenticatable
12 {
13     use HasFactory, Notifiable, SoftDeletes;
14
15     protected $dates = ['deleted_at'];
16
17     protected $fillable = [
18         'id_perfil',
19         'id_especialidad',
20         'nombres',
21         'apellidos',
22         'dni',
23         'telefono',
24         'email',
25         'password',
26         'fecha_nacimiento',
27         'talla',
28         'peso'
29     ];
30
31     function perfil(){
32         return $this->belongsTo('App\Models\Perfil', 'id_perfil', 'id');
33     }
34
35     function especialidad(){
36         return $this->belongsTo('App\Models\Especialidad', 'id_especialidad', 'id');
37     }
38 }
39 }
40 }

```

Figura\_ 37: Vista de Gestión de pacientes - Registro

```

indexblade.php
1 @extends('layouts.app')
2 @section('content')
3 @include('sweetalert::alert')
4 <div class="row mb-2 mb-xl-3">
5 <div class="col-auto d-none d-sm-block">
6 <h3><strong>Pacientes</strong></h3>
7 </div>
8
9 <div class="col-auto ms-auto text-end mt-n1">
10 <a href="{{ route('user.paciente.create') }}" class="btn btn-primary">Agregar Paciente</a>
11 </div>
12 </div>
13 <div class="card">
14 <div class="card-body">
15 <table id="pacientes" class="table table-striped text-center" style="width:100%">
16 <thead>
17 <tr>
18 <th>#</th>
19 <th>DNI</th>
20 <th>Nombres y Apellidos</th>
21 <th>Teléfono</th>
22 <th>Fecha de nacimiento</th>
23 <th class="no-export">Acciones</th>
24 </tr>
25 </thead>
26 </table>
27 </div>
28 </div>
29 @endsection
30
31 @section('js')
32 <script>
33
34 var table = $('#pacientes').DataTable( {
35     "language": {
36         "lengthMenu": "Ver los MENU Primeros Registros",
37         "info": " _END_ de TOTAL registros",
38         "infoEmpty": "No se encontraron registros",
39         "infoFiltered": "(Filtrado de _MAX_ total entradas)",
40         "loadingRecords": "Cargando...",
41         "processing": "Procesando...",
42         "sSearch": "BUSCAR:",
43         "sZeroRecords": "No se encontraron resultados",
44         "sEmptyTable": "Ningún dato disponible en esta tabla",
45         "oPaginate": {
46             "sFirst": "Primero",
47             "sLast": "Último",
48             "sNext": "Siguiente",
49             "sPrevious": "Anterior"
50         },
51         "fnInfoCallback": null
52     },
53     "aProcessing": true,
54     "asServerSide": true,
55     "destroy":true,
56     "ajax":{
57         "method":"GET",
58         "url": "{{ route('user.paciente.lista') }}"
59     },
60     "iDisplayLength":10,
61     dom: 'Bfrtip',
62     buttons: [
63         {
64             extend: 'excelHtml5',
65             title: 'Sistema experto - pacientes'
66         },
67         {
68             exportOptions: { columns: ":not(.no-export)" }
69         },
70         {
71             extend: 'pdfHtml5',
72             title: 'Sistema experto - pacientes',
73             orientation: 'landscape',
74             className: 'btn btn-danger',
75             text: '<i class="far fa-file-pdf"></i> PDF',
76             exportOptions: { columns: ":not(.no-export)" }
77         },
78     ],
79 });
80
81 function deleteRegister(id){
82     Swal.fire({
83         title: 'Esta seguro de eliminar este registro?',
84         text: "",
85         icon: 'warning',
86         showCancelButton: true,
87         confirmButtonColor: '#3085d6',
88         cancelButtonColor: '#d33',
89         confirmButtonText: 'Confirmar',
90         cancelButtonText: 'Cancelar'
91     }).then((result) => {
92         if (result.value) {
93             $.ajax({
94                 url: "{{URL::to('/user')}}" + "/" + id,
95                 type: 'DELETE'
96             })
97             .done(function(res) {
98                 if(res.result){
99                     table.ajax.reload();
100                     Swal.fire(res.message, "", "success");
101                 }else{
102                     Swal.fire(res.message, "", "error");
103                 }
104             });
105         }
106     })
107 }
108 </script>
109 @endsection
110

```

## IMPLEMENTACIÓN

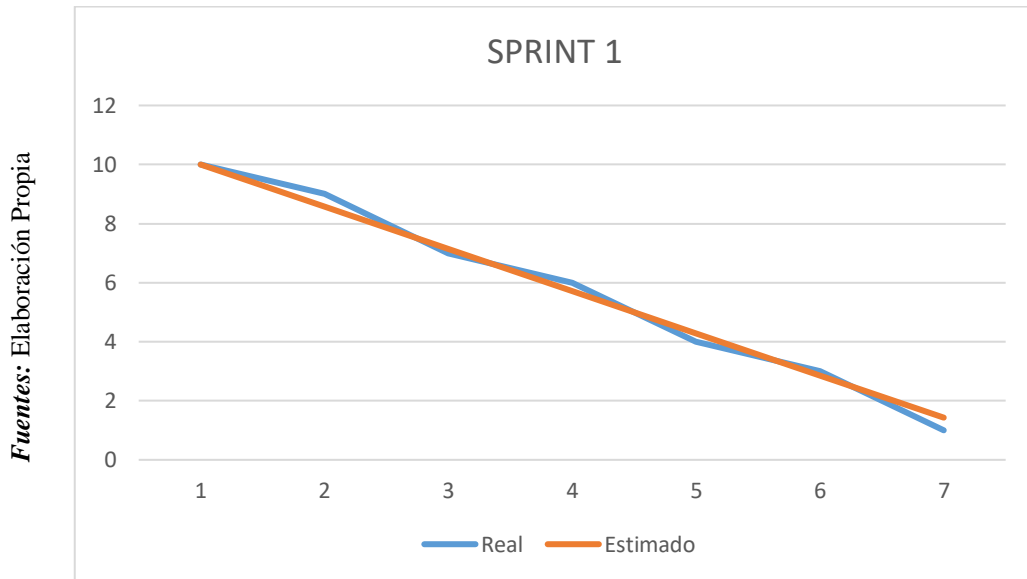
En la siguiente figura se presenta la interfaz de este requerimiento, la cual fue definida anteriormente para proceder con el desarrollo.

**Figura\_ 38: Implementación de Gestión de pacientes – Registro**

The screenshot displays a web application interface for patient registration. On the left, a dark sidebar contains a navigation menu with the following items: Inicio, Pacientes (highlighted), Médico, Síntomas, Medicina, Tipo de enfermedad, Enfermedades, Tratamiento por usuario, Tratamiento General, Diagnóstico, and Síntomas Paciente. The main content area is titled "Nuevo Paciente" and features a form with the following fields: DNI, Nombres, Apellidos, Teléfono, Correo electrónico, Fecha de Nacimiento (with a date picker), Talla, and Peso. At the bottom of the form, there are two buttons: a red "Cancelar" button and a blue "Guardar" button. The footer of the page reads "Sistema Experto ©".

## BURNDOWN DEL SPRINT N° 1

Figura\_ 39: Burndown del Sprint 1



### Burndown Sprint 1

A continuación, se presentará el significado de la figura anterior: comenzando con la línea roja la cual representa el tiempo ideal del desarrollo de la misma, así mismo la línea azul representa el tiempo real en la que se desarrolló dicho sprint. Es así como se puede decir que si la línea azul se ubica por debajo de la línea roja es porque existió un adelanto en la elaboración del sprint, y de manera contraria si la línea azul está por encima de la roja significa retraso en la elaboración del sprint.

## ACTA DE REUNION DEL SPRINT N° 1

Siendo las 3 pm del día 4 de octubre del 2021, se reúne en el Policlínico Monserrat.

Presentes:

ROL	NOMBRE
Scrum Master	Luz Gomez Taype
Team Member	Gianmarco Roger Nizama Antunez
Product Owner	Miguel Rios

El encargado del Policlínico Monserrat, Luz Gomez Taype culmina la reunión con el señor Gianmarco Nizama señalando los últimos puntos respecto al sprint.

De esta manera se logra definir la forma de trabajo y todos los requerimientos respecto a este sprint, así mismo firmar el acta que contiene la fecha de entrega, esto para el cierre de la reunión.

Los asistentes presentes aprobaron los puntos que fueron acordados en esta reunión, así mismo beneficio la planificación de este Sprint, realizando el compromiso de entregar los resultados referentes al tiempo que fue pactado.


## ACTA DE ENTREGA DEL SPRINT N° 1

Siendo la 1 pm del día 16 de octubre del 2021 se reúne en el Policlínico Monserrat.

ROL	NOMBRE
Scrum Master	Luz Gomez Taype
Team Member	Gianmarco Roger Nizama Antunez
Product Owner	Miguel Rios

El Sr. Gianmarco Nizama, señala y detalla cada requerimiento desarrollado, así mismo presenta las interfaces obtenidas del product owner.

Luego de realizar las verificaciones a la sustentación del Sr. Gianmarco Nizama para continuar con la aprobación del Sprint, de manera unánime se decide la aprobación de este sprint, referente al proyecto “Sistema experto para diagnóstico de enfermedades pulmonares en el policlínico Monserrat”.

Es así como los presentes imparten su aprobación al informe del Sr. Gianmarco Nizama, referente al Sprint N° 1 la cual fue concluida, del siguiente proyecto “Sistema experto para diagnóstico de enfermedades pulmonares en el policlínico Monserrat”.


## RESUMEN DE LA REUNIÓN RETROSPECTIVA DE SPRINT N° 1

Información de la empresa y proyecto:

<b>Empresa / Organización</b>	Policlínico Monserrat.
<b>Proyecto</b>	“Sistema experto para diagnóstico de enfermedades pulmonares en el policlínico Monserrat”.

Información de la reunión:

<b>Lugar</b>	Policlínico Monserrat.
<b>Fecha</b>	16/10/2021
<b>Número de iteración / Sprint</b>	Sprint 1
<b>Personas Convocadas a la reunión</b>	
<b>Persona que asistieron a la reunión</b>	

Formulario de reunión retrospectiva

<b>¿Qué salió bien en la Iteración? (Aciertos)</b>	<b>¿Qué no salió bien en la Iteración? (Errores)</b>
<p>Se logró ejecutar de manera correcta el inicio de sesión y todas las validaciones de los privilegios.</p> <p>Se desarrolló el Dashboard.</p> <p>Se logró ejecutar la lista de pacientes para la gestión.</p> <p>De igual manera el registro de cada paciente.</p>	<p>Se produjo un fallo en las validaciones de los privilegios, pero se solucionó.</p>

## PLANIFICACIÓN DEL SPRINT N°2

Siendo las 4 pm del día 16 de octubre del 2021, se reúne en el Policlínico Monserrat.

Presentes:

<b>ROL</b>	<b>NOMBRE</b>
Scrum Master	Luz Gomez Taype
Team Member	Gianmarco Roger Nizama Antunez
Product Owner	Miguel Rios

El gerente del Policlínico Monserrat, ejecutó la presentación de todos los requerimientos obtenidos, de igual manera indica los que son de mayor prioridad.

Luego de ser analizada los requerimientos presentados por el gerente del Policlínico Monserrat. La señora Luz Gomez aclara ciertas dudas y da su compromiso para el cumplimiento de los requerimientos expuestos en el Sprint 2.

Así mismo los presentes imparten su aprobación respecto a lo expuesto esto específicamente en el Sprint 2, señalando que la fecha de entrega respecto a este Sprint será el 2 de noviembre del 2021.




## EJECUCIÓN DEL SPRINT 2

Anterior a la fase del diseño se necesita saber exactamente el funcionamiento del sistema, conocer y entender cada parte de ella, dicho de una manera diferente el análisis respectivo a lo que se requiere esto de acuerdo a las historias de usuario.

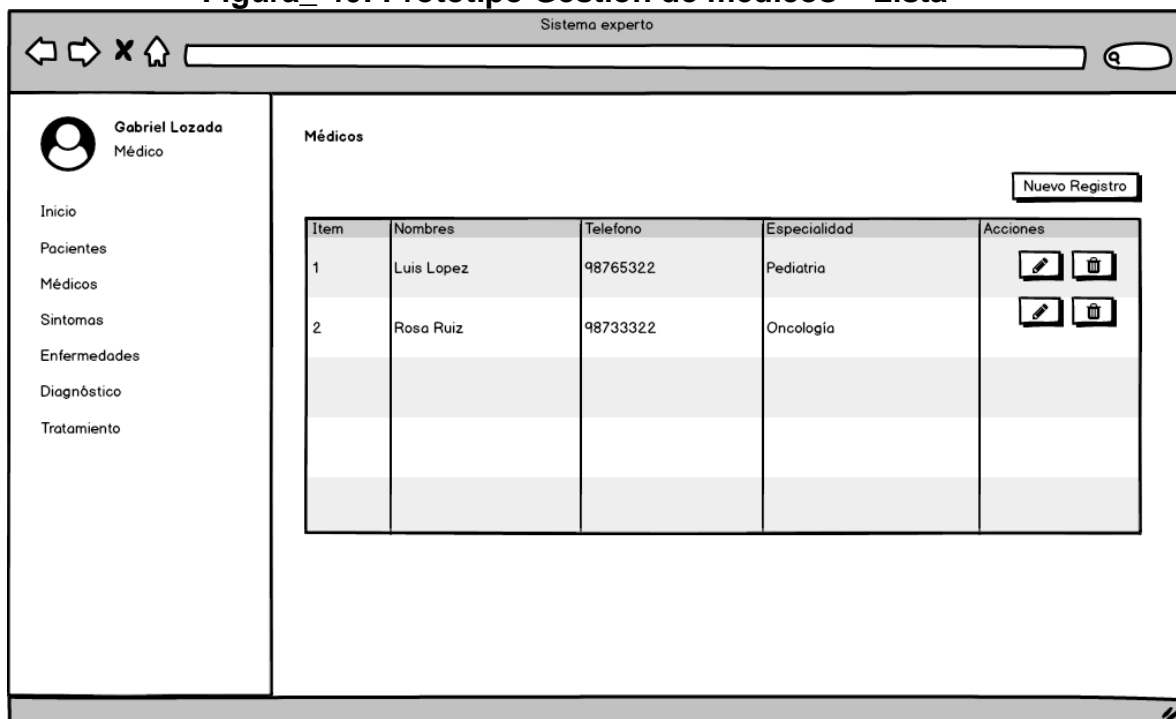
**RF5: El sistema permitirá gestionar la lista de todos los médicos registrados.**

### DISEÑO

#### PROTOTIPO

De la misma manera a continuación se muestra el prototipo desarrollado para este requerimiento, la cual fue presentado mediante el producto owner para poder continuar con la parte del desarrollo, el programa en el que se desarrollo fue el Balsamiq mockups.

**Figura\_ 40: Prototipo Gestión de médicos – Lista**



### DESARROLLO

En las siguientes figuras se pueden observar la parte del desarrollo del sistema, la cual va comenzando por el controlador, modelo y vista.

**Figura\_ 41: Controlador de Gestión de médicos – Lista**

```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use App\Models\User;
7 use App\Models\EnfermedadTratamiento;
8 use App\Models\Especialidad;
9 use Illuminate\Support\Facades\Auth;
10 use Alert;
11
12 class UserController extends Controller
13 {
14     public function indexPaciente()
15     {
16         return view('paciente.index');
17     }
18
19     public function indexMedico()
20     {
21         return view('medico.index');
22     }
23
24     public function listaPaciente()
25     {
26         $users = User::where('id_perfil',2)->orderByDesc('created_at')->get();
27
28         $data = collect();
29         foreach ($users as $key => $item) {
30             $route_edit = route('user.paciente.edit',$item->id);
31             $acciones="";
32             if($item->id != 1){
33                 $acciones .= '<a href="'.$route_edit.'" class="btn btn-warning" data-toggle="tooltip"
34                             title="Editar"><i class="fas fa-pen"></i></a>
35                 <button type="button" class="btn btn-danger" data-toggle="tooltip" title="Eliminar"
36                             onclick="deleteRegister('.$item->id.')"><i class="fas fa-trash-alt"></i></button>
37             ';
38             }else{
39                 $acciones .= '<a href="'.$route_edit.'" class="btn btn-warning" data-toggle="tooltip"
40                             title="Editar"><i class="fas fa-pen"></i></a>';
41             }
42
43             $data_users = array(
44                 ($key+1),
45                 $item->dni,
46                 $item->nombres." ".$item->apellidos,
47                 $item->telefono,
48                 date('d-m-Y',strtotime($item->fecha_nacimiento)),
49                 $acciones
50             );
51             $data->push($data_users);
52         }
53         $results = array(
54             "draw" => 0,
55             "recordsTotal" => count($data),
56             "recordsFiltered" => count($data),
57             "data" => $data
58         );
59
60         return response()->json($results);
61     }
62 }
```

```

58 public function listaMedico()
59 {
60     $users = User::where('id_perfil',3)->orderByDesc('created_at')->get();
61
62     $data = collect();
63     foreach ($users as $key => $item) {
64         $route_edit = route('user.medico.edit',$item->id);
65         $acciones = "";
66         if($item->id != 1){
67             $acciones .= '<a href="'.$route_edit.'" class="btn btn-warning" data-toggle="tooltip"
68                 title="Editar"><i class="fas fa-pen"></i></a>
69                 <button type="button" class="btn btn-danger" data-toggle="tooltip" title="Eliminar"
70                 onclick="deleteRegister('.$item->id.')"><i class="fas fa-trash-alt"></i></button>
71         }else{
72             $acciones .= '<a href="'.$route_edit.'" class="btn btn-warning" data-toggle="tooltip"
73                 title="Editar"><i class="fas fa-pen"></i></a>';
74         }
75         $data_users = array(
76             ($key),
77             $item->dni,
78             $item->nombres. " " . $item->apellidos,
79             $item->telefono,
80             $item->especialidad->nombre,
81             $acciones
82         );
83         $data->push($data_users);
84     }
85     $results = array(
86         "draw" => 0,
87         "recordsTotal" => count($data),
88         "recordsFiltered" => count($data),
89         "data" => $data
90     );
91     return response()->json($results);
92 }
93
94 public function createPaciente()
95 {
96     return view('paciente.create');
97 }
98
99 public function createMedico()
100 {
101     $lista_especialidad = Especialidad::all()->pluck('nombre','id')->prepend('Seleccionar','');
102     return view('medico.create',compact('lista_especialidad'));
103 }
104
105 public function storePaciente(Request $request)
106 {
107     $input = $request->all();
108     $input["id_perfil"] = 2;
109     $input["password"] = bcrypt($input['dni']);
110     User::create($input);
111     Alert::success('Usuario agregado', 'Registro guardado correctamente')->autoclose(5000);
112     return redirect(route('user.paciente.index'));
113 }
114
115 public function storeMedico(Request $request)
116 {
117     $input = $request->all();
118     $input["id_perfil"] = 3;
119     $input["password"] = bcrypt($input['dni']);
120     User::create($input);
121     Alert::success('Usuario agregado', 'Registro guardado correctamente')->autoclose(5000);
122     return redirect(route('user.medico.index'));
123 }

```

```

124 public function editPaciente($id)
125 {
126     $user = User::find($id);
127     return view('paciente.edit', compact('user'));
128 }
129
130 public function editMedico($id)
131 {
132     $user = User::find($id);
133     $lista_especialidad = Especialidad::all()->pluck('nombre', 'id')->prepend('seleccionar', '');
134     return view('medico.edit', compact('user', 'lista_especialidad'));
135 }
136
137 public function updatePaciente($id, Request $request)
138 {
139     $input = $request->all();
140     if (empty($input['password'])) {
141         $input['password'] = bcrypt($input['password']);
142     }else{
143         unset($input['password']);
144     }
145
146     unset($input['_method']);
147     unset($input['_token']);
148     User::where('id', $id)->update($input);
149     Alert::success('Usuario actualizado', 'Registro actualizado correctamente')->autoclose(5000);
150     return redirect(route('user.paciente.index'));
151 }
152
153 public function updateMedico($id, Request $request)
154 {
155     $input = $request->all();
156     if (empty($input['password'])) {
157         $input['password'] = bcrypt($input['password']);
158     }else{
159         unset($input['password']);
160     }
161
162     unset($input['_method']);
163     unset($input['_token']);
164     User::where('id', $id)->update($input);
165     Alert::success('Usuario actualizado', 'Registro actualizado correctamente')->autoclose(5000);
166     return redirect(route('user.medico.index'));
167 }
168
169 public function eliminar($id)
170 {
171     $result = true;
172     $message = null;
173     $enfermedad_tratamiento = EnfermedadTratamiento::where('id_usuario', $id)->first();
174     if($enfermedad_tratamiento){
175         $result = false;
176         $message = "Usuario en uso";
177     }else{
178         User::where('id', $id)->delete();
179         $message = "Usuario eliminado correctamente";
180     }
181     return response()->json(
182         array(
183             'result' => $result,
184             'message' => $message
185         )
186     );
187 }
188 }
189 }

```

Figura\_ 42: Modelo de Gestión de médicos – Lista

```

1 k?php
2
3 namespace App\Models;
4
5 use Illuminate\Contracts\Auth\MustVerifyEmail;
6 use Illuminate\Database\Eloquent\SoftDeletes;
7 use Illuminate\Database\Eloquent\Factories\HasFactory;
8 use Illuminate\Foundation\Auth\User as Authenticatable;
9 use Illuminate\Notifications\Notifiable;
10
11 class User extends Authenticatable
12 {
13     use HasFactory, Notifiable, SoftDeletes;
14
15     protected $dates = ['deleted_at'];
16
17     protected $fillable = [
18         'id_perfil',
19         'id_especialidad',
20         'nombres',
21         'apellidos',
22         'dni',
23         'telefono',
24         'email',
25         'password',
26         'fecha_nacimiento',
27         'talla',
28         'peso'
29     ];
30
31     function perfil(){
32         return $this->belongsTo('App\Models\Perfil', 'id_perfil', 'id');
33     }
34
35     function especialidad(){
36         return $this->belongsTo('App\Models\Especialidad', 'id_especialidad', 'id');
37     }
38 }
39 }
40 }

```

Figura\_ 43: Vista de Gestión de médicos – Lista

```

indexblade.php
1 @extends('layouts.app')
2 @section('content')
3 @include('sweetalert::alert')
4 <div class="row mb-2 mb-xl-3">
5 <div class="col-auto d-none d-sm-block">
6 <h3><strong>Pacientes</strong></h3>
7 </div>
8
9 <div class="col-auto ms-auto text-end mt-n1">
10 <a href="{{ route('user.paciente.create') }}" class="btn btn-primary">Agregar Paciente</a>
11 </div>
12 </div>
13 <div class="card">
14 <div class="card-body">
15 <table id="pacientes" class="table table-striped text-center" style="width:100%">
16 <thead>
17 <tr>
18 <th>#</th>
19 <th>DNI</th>
20 <th>Nombres y Apellidos</th>
21 <th>Teléfono</th>
22 <th>Fecha de nacimiento</th>
23 <th class="no-export">Acciones</th>
24 </tr>
25 </thead>
26 </table>
27 </div>
28 </div>
29 @endsection
30
31 @section('js')
32 <script>
33
34 var table = $('#pacientes').DataTable( {
35     "language": {
36         "lengthMenu": "Ver los MENU Primeros Registros",
37         "info": " _END_ de TOTAL registros",
38         "infoEmpty": "No se encontraron registros",
39         "infoFiltered": "(Filtrado de _MAX_ total entradas)",
40         "loadingRecords": "Cargando...",
41         "processing": "Procesando...",
42         "sSearch": "BUSCAR:",
43         "sZeroRecords": "No se encontraron resultados",
44         "sEmptyTable": "Ningún dato disponible en esta tabla",
45         "oPaginate": {
46             "sFirst": "Primero",
47             "sLast": "Último",
48             "sNext": "Siguiente",
49             "sPrevious": "Anterior"
50         },
51         "fnInfoCallback": null
52     },
53     "aProcessing": true,
54     "asServerSide": true,
55     "destroy":true,
56     "ajax":{
57         "method":"GET",
58         "url": "{{ route('user.paciente.lista') }}"
59     },
60     "iDisplayLength":10,
61     dom: 'Bfrtip',
62     buttons: [
63         {
64             extend: 'excelHtml5',
65             title: 'Sistema experto - pacientes'
66         },
67         {
68             exportOptions: { columns: ":not(.no-export)" }
69         },
70         {
71             extend: 'pdfHtml5',
72             title: 'Sistema experto - pacientes',
73             orientation: 'landscape',
74             className: 'btn btn-danger',
75             text: '<i class="far fa-file-pdf"></i> PDF',
76             exportOptions: { columns: ":not(.no-export)" }
77         },
78     ],
79 });
80
81 function deleteRegister(id){
82     Swal.fire({
83         title: 'Esta seguro de eliminar este registro?',
84         text: "",
85         icon: 'warning',
86         showCancelButton: true,
87         confirmButtonColor: '#3085d6',
88         cancelButtonColor: '#d33',
89         confirmButtonText: 'Confirmar',
90         cancelButtonText: 'Cancelar'
91     }).then((result) => {
92         if (result.value) {
93             $.ajax({
94                 url: "{{URL::to('/user')}}" + "/" + id,
95                 type: 'DELETE'
96             })
97             .done(function(res) {
98                 if(res.result){
99                     table.ajax.reload();
100                     Swal.fire(res.message, "", "success");
101                 }else{
102                     Swal.fire(res.message, "", "error");
103                 }
104             });
105         }
106     })
107 }
108 </script>
109 @endsection
110

```

## IMPLEMENTACIÓN

En la siguiente figura se presenta la interfaz de este requerimiento, la cual fue definida anteriormente para proceder con el desarrollo.

**Figura\_ 44: Implementación de Gestión de médicos – Lista**

#	DNI	Nombres y Apellidos	Teléfono	Especialidad	Acciones
1	65478213	Alexander Perez	987425312	ODCONTOLOGIA	[Iconos de acciones]
2	12345678	Luis Guillermo	954752818	PEDIATRIA	[Iconos de acciones]

**RF6: El sistema permitirá registrar los datos de cada médico, DNI, nombres, apellidos, teléfono, correo, fecha de nacimiento y su especialidad.**

## DISEÑO

### PROTOTIPO

De la misma manera a continuación se muestra el prototipo desarrollado para este requerimiento, la cual fue presentado mediante el producto owner para poder continuar con la parte del desarrollo, el programa en el que se desarrollo fue el Balsamiq mockups.

Figura\_ 45: Prototipo Gestión de médicos – Registro

El prototipo muestra una interfaz de usuario para el registro de médicos. En la parte superior, hay una barra de título que dice "Sistema experto" y una barra de búsqueda. A la izquierda, un menú de navegación muestra el perfil de "Gabriel Lozada Médico" y opciones como Inicio, Pacientes, Médicos, Síntomas, Enfermedades, Diagnóstico y Tratamiento. El área principal, titulada "Médicos", contiene campos de entrada para DNI, Nombres, Apellidos, Teléfono, Correo, Fecha N. (con un icono de calendario) y Especialidad (un menú desplegable). Al final, hay botones para "Cancelar" y "Guardar".

## DESARROLLO

En las siguientes figuras se pueden observar la parte del desarrollo del sistema, la cual va comenzando por el controlador, modelo y vista.

Figura\_ 46: Controlador de Gestión de médicos – Registro

```
UserController.php x
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use App\Models\User;
7 use App\Models\EnfermedadTratamiento;
8 use App\Models\Especialidad;
9 use Illuminate\Support\Facades\Auth;
10 use Alert;
11
12 class UserController extends Controller
13 {
14     public function indexPaciente()
15     {
16         return view('paciente.index');
17     }
18
19     public function indexMedico()
20     {
21         return view('medico.index');
22     }
23
24     public function listaPaciente()
25     {
26         $users = User::where('id_perfil',2)->orderByDesc('created_at')->get();
27
28         $data = collect();
29         foreach ($users as $key => $item) {
30             $route_edit = route('user.paciente.edit',$item->id);
31             $acciones="";
32             if($item->id != 1){
33                 $acciones .= '<a href="'.$route_edit.'" class="btn btn-warning" data-toggle="tooltip"
34                     title="Editar"><i class="fas fa-pen"></i></a>
35                 <button type="button" class="btn btn-danger" data-toggle="tooltip" title="Eliminar"
36                     onclick="deleteRegister('.$item->id.')"><i class="fas fa-trash-alt"></i></button>
37                 ';
38             }else{
39                 $acciones .= '<a href="'.$route_edit.'" class="btn btn-warning" data-toggle="tooltip"
40                     title="Editar"><i class="fas fa-pen"></i></a>';
41             }
42
43             $data_users = array(
44                 ($key+1),
45                 $item->dni,
46                 $item->nombres." ".$item->apellidos,
47                 $item->telefono,
48                 date('d-m-Y',strtotime($item->fecha_nacimiento)),
49                 $acciones
50             );
51             $data->push($data_users);
52         }
53         $results = array(
54             "draw" => 0,
55             "recordsTotal" => count($data),
56             "recordsFiltered" => count($data),
57             "data" => $data
58         );
59         return response()->json($results);
60     }
61 }
```



```

UserController.php x
58 public function listaMedico()
59 {
60     $users = User::where('id_perfil',3)->orderByDesc('created_at')->get();
61
62     $data = collect();
63     foreach ($users as $key => $item) {
64         $route_edit = route('user.medico.edit',$item->id);
65         $acciones = "";
66         if($item->id != 1){
67             $acciones .= '<a href="'.$route_edit.'" class="btn btn-warning" data-toggle="tooltip"
68                 title="Editar"><i class="fas fa-pen"></i></a>
69             <button type="button" class="btn btn-danger" data-toggle="tooltip" title="Eliminar"
70                 onclick="deleteRegister('.$item->id.')"><i class="fas fa-trash-alt"></i></button>
71         }else{
72             $acciones .= '<a href="'.$route_edit.'" class="btn btn-warning" data-toggle="tooltip"
73                 title="Editar"><i class="fas fa-pen"></i></a>';
74         }
75         $data_users = array(
76             ($key),
77             $item->dni,
78             $item->nombres. " " . $item->apellidos,
79             $item->telefono,
80             $item->especialidad->nombre,
81             $acciones
82         );
83         $data->push($data_users);
84     }
85     $results = array(
86         "draw" => 0,
87         "recordsTotal" => count($data),
88         "recordsFiltered" => count($data),
89         "data" => $data
90     );
91     return response()->json($results);
92 }
93
94 public function createPaciente()
95 {
96     return view('paciente.create');
97 }
98
99 public function createMedico()
100 {
101     $lista_especialidad = Especialidad::all()->pluck('nombre','id')->prepend('Seleccionar','');
102     return view('medico.create',compact('lista_especialidad'));
103 }
104
105 public function storePaciente(Request $request)
106 {
107     $input = $request->all();
108     $input["id_perfil"] = 2;
109     $input["password"] = bcrypt($input['dni']);
110     User::create($input);
111     Alert::success('Usuario agregado', 'Registro guardado correctamente')->autoclose(5000);
112     return redirect(route('user.paciente.index'));
113 }
114
115 public function storeMedico(Request $request)
116 {
117     $input = $request->all();
118     $input["id_perfil"] = 3;
119     $input["password"] = bcrypt($input['dni']);
120     User::create($input);
121     Alert::success('Usuario agregado', 'Registro guardado correctamente')->autoclose(5000);
122     return redirect(route('user.medico.index'));
123 }

```

```

124 public function editPaciente($id)
125 {
126     $user = User::find($id);
127     return view('paciente.edit', compact('user'));
128 }
129
130 public function editMedico($id)
131 {
132     $user = User::find($id);
133     $lista_especialidad = Especialidad::all()->pluck('nombre', 'id')->prepend('seleccionar', '');
134     return view('medico.edit', compact('user', 'lista_especialidad'));
135 }
136
137 public function updatePaciente($id, Request $request)
138 {
139     $input = $request->all();
140     if (empty($input['password'])) {
141         $input['password'] = bcrypt($input['password']);
142     }else{
143         unset($input['password']);
144     }
145
146     unset($input['_method']);
147     unset($input['_token']);
148     User::where('id', $id)->update($input);
149     Alert::success('Usuario actualizado', 'Registro actualizado correctamente')->autoclose(5000);
150     return redirect(route('user.paciente.index'));
151 }
152
153 public function updateMedico($id, Request $request)
154 {
155     $input = $request->all();
156     if (empty($input['password'])) {
157         $input['password'] = bcrypt($input['password']);
158     }else{
159         unset($input['password']);
160     }
161
162     unset($input['_method']);
163     unset($input['_token']);
164     User::where('id', $id)->update($input);
165     Alert::success('Usuario actualizado', 'Registro actualizado correctamente')->autoclose(5000);
166     return redirect(route('user.medico.index'));
167 }
168
169 public function eliminar($id)
170 {
171     $result = true;
172     $message = null;
173     $enfermedad_tratamiento = EnfermedadTratamiento::where('id_usuario', $id)->first();
174     if($enfermedad_tratamiento){
175         $result = false;
176         $message = "Usuario en uso";
177     }else{
178         User::where('id', $id)->delete();
179         $message = "Usuario eliminado correctamente";
180     }
181     return response()->json(
182         array(
183             'result' => $result,
184             'message' => $message
185         )
186     );
187 }
188 }
189 }

```

Figura\_ 47: Modelo de Gestión de médicos – Registro

```

1 k?php
2
3 namespace App\Models;
4
5 use Illuminate\Contracts\Auth\MustVerifyEmail;
6 use Illuminate\Database\Eloquent\SoftDeletes;
7 use Illuminate\Database\Eloquent\Factories\HasFactory;
8 use Illuminate\Foundation\Auth\User as Authenticatable;
9 use Illuminate\Notifications\Notifiable;
10
11 class User extends Authenticatable
12 {
13     use HasFactory, Notifiable, SoftDeletes;
14
15     protected $dates = ['deleted_at'];
16
17     protected $fillable = [
18         'id_perfil',
19         'id_especialidad',
20         'nombres',
21         'apellidos',
22         'dni',
23         'telefono',
24         'email',
25         'password',
26         'fecha_nacimiento',
27         'talla',
28         'peso'
29     ];
30
31     function perfil(){
32         return $this->belongsTo('App\Models\Perfil', 'id_perfil', 'id');
33     }
34
35     function especialidad(){
36         return $this->belongsTo('App\Models\Especialidad', 'id_especialidad', 'id');
37     }
38 }
39 }
40 }

```

Figura\_ 48: Vista de Gestión de médicos – Registro

```

indexblade.php
1 @extends('layouts.app')
2 @section('content')
3 @include('sweetalert::alert')
4 <div class="row mb-2 mb-xl-3">
5 <div class="col-auto d-none d-sm-block">
6 <h3><strong>Pacientes</strong></h3>
7 </div>
8
9 <div class="col-auto ms-auto text-end mt-n1">
10 <a href="{{ route('user.paciente.create') }}" class="btn btn-primary">Agregar Paciente</a>
11 </div>
12 </div>
13 <div class="card">
14 <div class="card-body">
15 <table id="pacientes" class="table table-striped text-center" style="width:100%">
16 <thead>
17 <tr>
18 <th>#</th>
19 <th>DNI</th>
20 <th>Nombres y Apellidos</th>
21 <th>Teléfono</th>
22 <th>Fecha de nacimiento</th>
23 <th class="no-export">Acciones</th>
24 </tr>
25 </thead>
26 </table>
27 </div>
28 </div>
29 @endsection
30
31 @section('js')
32 <script>
33
34 var table = $('#pacientes').DataTable( {
35     "language": {
36         "lengthMenu": "Ver los MENU Primeros Registros",
37         "info": " _END_ de TOTAL registros",
38         "infoEmpty": "No se encontraron registros",
39         "infoFiltered": "(Filtrado de _MAX_ total entradas)",
40         "loadingRecords": "Cargando...",
41         "processing": "Procesando...",
42         "sSearch": "BUSCAR:",
43         "sZeroRecords": "No se encontraron resultados",
44         "sEmptyTable": "Ningún dato disponible en esta tabla",
45         "oPaginate": {
46             "sFirst": "Primero",
47             "sLast": "Último",
48             "sNext": "Siguiente",
49             "sPrevious": "Anterior"
50         },
51         "fnInfoCallback": null
52     },
53     "aProcessing": true,
54     "asServerSide": true,
55     "destroy":true,
56     "ajax":{
57         "method":"GET",
58         "url": "{{ route('user.paciente.lista') }}"
59     },
60     "iDisplayLength":10,
61     dom: 'Bfrtip',
62     buttons: [
63         {
64             extend: 'excelHtml5',
65             title: 'Sistema experto - pacientes'
66         },
67         {
68             exportOptions: { columns: ":not(.no-export)" }
69         },
70         {
71             extend: 'pdfHtml5',
72             title: 'Sistema experto - pacientes',
73             orientation: 'landscape',
74             className: 'btn btn-danger',
75             text: '<i class="far fa-file-pdf"></i> PDF',
76             exportOptions: { columns: ":not(.no-export)" }
77         },
78     ],
79 });
80
81 function deleteRegister(id){
82     Swal.fire({
83         title: 'Esta seguro de eliminar este registro?',
84         text: "",
85         icon: 'warning',
86         showCancelButton: true,
87         confirmButtonColor: '#3085d6',
88         cancelButtonColor: '#d33',
89         confirmButtonText: 'Confirmar',
90         cancelButtonText: 'Cancelar'
91     }).then((result) => {
92         if (result.value) {
93             $.ajax({
94                 url: "{{URL::to('/user')}}"/ + "/" + id,
95                 type: 'DELETE'
96             })
97             .done(function(res) {
98                 if(res.result){
99                     table.ajax.reload();
100                     Swal.fire(res.message, "", "success");
101                 }else{
102                     Swal.fire(res.message, "", "error");
103                 }
104             });
105         }
106     })
107 }
108 </script>
109 @endsection

```

## IMPLEMENTACIÓN

En la siguiente figura se presenta la interfaz de este requerimiento, la cual fue definida anteriormente para proceder con el desarrollo.

**Figura\_ 49: Implementación de Gestión de médicos – Registro**

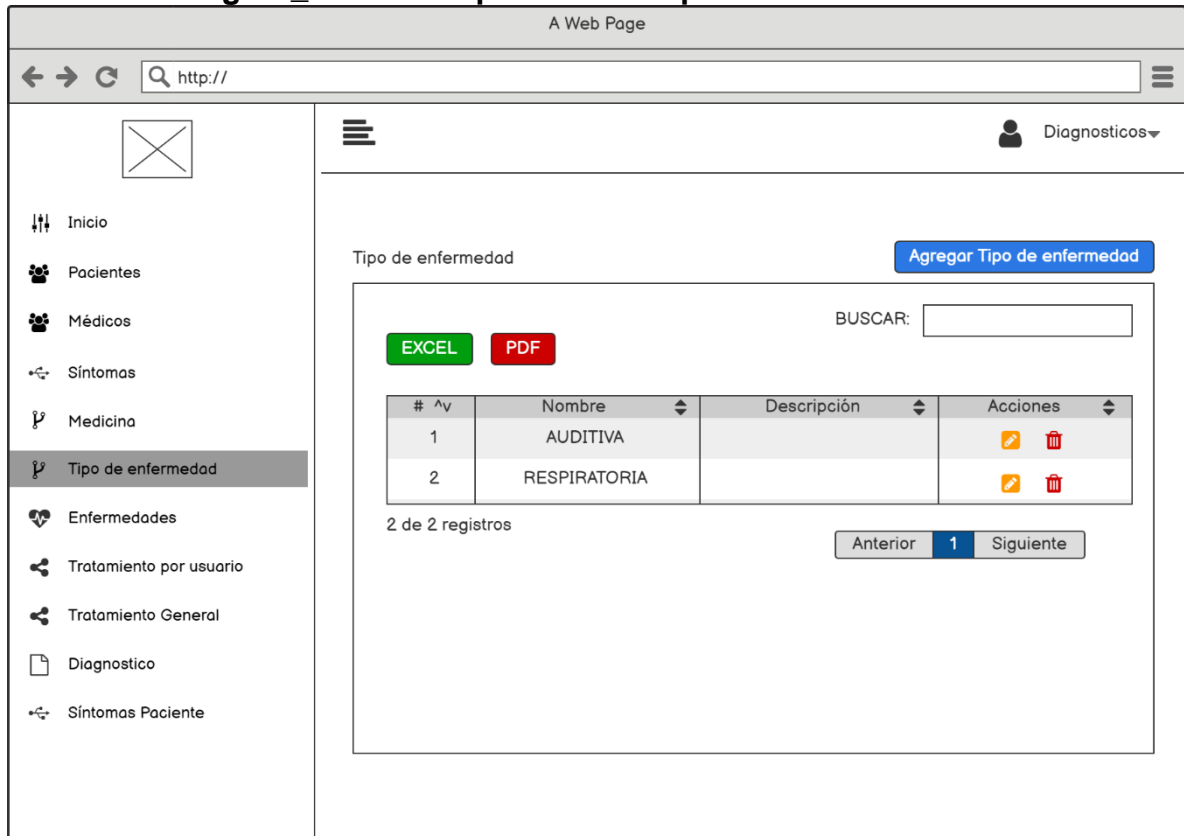
The image shows a web application interface for registering a new doctor. On the left, there is a dark sidebar menu with the following items: Inicio, Pacientes, Médicos (highlighted), Síntomas, Medicina, Tipo de enfermedad, Enfermedades, Tratamiento por usuario, Tratamiento General, Diagnóstico, and Síntomas Paciente. The main content area is titled 'Nuevo Médico' and contains a form with the following fields: DNI, Nombres, Apellidos, Teléfono, Correo electrónico, Fecha de Nacimiento (with a date picker), and Especialidad (with a dropdown menu). There are 'Cancelar' and 'Guardar' buttons at the bottom of the form. The footer of the page reads 'Sistema Experto ©'.

**RF7: El sistema permitirá gestionar el tipo de enfermedad, el nombre y la descripción.**

## **DISEÑO PROTOTIPO**

De la misma manera a continuación se muestra el prototipo desarrollado para este requerimiento, la cual fue presentado mediante el producto owner para poder continuar con la parte del desarrollo, el programa en el que se desarrollo fue el Balsamiq mockups.

**Figura\_ 50: Prototipo Gestión Tipo de enfermedad**



## DESARROLLO

En las siguientes figuras se pueden observar la parte del desarrollo del sistema, la cual va comenzando por el controlador, modelo y vista.

Figura\_ 51: Controlador de Gestión Tipo de enfermedad

```

1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6  use App\Models\TipoEnfermedad;
7  use App\Models\Urgonostico;
8  use App\Models\Enfermedad;
9  use App\Models\Sintomas;
10 use App\Models\UsuarioUrgonostico;
11 use Alert;
12
13 class TipoEnfermedadController extends Controller
14 {
15     public function index()
16     {
17         return view('tipo-enfermedad.index');
18     }
19
20     public function lista(Request $request)
21     {
22         $tipo_enfermedad = TipoEnfermedad::orderBy('nombre','asc')->get();
23
24         $data = collect();
25         foreach ($tipo_enfermedad as $key => $item) {
26             $route_edit = route('tipo-enfermedad.edit',$item->id);
27             $acciones = '<a href="'.$route_edit.'" class="btn btn-warning" data-toggle="tooltip" title="editar"><i
28                 class="fas fa-pen"></i></a>
29             <button type="button" class="btn btn-danger" data-toggle="tooltip" title="eliminar" onclick="deleteRegister(
30                 '.$item->id.')"><i class="fas fa-trash-alt"></i></button>';
31
32             $data_tipo_enfermedad = array(
33                 $key => [
34                     'item' => $item,
35                     'acciones' => $acciones,
36                 ];
37             $data->push($data_tipo_enfermedad);
38         }
39         $results = array(
40             'draw' => $request->input('draw'),
41             'recordsTotal' => count($data),
42             'recordsFiltered' => count($data),
43             'data' => $data
44         );
45         return response()->json($results);
46     }
47
48     public function create()
49     {
50         return view('tipo-enfermedad.create');
51     }
52
53     public function store(Request $request)
54     {
55         TipoEnfermedad::create($request->all());
56         Alert::success('Tipo de enfermedad agregado', 'Registro guardado correctamente')->autoclose(5000);
57         return redirect(route('tipo-enfermedad.index'));
58     }
59
60     public function edit($id)
61     {
62         $tipo_enfermedad = TipoEnfermedad::find($id);
63         if($tipo_enfermedad){
64             return view('tipo-enfermedad.edit',compact('tipo_enfermedad'));
65         }else{
66             abort(404);
67         }
68     }
69

```

```

70
71     public function update($id, Request $request)
72     {
73         $input = $request->all();
74
75         unset($input['_method']);
76         unset($input['_token']);
77
78         TipoEnfermedad::where('id',$id)->update($input);
79         Alert::success('Tipo de enfermedad actualizado', 'Registro actualizado correctamente')->autoclose(5000);
80         return redirect(route('tipo-enfermedad.index'));
81     }
82
83     public function destroy($id)
84     {
85         $result = true;
86         $message = '';
87         $diagnostico = Diagnostico::where('id_tipo_enfermedad',$id)->first();
88         $enfermedad = Enfermedad::where('id_tipo_enfermedad',$id)->first();
89         $sintomas = Sintomas::where('id_tipo_enfermedad',$id)->first();
90         $usuario_diagnostico = UsuarioDiagnostico::where('id_tipo_enfermedad',$id)->first();
91         if($diagnostico || $enfermedad || $sintomas || $usuario_diagnostico){
92             $result = false;
93             $message = "Tipo de enfermedad en uso";
94         }else{
95             TipoEnfermedad::where('id',$id)->delete();
96             $message = "Tipo de enfermedad eliminado correctamente";
97         }
98         return response()->json(
99             array(
100                 'result' => $result,
101                 'message' => $message
102             )
103         );
104     }
105 }

```

Figura\_ 52: Modelo de Gestión Tipo de enfermedad

```

1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7
8 class TipoEnfermedad extends Model
9 {
10     use HasFactory;
11     public $table = 'tipo_enfermedad';
12     public $timestamps = false;
13
14     public $fillable = [
15         'nombre',
16         'descripcion'
17     ];
18 }
19

```

Figura\_ 53: Vista de Gestión Tipo de enfermedad

```

1 @extends('layouts.app')
2 @section('content')
3 @include('sweetalert:alert')
4 <div class="row no-gutters">
5     <div class="col-auto d-none d-sm-block">
6         <a href="#">Tipo de enfermedad</a>
7     </div>
8
9     <div class="col-auto m-auto text-end mt-n1">
10         <a href="{{ route('tipo-enfermedad.create') }}" class="btn btn-primary">Agregar Tipo de enfermedad</a>
11     </div>
12 </div>
13 <div class="card">
14     <div class="card-body">
15         <table id="tipo-enfermedad" class="table table-striped text-center" style="width:100%">
16             <thead>
17                 <tr>
18                     <th>#</th>
19                     <th>Nombre</th>
20                     <th>Descripcion</th>
21                     <th class="no-export">Acciones</th>
22                 </tr>
23             </thead>
24             <tbody>
25                 <tr>
26                     <td></td>
27                 </tr>
28             </tbody>
29         </table>
30     </div>
31 @endsection
32 @section('js')
33 <script>
34     var table = $('#tipo-enfermedad').dataTable( {
35         "language": {
36             "lengthMenu": "Ver los MENU primeros Registros",
37             "info": "Página de TOTAL registros",
38             "infoEmpty": "No se encontraron registros",
39             "infoFiltered": "(filtrado de TOTAL entradas)",
40             "loadingRecords": "Cargando...",
41             "processing": "Procesando...",
42             "search": "BUSCAR:",
43             "zeroRecords": "No se encontraron resultados",
44             "emptyTable": "Ningun dato disponible en esta tabla",
45             "paginate": {
46                 "first": "Primero",
47                 "last": "Ultimo",
48                 "next": "Siguiente",
49                 "previous": "Anterior"
50             },
51             "infoCallback": null
52         },
53         "processing": true,
54         "serverSide": true,
55         "destroy": true,
56         "ajax": {
57             "url": "{{ route('tipo-enfermedad.lista') }}"
58         },
59         "displayLength": 10,
60         "dom": "Brtip",
61         "buttons": [
62             {
63                 extend: 'excelHtml5',
64                 title: 'sistema experto - Tipo de enfermedad',
65                 className: 'btn btn-success',
66                 text: '<i class="fa fa-file-excel"></i> EXCEL',
67                 exportOptions: { columns: "not(.no-export)" }
68             }
69         ]
70     } );

```

```

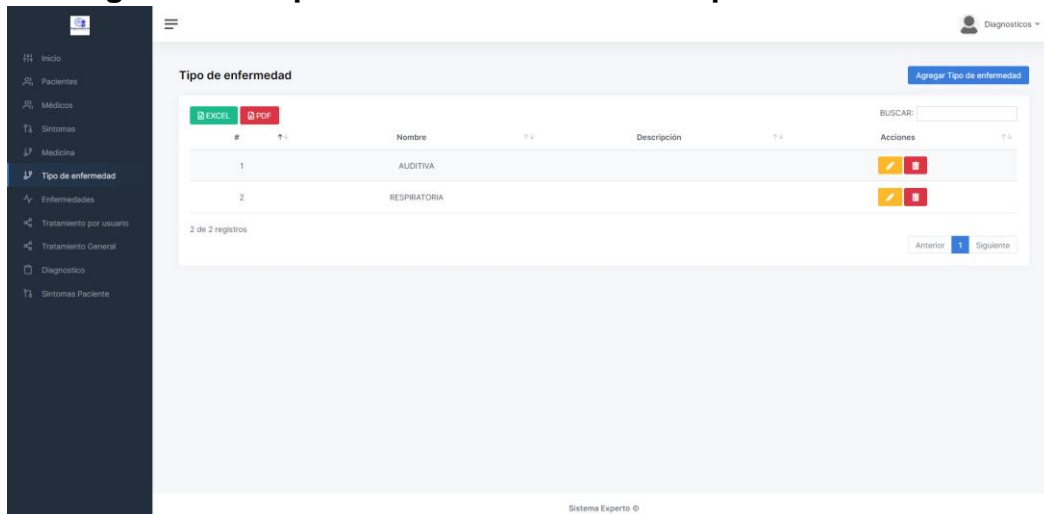
64     text: '❏ class="far fa-file-excel"></i> EXCEL',
65     exportoptions: { columns: ":(not(.no-export))" }
66   },
67   {
68     extend: 'pdfhtml5',
69     title: 'sistema experto - Tipo de enfermedad',
70     orientation: 'landscape',
71     className: 'btn btn-danger',
72     text: '❏ class="far fa-file-pdf"></i> PDF',
73     exportoptions: { columns: ":(not(.no-export))" }
74   },
75   ],
76   });
77
78   function deleteRegister(id){
79     Swal.fire({
80       title: 'Esta seguro de eliminar este registro?',
81       text: "",
82       icon: 'warning',
83       showCancelButton: true,
84       confirmButtonColor: '#3085d6',
85       cancelButtonColor: '#d33',
86       confirmButtonText: 'Confirmar',
87       cancelButtonText: 'Cancelar'
88     }).then((result) => {
89       if (result.value) {
90         $.ajax({
91           url: "{{URL::to('/tipo-enfermedad')}}" + "/" + id,
92           type: 'DELETE'
93         })
94         .done(function(res) {
95           if(res.result){
96             table.ajax.reload();
97             Swal.fire(res.message, "", "success");
98           }else{
99             Swal.fire(res.message, "", "error");
100           }
101         });
102       }
103     });
104   }
105 }
106 </script>
107 @endsection

```

## IMPLEMENTACIÓN

En la siguiente figura se presenta la interfaz de este requerimiento, la cual fue definida anteriormente para proceder con el desarrollo.

**Figura\_ 54: Implementación de Gestión Tipo de enfermedad**



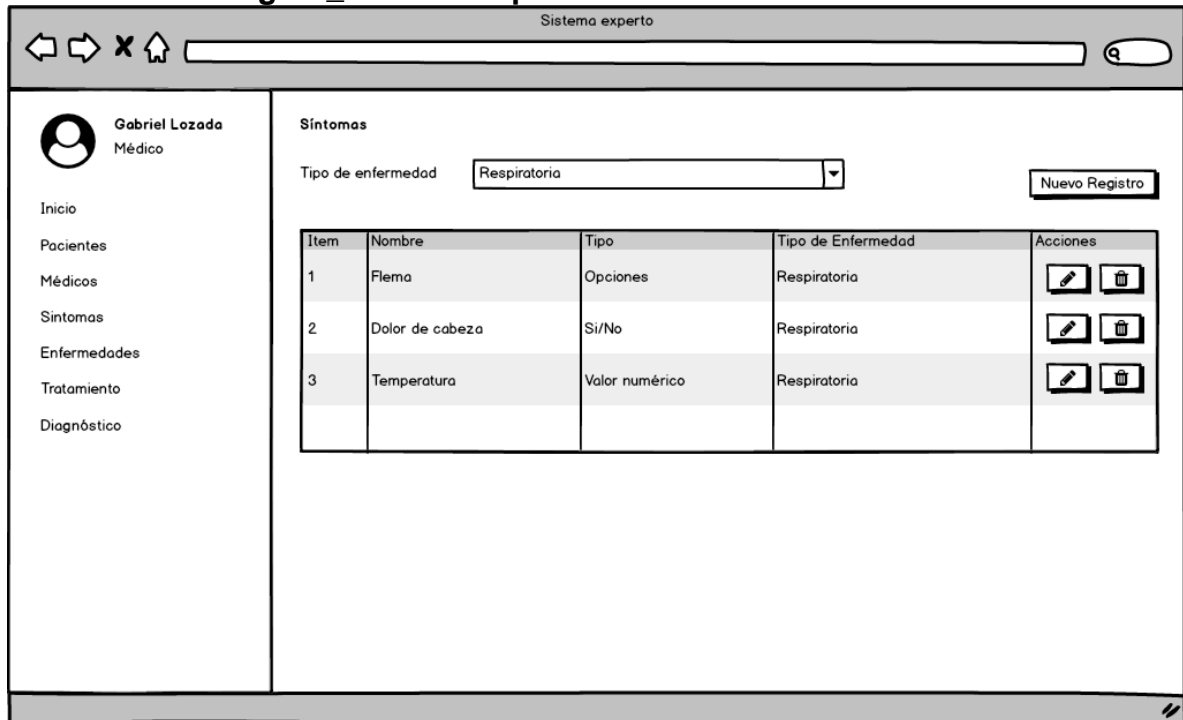
**RF8: El sistema web permitirá gestionar los síntomas donde se registrarán específicamente tres tipos de síntomas “opciones”, si/no y valor numérico.**

## DISEÑO PROTOTIPO



De la misma manera a continuación se muestra el prototipo desarrollado para este requerimiento, la cual fue presentado mediante el producto owner para poder continuar con la parte del desarrollo, el programa en el que se desarrollo fue el Balsamiq mockups.

**Figura\_ 55: Prototipo de Gestión de síntomas**



## DESARROLLO

En las siguientes figuras se pueden observar la parte del desarrollo del sistema, la cual va comenzando por el controlador, modelo y vista.

Figura\_ 56: Controlador de Gestión de síntomas

```
SintomasController.php x
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use App\Models\Sintomas;
7 use App\Models\TipoEnfermedad;
8 use App\Models\EnfermedadSintomas;
9 use Alert;
10
11 class SintomasController extends Controller
12 {
13     public function index()
14     {
15         $tipo_enfermedad = TipoEnfermedad::all()->pluck('nombre','id')->prepend('TOODS','');
16         return view('sintomas.index',compact('tipo_enfermedad'));
17     }
18
19     public function lista(Request $request)
20     {
21         if($request->tipo_enfermedad){
22             $sintomas = Sintomas::where('id_tipo_enfermedad',$request->tipo_enfermedad)
23                 ->orderBy('nombre','asc')
24                 ->get();
25         }else{
26             $sintomas = Sintomas::orderByDesc('created_at')->get();
27         }
28
29         $data = collect();
30         foreach ($sintomas as $key => $item) {
31             $route_edit = route('sintomas.edit',$item->id);
32             $acciones = '<a href="'.$route_edit.'" class="btn btn-warning" data-toggle="tooltip" title="Editar"><i
33                 class="fas fa-pen"></i></a>
34             <button type="button" class="btn btn-danger" data-toggle="tooltip" title="Eliminar" onclick="deleteRegister(
35                 '$item->id.')"><i class="fas fa-trash-alt"></i></button>';
36
37             $data_sintomas = array(
38                 ($key->id),
39                 $item->nombre,
40                 $item->setTipo($item->tipo),
41                 $item->tipoEnfermedad->nombre,
42                 $acciones
43             );
44             $data->push($data_sintomas);
45         }
46         $results = array(
47             "draw" => 0,
48             "recordsTotal" => count($data),
49             "recordsFiltered" => count($data),
50             "data" => $data
51         );
52         return response()->json($results);
53     }
54
55     public function create()
56     {
57         $tipo_enfermedad = TipoEnfermedad::all()->pluck('nombre','id')->prepend('Seleccionar','');
58         return view('sintomas.create',compact('tipo_enfermedad'));
59     }
60
61     public function store(Request $request)
62     {
63         Sintomas::create($request->all());
64         Alert::success('Síntoma agregado', 'Registro guardado correctamente')->autoclose(5000);
65         return redirect(route('sintomas.index'));
66     }
67 }
```

```
SintomasController.php x
67
68     public function edit($id)
69     {
70         $sintomas = Sintomas::find($id);
71         if($sintomas){
72             $tipo_enfermedad = TipoEnfermedad::all()->pluck('nombre','id')->prepend('Seleccionar','');
73             return view('sintomas.edit',compact('sintomas','tipo_enfermedad'));
74         }else{
75             abort(404);
76         }
77     }
78
79     public function update($id, Request $request)
80     {
81         $input = $request->all();
82         unset($input['_method']);
83         unset($input['_token']);
84
85         if($input["tipo"] != 1){
86             $input["items"] = null;
87         }
88
89         Sintomas::where('id',$id)->update($input);
90         Alert::success('Síntoma actualizado', 'Registro actualizado correctamente')->autoclose(5000);
91         return redirect(route('sintomas.index'));
92     }
93
94     public function destroy($id)
95     {
96         $result = true;
97         $message = null;
98         $enfermedad_sintomas = EnfermedadSintomas::where('id_sintoma',$id)->first();
99         if($enfermedad_sintomas){
100             $result = false;
101             $message = "Síntoma en uso";
102         }else{
103             Sintomas::where('id',$id)->delete();
104             $message = "Síntoma eliminado correctamente";
105         }
106         return response()->json(
107             array(
108                 'result' => $result,
109                 'message' => $message
110             )
111         );
112     }
113
114 }
```

Figura\_ 57: Modelo de Gestión de síntomas

```

Sintomas.php
1 |<?php
2
3 | namespace App\Models;
4
5 | use Illuminate\Database\Eloquent\Factories\HasFactory;
6 | use Illuminate\Database\Eloquent\Model;
7 | use Illuminate\Database\Eloquent\SoftDeletes;
8
9 | class Sintomas extends Model
10 | {
11 |     use HasFactory;
12 |     use SoftDeletes;
13 |     public $table = 'sintomas';
14 |     protected $dates = ['deleted_at'];
15
16 |     public $fillable = [
17 |         'id_tipo_enfermedad',
18 |         'tipo',
19 |         'nombre',
20 |         'items'
21 |     ];
22
23 |     function tipoEnfermedad(){
24 |         return $this->belongsTo('App\Models\TipoEnfermedad', 'id_tipo_enfermedad', 'id');
25 |     }
26
27 |     function setTipo($tipo){
28 |         switch ($tipo) {
29 |             case '1':
30 |                 return "Opciones";
31 |                 break;
32 |             case '2':
33 |                 return "Si/No";
34 |                 break;
35 |             case '3':
36 |                 return "Valor Numérico";
37 |                 break;
38 |             default:
39 |                 break;
40 |         }
41 |     }
42 | }
43

```

Figura\_ 58: Vista de Gestión de síntomas

```

index.blade.php
1 | @extends('layouts.app')
2 | @section('content')
3 | @include('sweetalert:alert')
4 | <div class="row mb-2 mb-n1">
5 |     <div class="col-auto d-none d-sm-block">
6 |         <strong>Sintomas</strong></div>
7 |
8 |
9 |     <div class="col-auto ms-auto text-end mt-n1">
10 |         @if (isset($errors['sintomas.create'])) <button type="button" class="btn btn-primary">Agregar Síntoma</button>
11 |         @endif
12 |     </div>
13 | </div>
14 | <div class="card">
15 |     <div class="card-body">
16 |         <div class="row">
17 |             <div class="col-form-label col-sm-2">Tipo de Enfermedad</div>
18 |             <div class="col-sm-3 mb-4">
19 |                 {!! Form::select('tipo', $tipos, null, ['class' => 'form-select', 'id' => 'tipo']) !!}
20 |             </div>
21 |         </div>
22 |         <table id="sintomas" class="table table-striped text-center" style="width:100%">
23 |             <thead>
24 |                 <tr>
25 |                     <th>#</th>
26 |                     <th>Nombre</th>
27 |                     <th>Tipo</th>
28 |                     <th>Tipo de Enfermedad</th>
29 |                     <th class="no-export">Acciones</th>
30 |                 </tr>
31 |             </thead>
32 |             <tbody>
33 |             </tbody>
34 |         </table>
35 |     </div>
36 | @endsection
37 | @section('js')
38 | <script>
39 |     listar();
40 |
41 |     $('#tipo').change(function() {
42 |         listar();
43 |     });
44 |
45 |     function listar(){
46 |         var tipo_enfermedad = $('#tipo').val();
47 |
48 |         $('#sintomas').dataTable( {
49 |             "language": {
50 |                 "lengthMenu": "Ver los MENU Primeros Registros",
51 |                 "info": " _END_ de TOTAL registros",
52 |                 "infoEmpty": "No se encontraron registros",
53 |                 "infoFiltered": "(Filtrado de _MAX_ total entradas)",
54 |                 "loadingRecords": "Cargando...",
55 |                 "processing": "Procesando...",
56 |                 "search": "BUSCAR:",
57 |                 "zeroRecords": "No se encontraron resultados",
58 |                 "emptyTable": "Ningún dato disponible en esta tabla",
59 |                 "oPaginate": {
60 |                     "sFirst": "Primero",
61 |                     "sLast": "Último",
62 |                     "sNext": "Siguiente",
63 |                     "sPrevious": "Anterior"
64 |                 },
65 |                 "fnInfoCallback": null
66 |             },
67 |             "processing": true,
68 |             "serverSide": true,
69 |             "destroy": true,
70 |

```

```

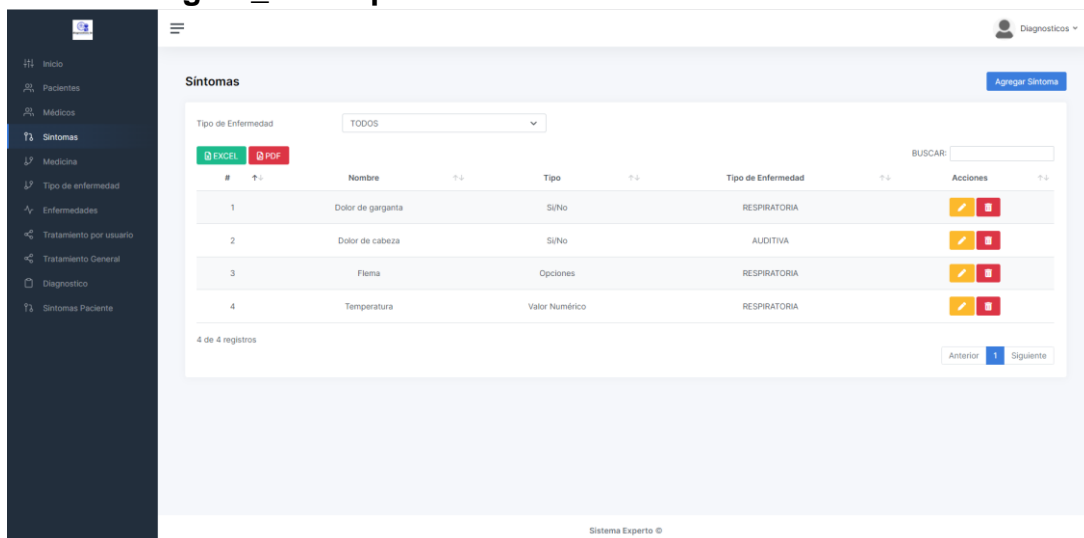
69     "aServerSide": true,
70     "destroy": true,
71     "ajax": {
72         "method": "POST",
73         "url": "{{ route('sintomas.lista') }}",
74         "data": {tipo_enfermedad: tipo_enfermedad}
75     },
76     "idDisplayLength": 10,
77     "dom": 'BfRtip',
78     "buttons": [
79         {
80             extend: 'excelHtml5',
81             title: 'Sistema experto - Sintomas',
82             className: 'btn btn-success',
83             text: '<i class="fa fa-file-excel"></i> EXCEL',
84             exportoptions: { columns: ":not(.no-export)" }
85         },
86         {
87             extend: 'pdfHtml5',
88             title: 'Sistema experto - Sintomas',
89             orientation: 'landscape',
90             className: 'btn btn-danger',
91             text: '<i class="fa fa-file-pdf"></i> PDF',
92             exportoptions: { columns: ":not(.no-export)" }
93         },
94     ],
95 });
96 }
97 }
98
99 function deleteRegister(id){
100     Swal.fire({
101         title: 'Esta seguro de eliminar este registro?',
102         text: '',
103         icon: 'warning',
104         showCancelButton: true,
105         confirmButtonColor: '#3085d6',
106         cancelButtonColor: '#d33',
107         confirmButtonText: 'Confirmar',
108         cancelButtonText: 'Cancelar'
109     }).then((result) => {
110         if (result.value) {
111             $.ajax({
112                 url: "{{URL::to('/sintomas')}}" + "/" + id,
113                 type: 'DELETE'
114             })
115             .done(function(res) {
116                 if(res.result){
117                     listar();
118                     Swal.fire(res.message, "", "success");
119                 }else{
120                     Swal.fire(res.message, "", "error");
121                 }
122             });
123         }
124     })
125 }
126 }
127 </script>
128 @endsection

```

## IMPLEMENTACIÓN

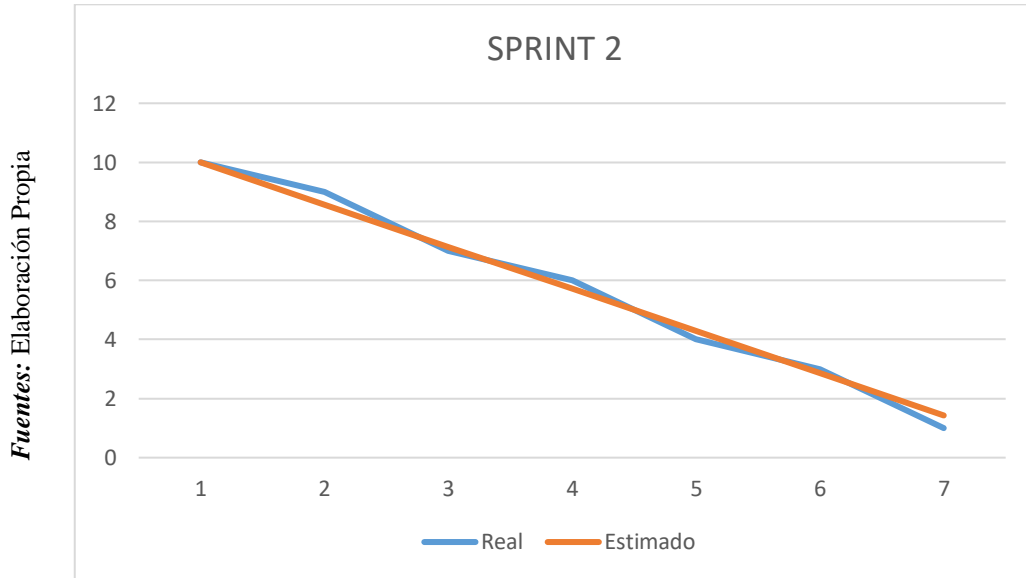
En la siguiente figura se presenta la interfaz de este requerimiento, la cual fue definida anteriormente para proceder con el desarrollo.

**Figura\_ 59: Implementación de Gestión de síntomas**



## BURNDOWN DEL SPRINT N° 2

Figura\_ 60: Burndown del Sprint 2



### Burndown Sprint 2

A continuación, se presentará el significado de la figura anterior: comenzando con la línea roja la cual representa el tiempo ideal del desarrollo de la misma, así mismo la línea azul representa el tiempo real en la que se desarrolló dicho sprint. Es así como se puede decir que si la línea azul se ubica por debajo de la línea roja es porque existió un adelanto en la elaboración del sprint, y de manera contraria si la línea azul está por encima de la roja significa retraso en la elaboración del sprint.

## ACTA DE REUNION DEL SPRINT N° 2

Siendo las 3 pm del día 16 de octubre del 2021, se reúne en el Policlínico Monserrat.

Presentes:

ROL	NOMBRE
Scrum Master	Luz Gomez Taype
Team Member	Gianmarco Roger Nizama Antunez
Product Owner	Miguel Rios

El encargado del Policlínico Monserrat, Luz Gomez Taype culmina la reunión con el Sr. Gianmarco Nizama señalando los últimos puntos respecto al sprint.

De esta manera se logra definir la forma de trabajo y todos los requerimientos respecto a este sprint, así mismo firmar el acta que contiene la fecha de entrega, esto para el cierre de la reunión.

Los asistentes presentes aprobaron los puntos que fueron acordados en esta reunión, así mismo beneficio la planificación de este Sprint, realizando el compromiso de entregar los resultados referentes al tiempo que fue pactado.


## ACTA DE ENTREGA DEL SPRINT N° 2

Siendo la 1 pm del día 2 de noviembre del 2021 se reúne en el Policlínico Monserrat.

ROL	NOMBRE
Scrum Master	Luz Gomez Taype
Team Member	Gianmarco Roger Nizama Antunez
Product Owner	Miguel Rios

El Sr. Gianmarco Nizama, señala y detalla cada requerimiento desarrollado, así mismo presenta las interfaces obtenidas del product owner.

Luego de realizar las verificaciones a la sustentación del Sr. Gianmarco Nizama para continuar con la aprobación del Sprint, de manera unánime se decide la aprobación de este sprint, referente al proyecto “Sistema experto para diagnóstico de enfermedades pulmonares en el policlínico Monserrat”.

Es así como los presentes imparten su aprobación al informe del Sr. Gianmarco Nizama, referente al Sprint N° 2 la cual fue concluida, del siguiente proyecto “Sistema experto para diagnóstico de enfermedades pulmonares en el policlínico Monserrat”.


## RESUMEN DE LA REUNIÓN RETROSPECTIVA DE SPRINT N° 2

Información de la empresa y proyecto:

<b>Empresa / Organización</b>	Policlínico Monserrat.
<b>Proyecto</b>	“Sistema experto para diagnóstico de enfermedades pulmonares en el policlínico Monserrat”

Información de la reunión:

<b>Lugar</b>	Policlínico Monserrat
<b>Fecha</b>	02/11/2021
<b>Número de iteración / Sprint</b>	Sprint 2
<b>Personas Convocadas a la reunión</b>	
<b>Persona que asistieron a la reunión</b>	

Formulario de reunión retrospectiva

<b>¿Qué salió bien en la Iteración? (Aciertos)</b>	<b>¿Qué no salió bien en la Iteración? (Errores)</b>
<p>Se ejecutó de manera correcta el registro de los médicos para tener luego una lista de todos ellos y poder gestionarlos.</p> <p>Se desarrolló correctamente la gestión del tipo de enfermedad.</p> <p>Se ejecutó la gestión de síntomas de cada paciente.</p>	<p>Existieron fallos referentes a la gestión de enfermedades, esto debido a que se tuvo que perfeccionar la selección de síntomas, sin embargo, se pudo solucionar.</p>

## PLANIFICACIÓN DEL SPRINT N°3



Siendo las 4 pm del día 2 de noviembre del 2021, se reúne en el Policlínico Monserrat.

Presentes:

<b>ROL</b>	<b>NOMBRE</b>
Scrum Master	Luz Gomez Taype
Team Member	Gianmarco Roger Nizama Antunez
Product Owner	Miguel Rios

El gerente del Policlínico Monserrat, ejecutó la presentación de todos los requerimientos obtenidos, de igual manera indica los que son de mayor prioridad.

Luego de ser analizada los requerimientos presentados por el gerente del Policlínico Monserrat. La señora Luz Gomez aclara ciertas dudas y da su compromiso para el cumplimiento de los requerimientos expuestos en el Sprint 3.

Así mismo los presentes imparten su aprobación respecto a lo expuesto esto específicamente en el Sprint 3, señalando que la fecha de entrega respecto a este Sprint será el 21 de noviembre del 2021.


## EJECUCIÓN DEL SPRINT 3

Anterior a la fase del diseño se necesita saber exactamente el funcionamiento del sistema, conocer y entender cada parte de ella, dicho de una manera diferente el análisis respectivo a lo que se requiere esto de acuerdo a las historias de usuario.

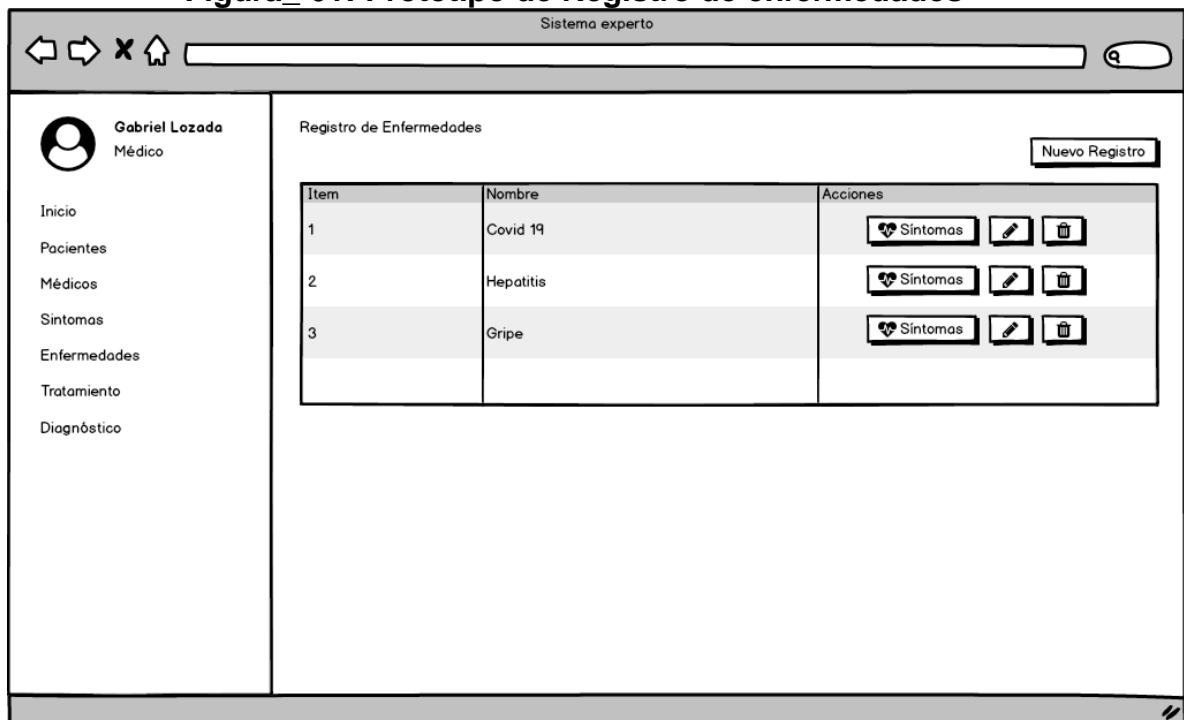
**RF9: El sistema web permitirá registrar enfermedades con sus respectivos síntomas en la cual cada síntoma tendrá un puntaje y en base a ello se medirá que enfermedades puede tener el paciente.**

## DISEÑO

### PROTOTIPO

De la misma manera a continuación se muestra el prototipo desarrollado para este requerimiento, la cual fue presentado mediante el producto owner para poder continuar con la parte del desarrollo, el programa en el que se desarrollo fue el Balsamiq mockups.

**Figura\_ 61: Prototipo de Registro de enfermedades**



Sistema experto

← → × ↶

🔍

**Gabriel Lozada**  
Médico

Inicio

Pacientes

Médicos

Sintomas

Enfermedades

Tratamiento

Diagnóstico

**Registro de Enfermedades**

Tipo enfermedad:

Nombre:

Descripción:

Puntaje de confirmación de enfermedad:  Si la evaluación es mayor o igual a esta, el paciente tendrá una probabilidad del 100% de tener esta enfermedad

## DESARROLLO

En las siguientes figuras se pueden observar la parte del desarrollo del sistema, la cual va comenzando por el controlador, modelo y vista.

Figura\_ 62: Controlador de Registro de enfermedades

```
1 |>php
2
3 | namespace App\Http\Controllers;
4
5 | use Illuminate\Http\Request;
6 | use App\Models\TipoEnfermedad;
7 | use App\Models\Enfermedad;
8 | use App\Models\EnfermedadSintomas;
9 | use App\Models\EnfermedadSintomasDetalle;
10 | use App\Models\EnfermedadTratamiento;
11 | use App\Models\DiagnosticoEnfermedad;
12 | use App\Models\Tratamiento;
13 | use App\Models\Sintomas;
14 | use Alert;
15
16 | class EnfermedadController extends Controller
17 | {
18 |     public function index()
19 |     {
20 |         return view('enfermedad.index');
21 |     }
22
23 |     public function lista(Request $request)
24 |     {
25 |         $enfermedad = Enfermedad::orderByDesc('created_at')->get();
26
27 |         $data = collect();
28 |         foreach ($enfermedad as $key => $item) {
29 |             $route_edit = route('enfermedad.edit', $item->id);
30 |             $route_sintomas = route('enfermedad.sintomas.index', $item->id);
31 |             $acciones = '<a href="' . $route_sintomas . '" class="btn btn-primary" data-toggle="tooltip" title="Sintomas">
32 |                 <i class="fas fa-heartbeat"></i>
33 |             </a>
34 |             <a href="' . $route_edit . '" class="btn btn-warning" data-toggle="tooltip" title="Editar"><i class="fas
35 |                 fa-pen"></i></a>
36 |             <button type="button" class="btn btn-danger" data-toggle="tooltip" title="Eliminar" onclick="deleteRegister(
37 |                 '. $item->id .')"><i class="fas fa-trash-alt"></i></button>;
38
39 |             $data_enfermedad = array(
40 |                 ($key+1),
41 |                 $item->nombre,
42 |                 $item->tipoEnfermedad->nombre,
43 |                 $item->puntaje,
44 |                 $acciones
45 |             );
46 |             $data->push($data_enfermedad);
47
48 |         }
49 |         $results = array(
50 |             "draw" => 0,
51 |             "recordsTotal" => count($data),
52 |             "recordsFiltered" => count($data),
53 |             "data" => $data
54 |         );
55 |         return response()->json($results);
56
57 |     public function create()
58 |     {
59 |         $tipo_enfermedad = TipoEnfermedad::all()->pluck('nombre', 'id')->prepend('seleccionar,');
60 |         return view('enfermedad.create', compact('tipo_enfermedad'));
61
62 |     public function store(Request $request)
63 |     {
64 |         Enfermedad::create($request->all());
65 |         Alert::success('Enfermedad agregado', 'Registro guardado correctamente')->autoclose(5000);
66 |         return redirect(route('enfermedad.index'));
67 |     }
68 | }
```

```

68
69
70
71     public function edit($id)
72     {
73         $enfermedad = Enfermedad::find($id);
74         if($enfermedad){
75             $tipo_enfermedad = TipoEnfermedad::all()->pluck('nombre','id')->prepend('Seleccionar,');
76             return view('enfermedad.edit',compact('tipo_enfermedad','enfermedad'));
77         }
78         abort(404);
79     }
80
81     public function update($id, Request $request)
82     {
83         $input = $request->all();
84
85         unset($input['_method']);
86         unset($input['_token']);
87
88         Enfermedad::where('id',$id)->update($input);
89         Alert::success('Enfermedad actualizado', 'Registro actualizado correctamente')->autoclose(5000);
90         return redirect(route('enfermedad.index'));
91     }
92
93     public function eliminar($id)
94     {
95         $result = true;
96         $message = null;
97         $diagnostico_enfermedad = DiagnosticoEnfermedad::where('id_enfermedad',$id)->first();
98         $enfermedad_tratamiento = EnfermedadTratamiento::where('id_enfermedad',$id)->first();
99         $tratamiento = Tratamiento::where('id_enfermedad',$id)->first();
100         if($diagnostico_enfermedad || $enfermedad_tratamiento || $tratamiento){
101             $result = false;
102             $message = "Enfermedad en uso";
103         }else{
104             Enfermedad::where('id',$id)->delete();
105             $message = "Enfermedad eliminado correctamente";
106         }
107         return response()->json(
108             array(
109                 'result' => $result,
110                 'message' => $message
111             )
112         );
113     }
114
115     public function sintomas($id)
116     {
117         $enfermedad = Enfermedad::find($id);
118         if($enfermedad){
119             return view('enfermedad.sintomas',compact('enfermedad'));
120         }else{
121             abort(404);
122         }
123     }
124
125     public function createSintoma($id)
126     {
127         $enfermedad = Enfermedad::find($id);
128         if($enfermedad){
129             $sintomas = Sintomas::where('id_tipo_enfermedad',$enfermedad->id_tipo_enfermedad)->pluck('nombre','id')->
130                 prepend('Seleccionar,');
131             return view('enfermedad.sintomas-create',compact('enfermedad','sintomas'));
132         }else{
133             abort(404);
134         }
135     }

```

```
134
135 public function consultarSintoma($id)
136 {
137     $sintomas = Sintomas::find($id);
138     return response()->json(
139         array(
140             'result' => true,
141             'data' => $sintomas
142         )
143     );
144 }
145
146 public function storesintoma($id ,Request $request)
147 {
148     $result = true;
149     $message = "";
150
151     $validar = EnfermedadSintomas::where('id_enfermedad',$id)
152     ->where('id_sintoma',$request->id_sintoma)
153     ->first();
154     if($validar){
155         $result = false;
156         $enfermedad = Enfermedad::find($id);
157         $message = "Síntoma " . $validar->sintomas->nombre. " ya está registrado para " . $enfermedad->nombre;
158     }else{
159         $enfermedad_sintomas = EnfermedadSintomas::create([
160             'id_enfermedad' => $id,
161             'id_sintoma' => $request->id_sintoma
162         ]);
163
164         foreach ($request->data_sintomas as $value) {
165             EnfermedadSintomasDetalle::create([
166                 'id_enfermedad_sintomas' => $enfermedad_sintomas->id,
167                 'nombre' => $value['nombre'],
168                 'puntaje' => $value['puntaje']
169             ]);
170         }
171
172         Alert::success('Síntoma agregado correctamente', '')->autoclose(5000);
173     }
174
175     return response()->json(
176         array(
177             'result' => $result,
178             'message' => $message
179         )
180     );
181 }
182
```

```

182
183 public function listasintomas($id)
184 {
185     $sintomas = EnfermedadSintomas::orderByDesc('id')
186     ->where('id_enfermedad',$id)
187     ->get();
188
189     $data = collect();
190     foreach ($sintomas as $key => $item) {
191         $route_edit = route('enfermedad.sintomas.edit',[$id,$item->id]);
192         $acciones='<a href="'.$route_edit.'" class="btn btn-warning" data-toggle="tooltip" title="Editar"><i
193             class="fas fa-pen"></i></a>
194         <button type="button" class="btn btn-danger" data-toggle="tooltip" title="Eliminar" onclick="deleteRegiste
195             '.$item->id.'"><i class="fas fa-trash-alt"></i></button>';
196
197         $data_sintomas = array(
198             ($key->),
199             $item->sintomas->nombre,
200             $item->sintomas->setTipo($item->sintomas->tipo),
201             $acciones
202         );
203         $data->push($data_sintomas);
204     }
205     $results = array(
206         "draw" => 0,
207         "recordsTotal" => count($data),
208         "recordsFiltered" => count($data),
209         "data" => $data
210     );
211     return response()->json($results);
212 }
213
214 public function editSintoma($id_enfermedad,$id_sintoma)
215 {
216     $enfermedad = Enfermedad::find($id_enfermedad);
217     if($enfermedad){
218         $sintomas = Sintomas::where('id_tipo_enfermedad',$enfermedad->id_tipo_enfermedad)->pluck('nombre','id')->
219             prepend('Seleccionar,');
220         $enfermedad_sintomas = EnfermedadSintomas::find($id_sintoma);
221         return view('enfermedad.sintomas-edit',compact('enfermedad','sintomas','enfermedad_sintomas'));
222     }else{
223         abort(404);
224     }
225 }
226
227 public function sintomaDelete($id)
228 {
229     EnfermedadSintomasDetalle::where('id_enfermedad_sintomas',$id)->delete();
230     EnfermedadSintomas::where('id',$id)->delete();
231
232     return response()->json(
233         array(
234             'result' => true,
235             'message' => "Sintoma eliminado correctamente"
236         )
237     );
238 }
239
240 public function sintomasDetalle($id_enfermedad_sintomas)
241 {
242     $data = EnfermedadSintomasDetalle::where('id_enfermedad_sintomas',$id_enfermedad_sintomas)->get();
243     return response()->json(
244         array(
245             'result' => true,
246             'data' => $data
247         )
248     );
249 }

```

```

248
249
250     public function sintomasDetalleDelete($id)
251     {
252         $validar = EnfermedadSintomasDetalle::find($id);
253         $result = true;
254         $message = "";
255         if($validar){
256             EnfermedadSintomasDetalle::where('id',$id)->delete();
257             $message = "Registro eliminado correctamente";
258         }else{
259             $result = false;
260             $message = "Registro no encontrado en la base de datos";
261         }
262         return response()->json(
263             array(
264                 'result' => $result,
265                 'message' => $message
266             )
267         );
268     }
269
270     public function consultarSintomasDetalle(Request $request)
271     {
272         $validar = EnfermedadSintomasDetalle::where('nombre',$request->nombre)
273         ->where('id_enfermedad_sintomas',$request->id_enfermedad_sintomas)
274         ->first();
275
276         $result = true;
277         $message = "";
278         $data = null;
279
280         if($validar){
281             $result = false;
282             $message = "Nombre ya existe";
283         }else{
284             $data = EnfermedadSintomasDetalle::create($request->all());
285             $message = "Registro agregado correctamente";
286         }
287
288         return response()->json(
289             array(
290                 'result' => $result,
291                 'data' => $data,
292                 'message' => $message
293             )
294         );
295     }

```

```

296
297     public function sintomasDetalleUpdate($id_enfermedad,Request $request)
298     {
299         $result = true;
300         $message = true;
301
302         $enfermedad_sintomas = EnfermedadSintomas::where("id_enfermedad",$id_enfermedad)
303         ->where('id','!',$request->id)
304         ->where('id_sintoma',$request->id_sintoma)
305         ->first();
306
307         if($enfermedad_sintomas){
308             $result = false;
309             $enfermedad = Enfermedad::find($id_enfermedad);
310             $message = "Síntoma ". $enfermedad_sintomas->sintomas->nombre. " ya está registrado para ". $enfermedad->nombre;
311         }else{
312             EnfermedadSintomas::where('id',$request->id)->update([
313                 'id_sintoma' => $request->id_sintoma
314             ]);
315
316             if($request->tipo_sintoma == 1){
317                 EnfermedadSintomasDetalle::where('id_enfermedad_sintomas',$request->id)->delete();
318                 foreach ($request->data_sintomas as $value) {
319                     EnfermedadSintomasDetalle::create([
320                         'id_enfermedad_sintomas' => $request->id,
321                         'nombre' => $value['nombre'],
322                         'puntaje' => $value['puntaje']
323                     ]);
324                 }
325             }else if($request->tipo_sintoma == 2){
326                 foreach ($request->data_sintomas as $value) {
327                     EnfermedadSintomasDetalle::where('id',$value['id'])->update([
328                         'puntaje' => $value['puntaje']
329                     ]);
330                 }
331             }
332             Alert::success('Síntoma actualizado correctamente', '')->autoclose(5000);
333         }
334
335         return response()->json(
336             array(
337                 'result' => $result,
338                 'message' => $message
339             )
340         );
341     }
342

```



Figura\_ 63: Modelo de Registro de enfermedades

```

1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7 use Illuminate\Database\Eloquent\SoftDeletes;
8
9 class Enfermedad extends Model
10 {
11     use HasFactory, SoftDeletes;
12
13     public $table = 'enfermedad';
14     protected $dates = ['deleted_at'];
15
16     public $fillable = [
17         'id_tipo_enfermedad',
18         'nombre',
19         'descripcion',
20         'puntaje'
21     ];
22
23     function tipoEnfermedad(){
24         return $this->belongsTo('App\Models\TipoEnfermedad', 'id_tipo_enfermedad', 'id');
25     }
26 }
27

```

Figura\_ 64: Vista de Registro de enfermedades

```

1 @extends('layouts.app')
2 @section('content')
3 @include('sweetalert:alert')
4 <div class="row no-gutters">
5     <div class="col-auto d-none d-sm-block">
6         <h3><strong>Enfermedades</strong></h3>
7     </div>
8
9     <div class="col-auto ms-auto text-end mt-n1">
10         <a href="{{ route('enfermedad.create') }}" class="btn btn-primary">Agregar Enfermedad</a>
11     </div>
12 </div>
13 <div class="card">
14     <div class="card-body">
15         <table id="enfermedad" class="table table-striped text-center" style="width:100%">
16             <thead>
17                 <tr>
18                     <th>#</th>
19                     <th>Nombre</th>
20                     <th>Tipo de Enfermedad</th>
21                     <th>Puntaje</th>
22                     <th class="no-export">Acciones</th>
23                 </tr>
24             </thead>
25             <tbody>
26                 <tr>
27                     <td></td>
28                 </tr>
29             </tbody>
30         </table>
31     </div>
32 @endsection
33
34 @section('js')
35 <script>
36     let table = $('#enfermedad').dataTable( {
37         "language": {
38             "lengthMenu": "Ver los _MENU_ Primeros Registros",
39             "info": "END de _TOTAL_ registros",
40             "infoEmpty": "No se encontraron registros",
41             "infoFiltered": "(Filtrado de _MAX_ total entradas)",
42             "loadingRecords": "Cargando...",
43             "processing": "Procesando...",
44             "search": "BUSCAR:",
45             "zeroRecords": "No se encontraron resultados",
46             "emptyTable": "Ningún dato disponible en esta tabla",
47             "paginate": {
48                 "first": "Primero",
49                 "last": "Ultimo",
50                 "next": "Siguiente",
51                 "previous": "Anterior"
52             },
53             "infoCallback": null
54         },
55         "processing": true,
56         "serverSide": true,
57         "destroy": true,
58         "ajax": {
59             "method": "GET",
60             "url": "{{ route('enfermedad.lista') }}"
61         },
62         "displayLength": 10,
63         dom: 'Brtip',
64         buttons: {
65             {
66                 extend: 'excelHtml5',
67                 title: 'sistema experto - Enfermedades',
68                 className: 'btn btn-success',
69                 text: '<span class="fa fa-file-excel"></span> EXCEL',
70                 exportOptions: { columns: "not(.no-export)" }
71             }
72         }
73     } );
74

```

```

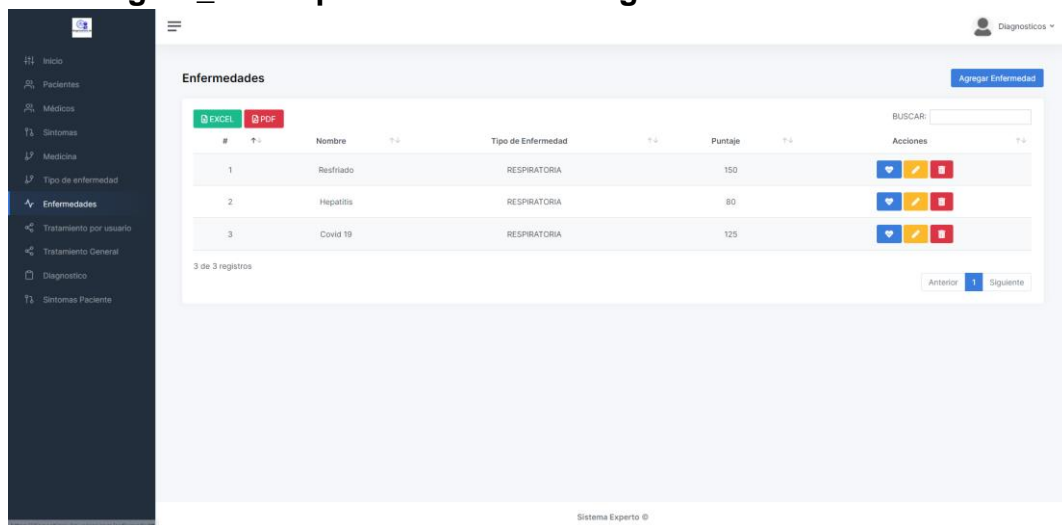
63     title: 'Sistema experto - Enfermedades',
64     className: 'btn btn-success',
65     text: '<i class="far fa-file-excel"></i> EXCEL',
66     exportOptions: { columns: ":not(.no-export)" }
67   },
68   {
69     extend: 'pdfHtml5',
70     title: 'Sistema experto - Enfermedades',
71     orientation: 'landscape',
72     className: 'btn btn-danger',
73     text: '<i class="far fa-file-pdf"></i> PDF',
74     exportOptions: { columns: ":not(.no-export)" }
75   },
76   ],
77 });
78
79 function deleteRegister(id){
80   Swal.fire({
81     title: "Esta seguro de eliminar este registro?",
82     text: "",
83     icon: "warning",
84     showCancelButton: true,
85     confirmButtonColor: "#3085d6",
86     cancelButtonColor: "#d32f2f",
87     confirmButtonText: 'Confirmar',
88     cancelButtonText: 'cancelar',
89   }).then((result) => {
90     if (result.value) {
91       $.ajax({
92         url: "{{URL::to('/enfermedad')}}" + "/" + id,
93         type: 'DELETE'
94       })
95       .done(function(res) {
96         if(res.result){
97           table.ajax.reload();
98           Swal.fire(res.message, "", "success");
99         }else{
100           Swal.fire(res.message, "", "error");
101         }
102       });
103     }
104   });
105 }
106 }
107 </script>
108 @endsection

```

## IMPLEMENTACIÓN

En la siguiente figura se presenta la interfaz de este requerimiento, la cual fue definida anteriormente para proceder con el desarrollo.

**Figura\_ 65: Implementación de Registro de enfermedades**

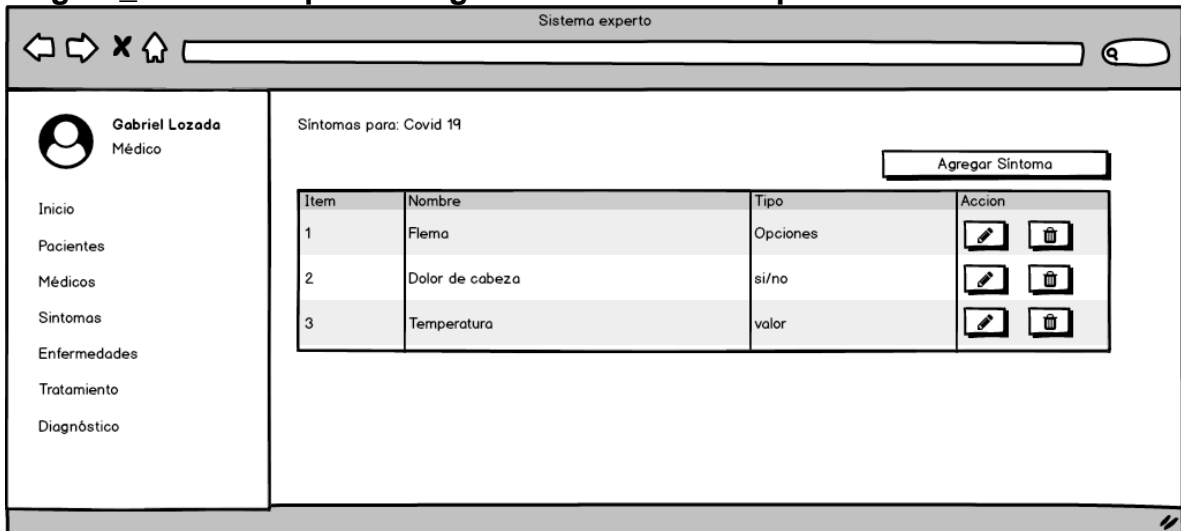


**RF10: El sistema web permitirá asignar los síntomas para las enfermedades esto según a los tipos que se especificaron, “opciones”, si/no y valor numérico.**

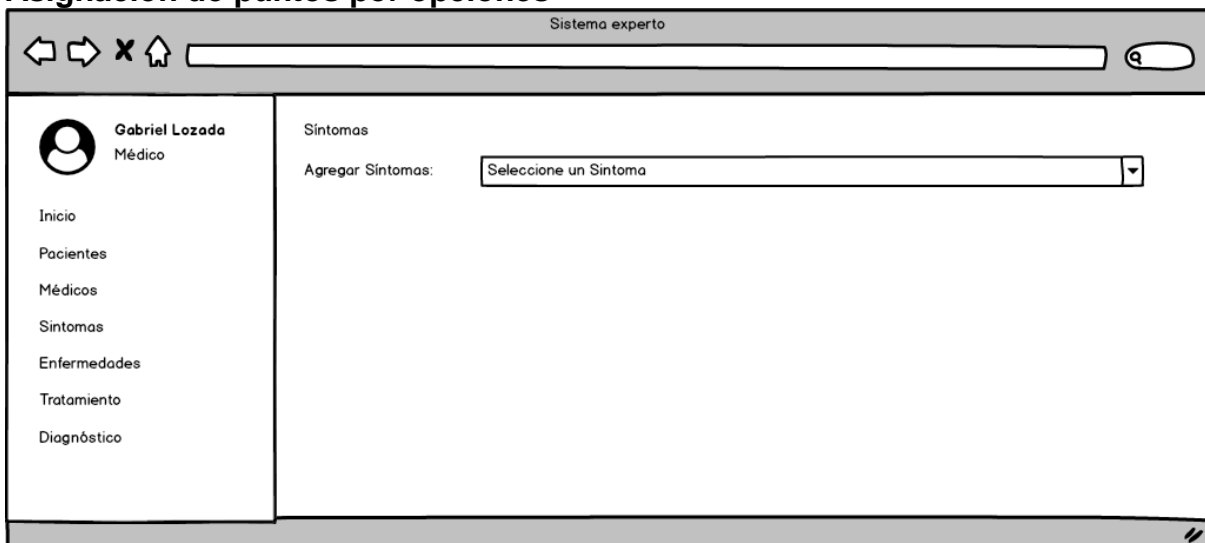
## DISEÑO PROTOTIPO

De la misma manera a continuación se muestra el prototipo desarrollado para este requerimiento, la cual fue presentado mediante el producto owner para poder continuar con la parte del desarrollo, el programa en el que se desarrollo fue el Balsamiq mockups.

**Figura\_ 66: Prototipo de Asignación de síntomas para las enfermedades**



### Asignación de puntos por opciones



Sistema experto

← → × ↶

🔍

**Gabriel Lozada**  
Médico

- Inicio
- Pacientes
- Médicos
- Sintomas
- Enfermedades
- Tratamiento
- Diagnóstico

Sintomas

Agregar Sintomas:

Item	Nombre	Puntaje
1	Amarilla	<input style="width: 60%; border: 1px solid gray;" type="text" value="10"/>
2	Transparente	<input style="width: 60%; border: 1px solid gray;" type="text" value="4"/>
3	Verde	<input style="width: 60%; border: 1px solid gray;" type="text" value="0"/>

### Asignación de puntos si/no

Sistema experto

← → × ↶

🔍

**Gabriel Lozada**  
Médico

- Inicio
- Pacientes
- Médicos
- Sintomas
- Enfermedades
- Tratamiento
- Diagnóstico

Sintomas

Agregar Sintomas:

Item	Nombre	Puntaje
1	Si	<input style="width: 60%; border: 1px solid gray;" type="text" value="10"/>
2	No	<input style="width: 60%; border: 1px solid gray;" type="text" value="1"/>

### Asignación de puntos por valores

Sistema experto

← → × ↶

🔍

**Gabriel Lozada**  
Médico

- Inicio
- Pacientes
- Médicos
- Sintomas
- Enfermedades
- Tratamiento
- Diagnóstico

Sintomas

Agregar Sintomas:

Rango  y  puntaje

Item	Nombre	Puntaje	Acciones
1	entre 37 a 39	<input style="width: 60%; border: 1px solid gray;" type="text" value="8"/>	<input style="width: 20px; height: 20px; border: 1px solid gray;" type="button" value="🗑"/>
2	entre 39 a 40	<input style="width: 60%; border: 1px solid gray;" type="text" value="10"/>	<input style="width: 20px; height: 20px; border: 1px solid gray;" type="button" value="🗑"/>
3	entre 40 a 45	<input style="width: 60%; border: 1px solid gray;" type="text" value="15"/>	<input style="width: 20px; height: 20px; border: 1px solid gray;" type="button" value="🗑"/>

## DESARROLLO

En las siguientes figuras se pueden observar la parte del desarrollo del sistema, la cual va comenzando por el controlador, modelo y vista.

Figura\_ 67: Controlador de Asignación de síntomas para las

```
EnfermedadController.php x
145
146 public function storeSintoma($id, Request $request)
147 {
148     $result = true;
149     $message = "";
150
151     $validar = EnfermedadSintomas::where('id_enfermedad', $id)
152     ->where('id_sintoma', $request->id_sintoma)
153     ->first();
154     if($validar){
155         $result = false;
156         $enfermedad = Enfermedad::find($id);
157         $message = "síntoma " . $validar->sintomas->nombre . " ya está registrado para " . $enfermedad->nombre;
158     }else{
159         $enfermedad_sintomas = EnfermedadSintomas::create([
160             'id_enfermedad' => $id,
161             'id_sintoma' => $request->id_sintoma
162         ]);
163
164         foreach ($request->data_sintomas as $value) {
165             $enfermedad_sintomas_detalle = create([
166                 'id_enfermedad_sintomas' => $enfermedad_sintomas->id,
167                 'nombre' => $value['nombre'],
168                 'puntaje' => $value['puntaje']
169             ]);
170         }
171
172         Alert::success('síntoma agregado correctamente', '')->autoclose(5000);
173     }
174
175     return response()->json(
176         array(
177             'result' => $result,
178             'message' => $message
179         )
180     );
181 }
182
183 public function listaSintomas($id)
184 {
185     $sintomas = EnfermedadSintomas::orderByDesc('id')
186     ->where('id_enfermedad', $id)
187     ->get();
188
189     $data = collect();
190     foreach ($sintomas as $key => $item) {
191         $route_edit = route('enfermedad.sintomas.edit', [$id, $item->id]);
192         $acciones = '<a href="' . $route_edit . '" class="btn btn-warning" data-toggle="tooltip" title="Editar"><i class="fas fa-pen"></i></a>
193         <button type="button" class="btn btn-danger" data-toggle="tooltip" title="Eliminar" onclick="deleteRegister(' . $item->id . ')"><i class="fas fa-trash-alt"></i></button>';
194
195         $data_sintomas = array(
196             ($key+1),
197             $item->sintomas->nombre,
198             $item->sintomas->setTipo($item->sintomas->tipo),
199             $acciones
200         );
201         $data->push($data_sintomas);
202     }
203     $results = array(
204         "draw" => 0,
205         "recordsTotal" => count($data),
206         "recordsFiltered" => count($data),
207         "data" => $data
208     );
209
210     return response()->json($results);
211 }
212 }
```

enfermedades

```

213 public function editSintoma($id_enfermedad,$id_sintoma)
214 {
215     $enfermedad = Enfermedad::find($id_enfermedad);
216     if($enfermedad){
217         $$sintomas = Sintomas::where('id_tipo_enfermedad',$enfermedad->id_tipo_enfermedad)->pluck('nombre','id')->
                prepend("seleccionar,");
218         $enfermedad_sintomas = EnfermedadSintomas::find($id_sintoma);
219         return view('enfermedad.sintomas-edit',compact('enfermedad','sintomas','enfermedad_sintomas'));
220     }else{
221         abort(404);
222     }
223 }
224
225 public function sintomaDelete($id)
226 {
227     EnfermedadSintomasDetalle::where('id_enfermedad_sintomas',$id)->delete();
228     EnfermedadSintomas::where('id',$id)->delete();
229
230     return response()->json(
231         array(
232             'result' => true,
233             'message' => "síntoma eliminado correctamente"
234         )
235     );
236 }
237
238 public function sintomasDetalle($id_enfermedad_sintomas)
239 {
240     $data = EnfermedadSintomasDetalle::where('id_enfermedad_sintomas',$id_enfermedad_sintomas)->get();
241     return response()->json(
242         array(
243             'result' => true,
244             'data' => $data
245         )
246     );
247 }
248
249 public function sintomasDetalleDelete($id)
250 {
251     $validar = EnfermedadSintomasDetalle::find($id);
252     $result = true;
253     $message = "";
254     if($validar){
255         EnfermedadSintomasDetalle::where('id',$id)->delete();
256         $message = "Registro eliminado correctamente";
257     }else{
258         $result = false;
259         $message = "Registro no encontrado en la base de datos";
260     }
261     return response()->json(
262         array(
263             'result' => $result,
264             'message' => $message
265         )
266     );
267 }
268 }

```

Figura\_ 68: Modelo de Asignación de síntomas para las enfermedades

```

1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7
8 class EnfermedadSintomas extends Model
9 {
10     use HasFactory;
11
12     public $table = 'enfermedad_sintomas';
13     public $timestamps = false;
14
15     public $fillable = [
16         'id_enfermedad',
17         'id_sintoma'
18     ];
19
20     function enfermedad(){
21         return $this->belongsTo('App\Models\Enfermedad','id_enfermedad','id');
22     }
23
24     function sintomas(){
25         return $this->belongsTo('App\Models\Sintomas','id_sintoma','id');
26     }
27
28 }
29

```

Figura\_ 69: Vista de Asignación de síntomas para las enfermedades

```

sintomas.blade.php
1 @extends('layouts.app')
2 @section('content')
3 @include('sweetalert::alert')
4 <div class="row mb-2 mb-xl-3">
5 <div class="col-auto d-none d-sm-block">
6 <h3><strong>Síntomas para {{ $enfermedad->nombre }}</strong></h3>
7 </div>
8
9 <div class="col-auto ms-auto text-end mt-n1">
10 <a href="{{ route('enfermedad.sintomas.create', $enfermedad->id) }}" class="btn btn-primary">Agregar Síntoma</a>
11 </div>
12 </div>
13 <div class="card">
14 <div class="card-body">
15 <table id="sintomas" class="table table-striped text-center" style="width:100%">
16 <thead>
17 <tr>
18 <th>#</th>
19 <th>Nombre</th>
20 <th>Tipo</th>
21 <th class="no-export">Acciones</th>
22 </tr>
23 </thead>
24 </table>
25 </div>
26 </div>
27 @endsection
28
29 @section('js')
30 <script>
31 let url = "{{URL::to('/enfermedad')}}" + "/" + "{{ $enfermedad->id }}" + "/sintomas/lista";
32 let nombre_enfermedad = "{{ $enfermedad->nombre }}";
33 let table = $('#sintomas').dataTable( {
34   "language": {
35     "lengthMenu": "Ver los _MENU_ Primeros Registros",
36     "info": "_END_ de _TOTAL_ registros",
37     "infoEmpty": "No se encontraron registros",
38     "infoFiltered": "(filtrado de _MAX_ total entradas)",
39     "loadingRecords": "Cargando...",
40     "processing": "Procesando...",
41     "sSearch": "BUSCAR:",
42     "sZeroRecords": "No se encontraron resultados",
43     "sEmptyTable": "Ningún dato disponible en esta tabla",
44     "oPaginate": {
45       "sFirst": "Primero",
46       "sLast": "Último",
47       "sNext": "Siguiente",
48       "sPrevious": "Anterior"
49     },
50     "fnInfoCallback": null
51   },
52   "aProcessing": true,
53   "aServerSide": true,
54   "destroy": true,
55   "ajax": {
56     "method": "GET",
57     "url": url
58   },
59   "iDisplayLength": 10,
60   dom: 'Bfrtip',
61   buttons: [
62     {
63       extend: 'excelHtml5',
64       title: 'Síntomas para ' + nombre_enfermedad,
65       className: 'btn btn-success',
66       text: '<i class="far fa-file-excel"></i> EXCEL',
67       exportoptions: { columns: ":not(.no-export)" }
68     },

```

```
sintomas.blade.php
83     EXCEL => EXCELHTML5,
84     title: 'Síntomas para '+nombre_enfermedad,
85     className: 'btn btn-success',
86     text: '<i class="far fa-file-excel"></i> EXCEL',
87     exportOptions: { columns: ":not(.no-export)" }
88   },
89   {
90     extend: 'pdfHtml5',
91     title: 'Síntomas para '+nombre_enfermedad,
92     orientation: 'landscape',
93     className: 'btn btn-danger',
94     text: '<i class="far fa-file-pdf"></i> PDF',
95     exportOptions: { columns: ":not(.no-export)" }
96   },
97   ],
98 });
99
100 function deleteRegister(id){
101   Swal.fire({
102     title: '¿Está seguro de eliminar este registro?',
103     text: "",
104     icon: 'warning',
105     showCancelButton: true,
106     confirmButtonColor: '#3085d6',
107     cancelButtonColor: '#d33',
108     confirmButtonText: 'Confirmar',
109     cancelButtonText: 'Cancelar'
110   }).then((result) => {
111     if (result.value) {
112       $.ajax({
113         url: "{{URL::to('/enfermedad')}}" + "/" + id + "/sintomas",
114         type: 'DELETE'
115       })
116       .done(function(res) {
117         if(res.result){
118           table.ajax.reload();
119           Swal.fire(res.message,"","success");
120         }else{
121           Swal.fire("Ocurrió un error","","error");
122         }
123       });
124     }
125   });
126 }
127 </script>
128 @endsection
129
```

## IMPLEMENTACIÓN

En la siguiente figura se presenta la interfaz de este requerimiento, la cual fue definida anteriormente para proceder con el desarrollo.



**Figura\_ 70: Implementación de Asignación de síntomas para las enfermedades**

Sistema Experto ©

RF11: El sistema web permite registrar diagnósticos de la cual se puede tener como resultados varias enfermedades. Así mismo se deberá evaluar cada enfermedad en base a síntomas.

## DISEÑO

### PROTOTIPO

De la misma manera a continuación se muestra el prototipo desarrollado para este requerimiento, la cual fue presentado mediante el producto owner para poder continuar con la parte del desarrollo, el programa en el que se desarrollo fue el Balsamiq mockups.

Figura\_ 71: Prototipo de Registro de diagnósticos

The image displays two screenshots of a web application prototype titled 'Sistema experto'. Both screenshots feature a left sidebar with a user profile for 'Gabriel Lozada Médico' and a navigation menu with items: Inicio, Pacientes, Médicos, Síntomas, Enfermedades, Tratamiento, and Diagnóstico.

**Top Screenshot: Diagnóstico**

Search bar:

Paciente: Juan Lopez Perez

Item	Fecha	Doctor	Acciones
1	01 Nov de 2021	Perez Albela	<input type="button" value="Tratamiento"/>
2	20 Nov de 2021	Dr. House	<input type="button" value="Tratamiento"/>

**Bottom Screenshot: Registro de Diagnóstico**

Paciente: Juan Lopez Perez

Tipo enfermedad:

Selecciones los síntomas del paciente, si no aplica deje en blanco

1. Flema:

2. Dolor de cabeza:  Si  No

3. Temeperatura:

## DESARROLLO

En las siguientes figuras se pueden observar la parte del desarrollo del sistema, la cual va comenzando por el controlador, modelo y vista.

**Figura\_ 72: Controlador de Registro de diagnósticos**

```
DiagnosticoController.php x
1 |<?php
2
3 | namespace App\Http\Controllers;
4
5 | use Illuminate\Http\Request;
6 | use App\Models\Enfermedad;
7 | use App\Models\EnfermedadSintomas;
8 | use App\Models\EnfermedadSintomasDetalle;
9 | use App\Models\EnfermedadTratamiento;
10 | use App\Models\EnfermedadTratamientoDetalle;
11 | use App\Models\TipoEnfermedad;
12 | use App\Models\Diagnostico;
13 | use App\Models\DiagnosticoEnfermedad;
14 | use App\Models\User;
15 | use App\Models\Sintomas;
16 | use App\Models\Medicina;
17 | use App\Models\UsuarioDiagnostico;
18 | use App\Models\UsuarioDiagnosticoSintomas;
19 | use Illuminate\Support\Facades\Auth;
20 | use Alert;
21
22
23 | class DiagnosticoController extends Controller
24 | {
25 |     public function index()
26 |     {
27 |         return view('diagnostico.index');
28 |     }
29
30 |     public function create($id_usuario)
31 |     {
32 |         $user = User::find($id_usuario);
33 |         if($user){
34 |             $tipo_enfermedad = TipoEnfermedad::all()->pluck('nombre','id')->prepend('Seleccionar,');
35 |             return view('diagnostico.create',compact('user','tipo_enfermedad'));
36 |         }else{
37 |             abort(404);
38 |         }
39 |     }
40 | }
```

```

40
41 public function store(Request $request)
42 {
43     $diagnostico = Diagnostico::create([
44         'id_paciente' => $request->id_paciente,
45         'id_medico' => Auth::user()->id,
46         'id_tipo_enfermedad' => $request->id_tipo_enfermedad
47     ]);
48
49     $enfermedades = Enfermedad::where('id_tipo_enfermedad',$request->id_tipo_enfermedad)->get();
50     foreach ($enfermedades as $value_enfermedad) {
51         $validar = false;
52         $puntaje = 0;
53         $enfermedad_sintomas = EnfermedadSintomas::where('id_enfermedad',$value_enfermedad->id)->get();
54         foreach ($request->data_sintomas as $value_sintomas_input) {
55             foreach ($enfermedad_sintomas as $value_sintomas) {
56                 if($value_sintomas_input['id'] == $value_sintomas->id_sintoma){
57                     $enfermedad_sd = EnfermedadSintomasDetalle::where('id_enfermedad_sintomas',$value_sintomas->id)
58                         ->get();
59                     foreach ($enfermedad_sd as $value_esd) {
60                         if($value_sintomas_input['tipo'] == 1 || $value_sintomas_input['tipo'] == 2){
61                             if($value_sintomas_input['value'] == $value_esd->nombre){
62                                 $validar = true;
63                                 $puntaje += $value_esd->puntaje;
64                             }
65                         }else if($value_sintomas_input['tipo'] == 3){
66                             $rango = explode(" ", $value_esd->nombre);
67                             if(isset($value_sintomas_input['value'])){
68                                 if($rango[1] <= $value_sintomas_input['value'] && $rango[3] >= $value_sintomas_input
69                                     ['value']){
70                                     $validar = true;
71                                     $puntaje += $value_esd->puntaje;
72                                 }
73                             }
74                         }
75                     }
76                 }
77             }
78         }
79         if($validar){
80             $diagnosticoEnfermedad::create([
81                 'id_diagnostico' => $diagnostico->id,
82                 'id_enfermedad' => $value_enfermedad->id,
83                 'probabilidad' => $puntaje
84             ]);
85         }
86     }
87
88     return response()->json(
89         array(
90             'result' => true,
91             'data' => $diagnostico->id
92         )
93     );
94 }
95
96 public function resultados($id_diagnostico)
97 {
98     $diagnostico = Diagnostico::find($id_diagnostico);
99     if($diagnostico){
100         $enfermedades = DiagnosticoEnfermedad::where('id_diagnostico',$id_diagnostico)->get();
101         return view('diagnostico.resultado',compact('diagnostico','enfermedades'));
102     }else{
103         abort(404);
104     }
105 }

```

```

105
106 public function tratamientoCreate($id_diagnostico_enfermedad)
107 {
108     $diagnostico_enfermedad = DiagnosticoEnfermedad::find($id_diagnostico_enfermedad);
109     $diagnostico = Diagnostico::find($diagnostico_enfermedad->id_diagnostico);
110     $enfermedades = Enfermedad::all()->pluck('nombre','id')->prepend('Todos','');
111     $medicina = Medicina::all()->pluck('nombre','id')->prepend('Seleccionar','');
112
113     $enfermedad_tratamiento = EnfermedadTratamiento::where('id_diagnostico',$diagnostico->id)
114     ->where('id_enfermedad',$diagnostico_enfermedad->id_enfermedad)
115     ->where('id_usuario',$diagnostico->paciente->id)
116     ->whereDate('fecha',$diagnostico->created_at)
117     ->first();
118
119     if($enfermedad_tratamiento){
120         return view('diagnostico.tratamiento-edit',compact('diagnostico_enfermedad','diagnostico','enfermedades','
121             medicina','enfermedad_tratamiento'));
122     }else{
123         return view('diagnostico.tratamiento-create',compact('diagnostico','diagnostico_enfermedad','enfermedades',
124             medicina));
125     }
126 }
127
128 public function tratamientoStore(Request $request)
129 {
130     $enfermedad_tratamiento = EnfermedadTratamiento::create($request->all());
131
132     foreach ($request->data_medicina as $value) {
133         EnfermedadTratamientoDetalle::create([
134             'id_enfermedad_tratamiento' => $enfermedad_tratamiento->id,
135             'id_medicina' => $value['id_medicina'],
136             'frecuencia' => $value['frecuencia'],
137             'cantidad' => $value['cantidad'],
138             'detalles' => $value['detalles']
139         ]);
140     }
141
142     Alert::success('Tratamiento agregado correctamente', '')->autoclose(5000);
143
144     return response()->json(
145         array(
146             'result' => true,
147             'data' => $enfermedad_tratamiento->id
148         )
149     );
150 }
151
152 public function tratamientoUpdate($id,Request $request)
153 {
154     $enfermedad_tratamiento = EnfermedadTratamiento::where('id',$id)->update($request->all());
155
156     return response()->json(
157         array(
158             'result' => true,
159             'message' => "Tratamiento actualizado correctamente"
160         )
161     );
162 }

```

```

162 public function buscarUsuario(Request $request)
163 {
164     $result = true;
165     $message = "";
166     $data = null;
167
168     if($request->id_usuario){
169         $user = User::where('id_perfil',2)->where('id',$request->id_usuario)->first();
170         if($user){
171             $data = $user;
172         }else{
173             $result = false;
174             $message = "Usuario no encontrado en nuestros registros";
175         }
176     }else{
177         $user = User::where('dni',$request->dni)
178             ->where('id_perfil',2)
179             ->first();
180         if($user){
181             $data = $user;
182         }else{
183             $result = false;
184             $message = "DNI no encontrado en nuestros registros";
185         }
186     }
187
188     return response()->json(
189         array(
190             'result' => $result,
191             'data' => $data,
192             'message' => $message
193         )
194     );
195 }
196
197 public function listaUsuarios(Request $request)
198 {
199     if(Auth::user()->id_perfil == 1){
200         $diagnostico = Diagnostico::where('id_paciente',$request->id_usuario)
201             ->orderByDesc('created_at')
202             ->get();
203     }else if(Auth::user()->id_perfil == 3){
204         $diagnostico = Diagnostico::where('id_paciente',$request->id_usuario)
205             ->where('id_medico',Auth::user()->id)
206             ->orderByDesc('created_at')
207             ->get();
208     }
209
210     $data = collect();
211     foreach ($diagnostico as $key => $item) {
212         $route_edit = route('diagnostico.resultados',$item->id);
213         $acciones = '<a href="'.$route_edit.'" class="btn btn-success" data-toggle="tooltip" title="Tratamiento"><i
                class="fas fa-notes-medical"></i>Tratamiento</a>';
214
215         $data_diagnostico = array(
216             ($key+1),
217             date('d/m/Y',strtotime($item->created_at)),
218             $item->medico->nombres." ".$item->medico->apellidos,
219             $acciones
220         );
221         $data->push($data_diagnostico);
222     }
223     $results = array(
224         "draw" => 0,
225         "recordsTotal" => count($data),
226         "recordsFiltered" => count($data),
227         "data" => $data
228     );
229
230     return response()->json($results);
231 }

```

```

232
233 public function getSintomas($id_tipo_enfermedad)
234 {
235     $data = Sintomas::where('id_tipo_enfermedad',$id_tipo_enfermedad)
236     ->orderBy('nombre','asc')
237     ->get();
238     return response()->json(
239         array(
240             'result' => true,
241             'data' => $data
242         )
243     );
244 }
245
246 public function indexUsersSintomas()
247 {
248     $tipo_enfermedad = TipoEnfermedad::all()->pluck('nombre','id')->prepend('TODOS','');
249     return view('sintomas-paciente.index',compact('tipo_enfermedad'));
250 }
251
252 public function listaUsersintomas(Request $request)
253 {
254     $usuario_diagnostico = UsuarioDiagnostico::where(function ($query) use ($request) {
255         if($request->tipo_enfermedad){
256             $query = $query->where('id_tipo_enfermedad',$request->tipo_enfermedad);
257         }
258     })
259     ->get();
260
261     $data = collect();
262     foreach ($usuario_diagnostico as $key => $item) {
263         $route_edit = route('sintomas.paciente.detalle',$item->id);
264         $acciones='<a href="'.$route_edit.'" class="btn btn-primary" data-toggle="tooltip" title="Ver síntomas"><i
                class="fas fa-eye"></i></a>';
265         $estado = "";
266         if($item->estado == 0){
267             $estado = '<span class="badge bg-warning">PENDIENTE</span>';
268         }else if($item->estado == 1){
269             $estado = '<span class="badge bg-success">ATENDIDO</span>';
270         }
271
272         $data_diagnostico_sintomas = array(
273             ($key+1),
274             $item->usuario->nombres." " . $item->usuario->apellidos,
275             $item->tipo_enfermedad->nombre,
276             date('d/m/Y h:i a',strtotime($item->created_at)),
277             $estado,
278             $acciones
279         );
280         $data->push($data_diagnostico_sintomas);
281     }
282     $results = array(
283         "draw" => 0,
284         "recordsTotal" => count($data),
285         "recordsFiltered" => count($data),
286         "data" => $data
287     );
288     return response()->json($results);
289 }
290
291 public function sintomasDetallePaciente($id_usuario_diagnostico)
292 {
293     $usuario_diagnostico = UsuarioDiagnostico::find($id_usuario_diagnostico);
294     return view('sintomas-paciente.create',compact('usuario_diagnostico'));
295 }
296
297

```

```

298     public function listarUsuarioDiagnosticoSintoma(Request $request)
299     {
300         $usuario_diagnostico = UsuarioDiagnostico::find($request->id_usuario_diagnostico);
301         $usuario_diagnostico_sintomas = UsuarioDiagnosticoSintomas::where('id_usuario_diagnostico',$request->
            id_usuario_diagnostico)->get();
302
303         $data = collect();
304         foreach ($usuario_diagnostico_sintomas as $key => $item) {
305             $acciones = '<button type="button" class="btn btn-danger" data-toggle="tooltip" title="Eliminar Síntomas"
                onclick="eliminarSintoma('.$item->id.')"><i class="fas fa-trash-alt"></i></button>';
306
307             if($usuario_diagnostico->tipo == 0){
308                 $data_diagnostico_sintomas = array(
309                     ($key+1),
310                     $item->sintoma->nombre,
311                     $item->value,
312                     $acciones
313                 );
314             }else if($usuario_diagnostico->tipo == 1){
315                 $data_diagnostico_sintomas = array(
316                     ($key+1),
317                     $item->sintoma->nombre,
318                     $item->value
319                 );
320             }
321             $data->push($data_diagnostico_sintomas);
322         }
323         $results = array(
324             "draw" => 0,
325             "recordsTotal" => count($data),
326             "recordsFiltered" => count($data),
327             "data" => $data
328         );
329
330         return response()->json($results);
331     }
332
333     public function eliminarUsuarioDiagnosticoSintoma($id_usuario_diagnostico_sintomas)
334     {
335         UsuarioDiagnosticoSintomas::where('id',$id_usuario_diagnostico_sintomas)->delete();
336         return response()->json(
337             array(
338                 'result' => true,
339                 'message' => 'Sintoma eliminado correctamente'
340             )
341         );
342     }
343 }

```

```

344     public function procesarSintomas($id_usuario_diagnostico)
345     {
346         $usuario_diagnostico = UsuarioDiagnostico::find($id_usuario_diagnostico);
347         $usuario_diagnostico_sintomas = UsuarioDiagnosticoSintomas::where('id_usuario_diagnostico',$
            id_usuario_diagnostico)->get();
348
349         $diagnostico = Diagnostico::create([
350             'id_paciente' => $usuario_diagnostico->id_usuario,
351             'id_medico' => Auth::user()->id,
352             'id_tipo_enfermedad' => $usuario_diagnostico->id_tipo_enfermedad
353         ]);
354
355         $enfermedades = Enfermedad::where('id_tipo_enfermedad',$usuario_diagnostico->id_tipo_enfermedad)->get();
356         foreach ($enfermedades as $value_enfermedad) {
357             $validar = false;
358             $puntaje = 0;
359             $enfermedad_sintomas = EnfermedadSintomas::where('id_enfermedad',$value_enfermedad->id)->get();
360             foreach ($usuario_diagnostico_sintomas as $value_sintomas_input) {
361                 foreach ($enfermedad_sintomas as $value_sintomas) {
362                     if($value_sintomas_input->id_sintoma == $value_sintomas->id_sintoma){
363                         $enfermedad_sd = EnfermedadSintomasDetalle::where('id_enfermedad_sintomas',$value_sintomas->id)
                            ->get();
364                         foreach ($enfermedad_sd as $value_esd) {
365                             if($value_sintomas_input->sintoma->tipo == 1 || $value_sintomas_input->sintoma->tipo == 2){
366                                 if($value_sintomas_input->value == $value_esd->nombre){
367                                     $validar = true;
368                                     $puntaje += $value_esd->puntaje;
369                                 }
370                             }else if($value_sintomas_input->sintoma->tipo == 3){
371                                 $rango = explode(" ", $value_esd->nombre);
372                                 if(isset($value_sintomas_input->value)){
373                                     if($rango[1] <= $value_sintomas_input->value && $rango[3] >= $value_sintomas_input->
                                    value){
374                                         $validar = true;
375                                         $puntaje += $value_esd->puntaje;
376                                     }
377                                 }
378                             }
379                         }
380                     }
381                 }
382             }
383             if($validar){
384                 DiagnosticoEnfermedad::create([
385                     'id_diagnostico' => $diagnostico->id,
386                     'id_enfermedad' => $value_enfermedad->id,
387                     'probabilidad' => $puntaje
388                 ]);
389             }
390         }
391
392         UsuarioDiagnostico::where('id',$id_usuario_diagnostico)->update([
393             'estado' => 1
394         ]);
395
396         Alert::success('síntomas procesado correctamente', '')->autoclose(5000);
397         return redirect(route('síntomas.paciente.detalle',$id_usuario_diagnostico));
398     }
399 }
400 }

```



Figura\_ 73: Modelo de Registro de diagnósticos

```
Diagnostico.php x
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7 use Illuminate\Database\Eloquent\SoftDeletes;
8
9 class Diagnostico extends Model
10 {
11     use HasFactory,SoftDeletes;
12
13     public $table = 'diagnostico';
14     protected $dates = ['deleted_at'];
15
16     public $fillable = [
17         'id_paciente',
18         'id_medico',
19         'id_tipo_enfermedad'
20     ];
21
22     function paciente(){
23         return $this->belongsTo('App\Models\User','id_paciente','id');
24     }
25
26     function medico(){
27         return $this->belongsTo('App\Models\User','id_medico','id');
28     }
29
30     function tipoEnfermedad(){
31         return $this->belongsTo('App\Models\TipoEnfermedad','id_tipo_enfermedad','id');
32     }
33 }
34
```

Figura\_ 74: Vista de Asignación de Registro de diagnósticos

```
index.blade.php x
1 @extends('layouts.app')
2 @section('content')
3 @include('sweetalert::alert')
4 <div class="row mb-2 mb-xl-3">
5     <div class="col-auto d-none d-sm-block">
6         <h3><strong>Diagnóstico</strong></h3>
7     </div>
8 </div>
9 <div class="card">
10     <div class="card-body">
11         <div class="row">
12             <div class="d-sm-inline-block" id="form-diagnostico">
13                 <div class="col-sm-12 mb-4">
14                     <div class="input-group input-group-navbar">
15                         <input type="text" class="form-control" id="dni" placeholder="Ingrese DNI del paciente"
16                             aria-label="Search" maxlength="8" onkeypress="return validaNumericos(event)">
17                         <button class="btn" type="button">
18                             <i class="align-middle" data-feather="search"></i>
19                         </button>
20                     </div>
21                 </div>
22             </div>
23 </div>
24 <div id="resultados" class="row d-none">
25     <div class="col-sm-8 mb-4 pt-2 ps-3">
26         <strong></strong>
27     </div>
28     <div class="col-sm-4 mb-4 text-sm-end">
29         <a href="#" class="btn btn-primary">Nuevo Diagnóstico</a>
30     </div>
31 </div>
32 <div class="table-responsive">
33     <table id="paciente" class="table table-striped text-center" style="width:100%">
34         <thead>
35             <tr>
36                 <th>#</th>
37                 <th>Fecha</th>
38                 <th>Doctor</th>
39                 <th class="no-export">Acciones</th>
40             </tr>
41         </thead>
42     </table>
43 </div>
44 </div>
45 @endsection
```

```

47 @section('js')
48 <script>
49     var id_usuario;
50
51     $("#form-diagnostico").submit(function(e) {
52         e.preventDefault();
53         let dni = $("#dni").val();
54         loading("show");
55         $.post('{{ route('diagnostico.search') }}', {dni: dni}, function(res) {
56             loading("hide");
57             if(res.result){
58                 $("#resultados").removeClass('d-none');
59                 $("#resultados strong").html(res.data.nombres + "-res.data.apellidos);
60                 id_usuario = res.data.id;
61                 let url = "{{URL::to('/diagnostico')}}" + "/" + id_usuario + "/create";
62                 $("#resultados a").attr("href", url);
63                 lista();
64             }else{
65                 $("#resultados").addClass('d-none');
66                 Swal.fire(res.message, 'error');
67             }
68         });
69     });
70
71     function lista(){
72         $('#paciente').DataTable( {
73             "language": {
74                 "lengthMenu": "Ver los _MENU_ Primeros Registros",
75                 "info": " _END_ de _TOTAL_ registros",
76                 "infoEmpty": "No se encontraron registros",
77                 "infoFiltered": "(Filtrado de _MAX_ total entradas)",
78                 "loadingRecords": "Cargando...",
79                 "processing": "Procesando...",
80                 "sSearch": "BUSCAR:",
81                 "sZeroRecords": "No se encontraron resultados",
82                 "sEmptyTable": "Ningún dato disponible en esta tabla",
83                 "oPaginate": {
84                     "sFirst": "Primero",
85                     "sLast": "Último",
86                     "sNext": "Siguiente",
87                     "sPrevious": "Anterior"
88                 },
89                 "fnInfoCallback": null
90             },
91             "aProcessing": true,
92             "aServerSide": true,
93             "destroy": true,
94             "ajax": {
95                 "method": "POST",
96                 "url": "{{ route('diagnostico.lista') }}",
97                 "data": {id_usuario: id_usuario}
98             },
99             "iDisplayLength": 10,
100            dom: 'Bfrtip',
101            buttons: [
102                {
103                    extend: 'excelHtml5',
104                    title: 'Sistema experto - Diagnóstico',
105                    className: 'btn btn-success',
106                    text: '<i class="far fa-file-excel"></i> EXCEL',
107                    exportOptions: { columns: ":not(.no-export)" }
108                },
109                {
110                    extend: 'pdfHtml5',
111                    title: 'Sistema experto - Diagnóstico',
112                    orientation: 'landscape',
113                    className: 'btn btn-danger',
114                    text: '<i class="far fa-file-pdf"></i> PDF',
115                    exportOptions: { columns: ":not(.no-export)" }
116                },
117            ],
118        });

```

## IMPLEMENTACIÓN

En la siguiente figura se presenta la interfaz de este requerimiento, la cual fue definida anteriormente para proceder con el desarrollo.

**Figura\_ 75: Implementación de Registro de diagnósticos**

#	Fecha	Doctor	Acciones
1	11/01/2022	Diagnosticos Pulmonares	Tratamiento
2	30/12/2021	Diagnosticos Pulmonares	Tratamiento
3	30/12/2021	Diagnosticos Pulmonares	Tratamiento
4	30/12/2021	Diagnosticos Pulmonares	Tratamiento
5	06/12/2021	Diagnosticos Pulmonares	Tratamiento
6	03/12/2021	Diagnosticos Pulmonares	Tratamiento

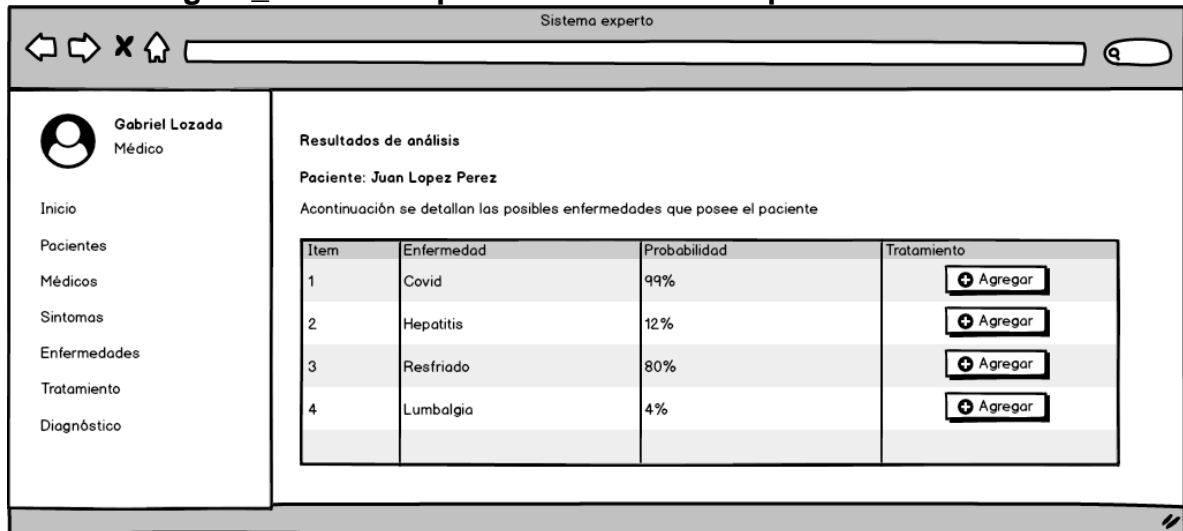
**RF12: El sistema web permitirá mostrar la probabilidad por cada enfermedad a la cual se podrá asignar un tratamiento por cada enfermedad detectada.**

## DISEÑO

### PROTOTIPO

De la misma manera a continuación se muestra el prototipo desarrollado para este requerimiento, la cual fue presentado mediante el producto owner para poder continuar con la parte del desarrollo, el programa en el que se desarrollo fue el Balsamiq mockups.

Figura\_ 76: Prototipo de Resultados del procesamiento



## DESARROLLO

En las siguientes figuras se pueden observar la parte del desarrollo del sistema, la cual va comenzando por el controlador, modelo y vista.

Figura\_ 77: Controlador de Resultados del procesamiento

```
DiagnosticoController.php x
40
41 public function store(Request $request)
42 {
43     $diagnostico = Diagnostico::create([
44         'id_paciente' => $request->id_paciente,
45         'id_medico' => Auth::user()->id,
46         'id_tipo_enfermedad' => $request->id_tipo_enfermedad
47     ]);
48
49     $enfermedades = Enfermedad::where('id_tipo_enfermedad',$request->id_tipo_enfermedad)->get();
50     foreach ($enfermedades as $value_enfermedad) {
51         $validar = false;
52         $puntaje = 0;
53         $enfermedad_sintomas = EnfermedadSintomas::where('id_enfermedad',$value_enfermedad->id)->get();
54         foreach ($request->data_sintomas as $value_sintomas_input) {
55             foreach ($enfermedad_sintomas as $value_sintomas) {
56                 if($value_sintomas_input['id'] == $value_sintomas->id_sintoma){
57                     $enfermedad_sd = EnfermedadSintomasDetalle::where('id_enfermedad_sintomas',$value_sintomas->id)
58                         ->get();
59                     foreach ($enfermedad_sd as $value_esd) {
60                         if($value_sintomas_input['tipo'] == 1 || $value_sintomas_input['tipo'] == 2){
61                             if($value_sintomas_input['value'] == $value_esd->nombre){
62                                 $validar = true;
63                                 $puntaje += $value_esd->puntaje;
64                             }
65                         }else if($value_sintomas_input['tipo'] == 3){
66                             $rango = explode(" ", $value_esd->nombre);
67                             if(isset($value_sintomas_input['value'])){
68                                 if($rango[1] <= $value_sintomas_input['value'] && $rango[3] >= $value_sintomas_input
69                                     ['value']){
70                                     $validar = true;
71                                     $puntaje += $value_esd->puntaje;
72                                 }
73                             }
74                         }
75                     }
76                 }
77             }
78         }
79         if($validar){
80             DiagnosticoEnfermedad::create([
81                 'id_diagnostico' => $diagnostico->id,
82                 'id_enfermedad' => $value_enfermedad->id,
83                 'probabilidad' => $puntaje
84             ]);
85         }
86     }
87
88     return response()->json(
89         array(
90             'result' => true,
91             'data' => $diagnostico->id
92         )
93     );
94 }
95
96 public function resultados($id_diagnostico)
97 {
98     $diagnostico = Diagnostico::find($id_diagnostico);
99     if($diagnostico){
100         $enfermedades = DiagnosticoEnfermedad::where('id_diagnostico',$id_diagnostico)->get();
101         return view('diagnostico.resultado',compact('diagnostico','enfermedades'));
102     }else{
103         abort(404);
104     }
105 }
```

Figura\_ 78: Modelo de Resultados del procesamiento

```

1 | k?php
2 |
3 | namespace App\Models;
4 |
5 | use Illuminate\Database\Eloquent\Factories\HasFactory;
6 | use Illuminate\Database\Eloquent\Model;
7 |
8 | class DiagnosticoEnfermedad extends Model
9 | {
10 |     use HasFactory;
11 |     public $table = 'diagnostico_enfermedad';
12 |     public $timestamps = false;
13 |
14 |     public $fillable = [
15 |         'id_diagnostico',
16 |         'id_enfermedad',
17 |         'probabilidad'
18 |     ];
19 |
20 |     function diagnostico(){
21 |         return $this->belongsTo('App\Models\Diagnostico','id_diagnostico','id');
22 |     }
23 |
24 |     function enfermedad(){
25 |         return $this->belongsTo('App\Models\Enfermedad','id_enfermedad','id');
26 |     }
27 | }
28 |

```

Figura\_ 79: Vista de Resultados del procesamiento

```

1 | @extends('layouts.app')
2 |
3 | @section('content')
4 | <div class="row mb-2 mb-xl-3">
5 |     <div class="col-auto d-none d-sm-block">
6 |         <h3><strong>Resultados de análisis</strong></h3>
7 |     </div>
8 | </div>
9 | <div class="card">
10 |     <div class="card-header">
11 |         <h5 class="card-title">Paciente: {{ $diagnostico->paciente->nombre." ".$diagnostico->paciente->apellidos }}</h5>
12 |         <h6 class="card-subtitle text-muted">A continuación se detallan las posibles enfermedades que posee el paciente</h6>
13 |     </div>
14 |     <div class="card-body">
15 |         <table class="table text-center">
16 |             <thead>
17 |                 <tr>
18 |                     <th>#</th>
19 |                     <th>Enfermedad</th>
20 |                     <th>Probabilidad</th>
21 |                     <th>Tratamiento</th>
22 |                 </tr>
23 |             </thead>
24 |             <tbody>
25 |                 @foreach($enfermedades as $key => $item)
26 |                 <tr>
27 |                     <td>{{ $key+1 }}</td>
28 |                     <td>{{ $item->enfermedad->nombre }}</td>
29 |                     <td>{{ $item->probabilidad }}%</td>
30 |                     <td>
31 |                         <a href="{{ route('diagnostico.tratamiento.create',$item->id) }}" class="btn btn-primary">
32 |                             <i class="fas fa-plus"></i>
33 |                             Agregar
34 |                         </a>
35 |                     </td>
36 |                 </tr>
37 |                 @endforeach
38 |             </tbody>
39 |         </table>
40 |     </div>
41 | </div>
42 | @endsection

```

## IMPLEMENTACIÓN

En la siguiente figura se presenta la interfaz de este requerimiento, la cual fue definida anteriormente para proceder con el desarrollo.

**Figura\_ 80: Implementación de Resultados del procesamiento**

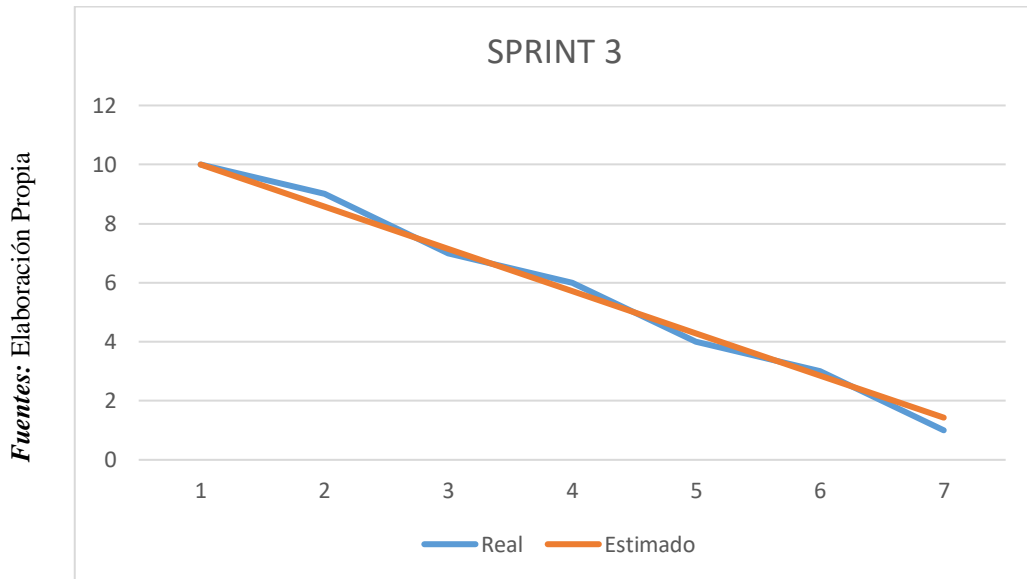
The screenshot displays a web application interface for a diagnostic system. On the left is a dark sidebar menu with the following items: Inicio, Pacientes, Médicos, Síntomas, Medicina, Tipo de enfermedad, Enfermedades, Tratamiento por usuario, Tratamiento General, Diagnostico (highlighted), and Síntomas Paciente. The main content area has a header with a hamburger menu icon and a user profile icon labeled 'Diagnosticos'. Below the header, the section is titled 'Resultados de análisis'. It shows patient information: 'Paciente: Diego Simon Samanlego' and a note: 'A continuación se detallan las posibles enfermedades que posee el paciente'. A table follows with three rows of results:

#	Enfermedad	Probabilidad	Tratamiento
1	Covid 19	75%	+ Agregar
2	Hepatitis	40%	+ Agregar
3	Resfriado	120%	+ Agregar

At the bottom of the page, the text 'Sistema Experto ©' is visible.

## BURNDOWN DEL SPRINT N° 3

Figura\_ 81: Burndown del Sprint 3



### Burndown Sprint 3

A continuación, se presentará el significado de la figura anterior: comenzando con la línea roja la cual representa el tiempo ideal del desarrollo de la misma, así mismo la línea azul representa el tiempo real en la que se desarrolló dicho sprint. Es así como se puede decir que si la línea azul se ubica por debajo de la línea roja es porque existió un adelanto en la elaboración del sprint, y de manera contraria si la línea azul está por encima de la roja significa retraso en la elaboración del sprint.



### ACTA DE REUNION DEL SPRINT N° 3

Siendo las 3 pm del día 2 de noviembre del 2021, se reúne en el Policlínico Monserrat.

Presentes:

ROL	NOMBRE
Scrum Master	Luz Gomez Taype
Team Member	Gianmarco Roger Nizama Antunez
Product Owner	Miguel Rios

El encargado del Policlínico Monserrat, Luz Gomez Taype culmina la reunión con el Sr. Gianmarco Nizama señalando los últimos puntos respecto al sprint.

De esta manera se logra definir la forma de trabajo y todos los requerimientos respecto a este sprint, así mismo firmar el acta que contiene la fecha de entrega, esto para el cierre de la reunión.

Los asistentes presentes aprobaron los puntos que fueron acordados en esta reunión, así mismo beneficio la planificación de este Sprint, realizando el compromiso de entregar los resultados referentes al tiempo que fue pactado.


### ACTA DE ENTREGA DEL SPRINT N° 3

Siendo la 1 pm del día 21 de noviembre del 2021 se reúne en el Policlínico Monserrat.

ROL	NOMBRE
Scrum Master	Luz Gomez Taype
Team Member	Gianmarco Roger Nizama Antunez
Product Owner	Miguel Rios

El Sr. Gianmarco Nizama, señala y detalla cada requerimiento desarrollado, así mismo presenta las interfaces obtenidas del product owner.

Luego de realizar las verificaciones a la sustentación del el Sr. Gianmarco Nizama para continuar con la aprobación del Sprint, de manera unánime se decide la aprobación de este sprint, referente al proyecto “Sistema experto para diagnóstico de enfermedades pulmonares en el policlínico Monserrat”.

Es así como los presentes imparten su aprobación al informe del el Sr. Gianmarco Nizama, referente al Sprint N° 3 la cual fue concluida, del siguiente proyecto “Sistema experto para diagnóstico de enfermedades pulmonares en el policlínico Monserrat”


## RESUMEN DE LA REUNIÓN RETROSPECTIVA DE SPRINT N° 3

Información de la empresa y proyecto:

<b>Empresa / Organización</b>	Policlínico Monserrat
<b>Proyecto</b>	“Sistema experto para diagnóstico de enfermedades pulmonares en el policlínico Monserrat”

Información de la reunión:

<b>Lugar</b>	Policlínico Monserrat
<b>Fecha</b>	21/11/2021
<b>Número de iteración / Sprint</b>	Sprint 3
<b>Personas Convocadas a la reunión</b>	
<b>Persona que asistieron a la reunión</b>	

Formulario de reunión retrospectiva

<b>¿Qué salió bien en la Iteración? (Aciertos)</b>	<b>¿Qué no salió bien en la Iteración? (Errores)</b>
<p>Se logró ejecutar el registro de enfermedades como también la asignación de síntomas para cada enfermedad.</p> <p>Se logró desarrollar el registro de diagnósticos como también los resultados de todo el procesamiento.</p>	<p>Se generaron fallos al registrar y mostrar los diagnósticos, sin embargo, se logró solucionar.</p>

## PLANIFICACIÓN DEL SPRINT N°4

Siendo las 4 pm del día 21 de noviembre del 2021, se reúne en el Policlínico Monserrat.

Presentes:

ROL	NOMBRE
Scrum Master	Luz Gomez Taype
Team Member	Gianmarco Roger Nizama Antunez
Product Owner	Miguel Rios

El gerente del Policlínico Monserrat, ejecutó la presentación de todos los requerimientos obtenidos, de igual manera indica los que son de mayor prioridad.

Luego de ser analizada los requerimientos presentados por el gerente del Policlínico Monserrat. La señora Luz Gomez aclara ciertas dudas y da su compromiso para el cumplimiento de los requerimientos expuestos en el Sprint 4.

Así mismo los presentes imparten su aprobación respecto a lo expuesto esto específicamente en el Sprint 4, señalando que la fecha de entrega respecto a este Sprint será el 7 de diciembre del 2021.


## EJECUCIÓN DEL SPRINT 4

Anterior a la fase del diseño se necesita saber exactamente el funcionamiento del sistema, conocer y entender cada parte de ella, dicho de una manera diferente el análisis respectivo a lo que se requiere esto de acuerdo a las historias de usuario.

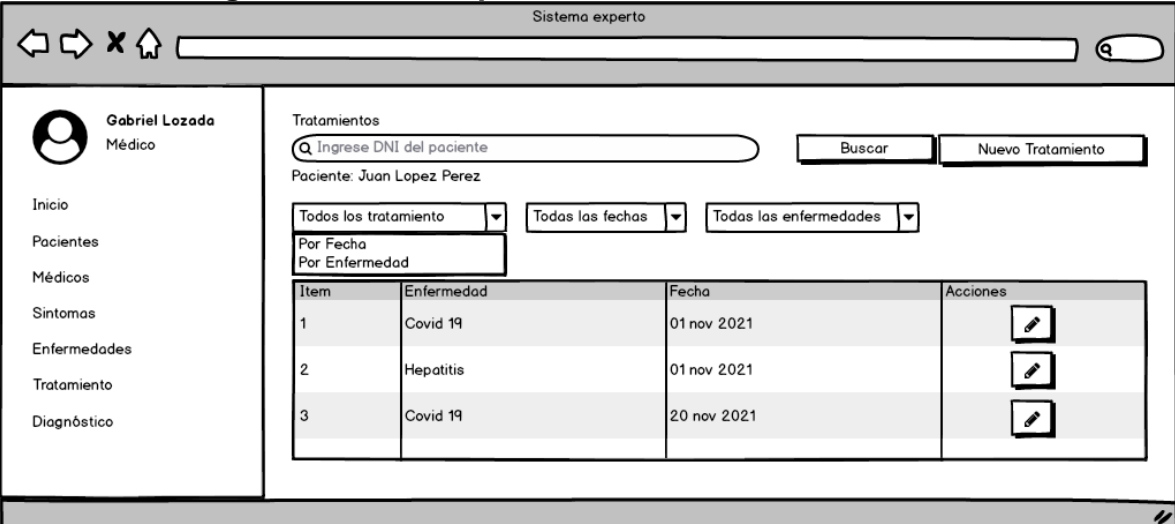
**RF13: El sistema permitirá mostrar los tratamientos del paciente, las cuales podrán ser varias, sin embargo, en distintas fechas. Estos tratamientos deberán ser como recetas respecto al peso y talla de los pacientes.**

## DISEÑO

### PROTOTIPO

De la misma manera a continuación se muestra el prototipo desarrollado para este requerimiento, la cual fue presentado mediante el producto owner para poder continuar con la parte del desarrollo, el programa en el que se desarrollo fue el Balsamiq mockups.

**Figura\_ 82: Prototipo de Gestión de tratamientos**



El prototipo muestra una interfaz de usuario para la gestión de tratamientos. En la parte superior, hay un navegador con el título "Sistema experto". A la izquierda, un menú de navegación muestra el perfil de "Gabriel Lozada Médico" y opciones como Inicio, Pacientes, Médicos, Síntomas, Enfermedades, Tratamiento y Diagnóstico. El área principal está dividida en secciones: "Tratamientos" con un campo de búsqueda "Ingrese DNI del paciente" y botones "Buscar" y "Nuevo Tratamiento"; "Paciente: Juan Lopez Perez"; y tres filtros desplegables: "Todos los tratamiento", "Todas las fechas" y "Todas las enfermedades". Debajo de los filtros, hay un menú desplegable con "Por Fecha" y "Por Enfermedad". En la parte inferior, una tabla muestra los tratamientos:

Item	Enfermedad	Fecha	Acciones
1	Covid 19	01 nov 2021	[Editar]
2	Hepatitis	01 nov 2021	[Editar]
3	Covid 19	20 nov 2021	[Editar]

## DESARROLLO

En las siguientes figuras se pueden observar la parte del desarrollo del sistema, la cual va comenzando por el controlador, modelo y vista.

Figura\_ 83: Controlador de Gestión de tratamientos

```
TratamientoController.php x
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use App\Models\Enfermedad;
7 use App\Models\Tratamiento;
8 use App\Models\TratamientoDetalle;
9 use App\Models\Medicina;
10 use App\Models\User;
11 use App\Models\EnfermedadTratamiento;
12 use App\Models\EnfermedadTratamientoDetalle;
13 use Alert;
14 use PDF;
15
16 class TratamientoController extends Controller
17 {
18     public function index()
19     {
20         $enfermedades = Enfermedad::all()->pluck('nombre','id')->prepend('Todos,');
21         return view('tratamiento-general.index',compact('enfermedades'));
22     }
23
24     public function lista(Request $request)
25     {
26         if($request->id_enfermedad){
27             $tratamientos = Tratamiento::where('id_enfermedad',$request->id_enfermedad)
28                 ->orderByDesc('created_at')
29                 ->get();
30         }else{
31             $tratamientos = Tratamiento::orderByDesc('created_at')->get();
32         }
33
34         $data = collect();
35         foreach ($tratamientos as $key => $item) {
36             $route_edit = route('tratamiento.edit',$item->id);
37             $acciones = <a href="$route_edit" class="btn btn-warning" data-toggle="tooltip" title="Editar">i
38                 class="fas fa-pen"</i></a>
39             <button type="button" class="btn btn-danger" data-toggle="tooltip" title="Eliminar" onclick="deleteRegister(
40                 '$item->id')">i class="fas fa-trash-alt"</i></button>;
41             $data_tratamiento = array(
42                 ($key->id),
43                 $item->enfermedad->nombre,
44                 $item->edad_min." a ".$item->edad_max." años",
45                 $item->peso_min." a ".$item->peso_max." kilos",
46                 $acciones
47             );
48             $data->push($data_tratamiento);
49         }
50         $results = array(
51             "draw" => 0,
52             "recordsTotal" => count($data),
53             "recordsFiltered" => count($data),
54             "data" => $data
55         );
56         return response()->json($results);
57     }
58
59     public function create()
60     {
61         $enfermedades = Enfermedad::all()->pluck('nombre','id')->prepend('Seleccionar,');
62         $medicina = Medicina::all()->pluck('nombre','id')->prepend('Seleccionar,');
63         return view('tratamiento-general.create',compact('medicina','enfermedades'));
64     }
65 }
```

```

TratamientoController.php x
64
65 public function store(Request $request)
66 {
67     $result = true;
68     $message = "";
69
70     $validar_tratamiento = Tratamiento::where('id_enfermedad',$request->id_enfermedad)->first();
71
72     if($validar_tratamiento){
73         $result = false;
74         $enfermedad = Enfermedad::find($request->id_enfermedad);
75         $message = "Ya existe un tratamiento con la enfermedad ".$enfermedad->nombre;
76     }else{
77         $tratamiento = Tratamiento::create([
78             'id_enfermedad' => $request->id_enfermedad,
79             'edad_min' => $request->edad_min,
80             'edad_max' => $request->edad_max,
81             'peso_min' => $request->peso_min,
82             'peso_max' => $request->peso_max,
83             'recomendaciones' => $request->recomendaciones
84         ]);
85
86         foreach ($request->data_medicina as $value) {
87             TratamientoDetalle::create([
88                 'id_tratamiento' => $tratamiento->id,
89                 'id_medicina' => $value['id_medicina'],
90                 'frecuencia' => $value['frecuencia'],
91                 'cantidad' => $value['cantidad'],
92                 'detalles' => $value['detalles']
93             ]);
94         }
95         Alert::success('Tratamiento agregado', 'Registro guardado correctamente')->autoclose(5000);
96     }
97
98
99     return response()->json(
100         array(
101             'result' => $result,
102             'message' => $message
103         )
104     );
105 }
106
107 public function edit($id)
108 {
109     $tratamiento = Tratamiento::find($id);
110     if($tratamiento){
111         $enfermedades = Enfermedad::all()->pluck('nombre','id')->prepend('Seleccionar','');
112         $medicina = Medicina::all()->pluck('nombre','id')->prepend('Seleccionar','');
113         return view('tratamiento-general.edit',compact('tratamiento','enfermedades','medicina'));
114     }else{
115         abort(404);
116     }
117 }
118

```

```
TratamientoController.php x
119 public function update($id, Request $request)
120 {
121     $result = true;
122     $message = "";
123
124     $validar_tratamiento = Tratamiento::where('id_enfermedad', $request->id_enfermedad)
125     ->where('id', '!=', $id)
126     ->first();
127
128     if($validar_tratamiento){
129         $result = false;
130         $enfermedad = Enfermedad::find($request->id_enfermedad);
131         $message = "Ya existe un tratamiento con la enfermedad ".$enfermedad->nombre;
132     }else{
133         Tratamiento::where('id', $id)->update($request->all());
134         Alert::success('Tratamiento actualizado correctamente', '')->autoclose(5000);
135     }
136
137     return response()->json(
138         array(
139             'result' => $result,
140             'message' => $message
141         )
142     );
143 }
144
145 public function listaDetalle($id_tratamiento)
146 {
147     $data = TratamientoDetalle::where('id_tratamiento', $id_tratamiento)
148     ->with('medicina')
149     ->get();
150     return response()->json(
151         array(
152             'result' => true,
153             'data' => $data
154         )
155     );
156 }
157
158 public function agregarDetalle(Request $request)
159 {
160     $result = true;
161     $message = "";
162     $data = null;
163
164     $tratamiento_detalle = TratamientoDetalle::where('id_tratamiento', $request->id_tratamiento)
165     ->where('id_medicina', $request->id_medicina)
166     ->first();
167
168     if($tratamiento_detalle){
169         $result = false;
170         $message = "Medicina ya está registrado";
171     }else{
172         TratamientoDetalle::create($request->all());
173         $message = "Medicina agregado correctamente";
174         $data = TratamientoDetalle::where('id_tratamiento', $request->id_tratamiento)
175         ->with('medicina')
176         ->get();
177     }
178
179     return response()->json(
180         array(
181             'result' => $result,
182             'data' => $data,
183             'message' => $message
184         )
185     );
186 }
187 }
```



```

188     public function eliminarDetalle($id)
189     {
190         $result = true;
191         $message = "";
192         $data = null;
193
194         $tratamiento_detalle = TratamientoDetalle::where('id',$id)->first();
195         if($tratamiento_detalle){
196             $id_tratamiento = $tratamiento_detalle->id_tratamiento;
197             $tratamiento_detalle->delete();
198             $message = "Medicina eliminado correctamente";
199             $data = TratamientoDetalle::where('id_tratamiento',$id_tratamiento)
200                 ->with('medicina')
201                 ->get();
202         }else{
203             $result = false;
204             $message = "Registro no existe";
205         }
206     }
207
208     return response()->json(
209         array(
210             'result' => $result,
211             'data' => $data,
212             'message' => $message
213         )
214     );
215 }
216
217 public function eliminar($id)
218 {
219     Tratamiento::where('id',$id)->delete();
220     return response()->json(
221         array(
222             'result' => true,
223             'message' => "Tratamiento eliminado correctamente"
224         )
225     );
226 }
227
228 public function indexUser()
229 {
230     return view('tratamiento.index');
231 }
232
233 public function createUser($id_user)
234 {
235     $usuario = User::find($id_user);
236     if($usuario){
237         $enfermedades = Enfermedad::all()->pluck('nombre','id')->prepend('Todos,');
238         $medicina = Medicina::all()->pluck('nombre','id')->prepend('Seleccionar,');
239         return view('tratamiento.create',compact('usuario','enfermedades','medicina'));
240     }else{
241         abort(404);
242     }
243 }
244

```

```

244
245 public function storeUser(Request $request)
246 {
247     $result = true;
248     $message = "";
249
250     $validar_tratamiento = EnfermedadTratamiento::where('id_enfermedad',$request->id_enfermedad)
251         ->whereDate('fecha',$request->fecha)
252         ->where('id_usuario',$request->id_usuario)
253         ->whereNull('id_diagnostico')
254         ->first();
255
256     if($validar_tratamiento){
257         $result = false;
258         $enfermedad = Enfermedad::find($request->id_enfermedad);
259         $message = "Ya existe un tratamiento con la enfermedad ".$enfermedad->nombre." en la fecha ".date('d/m/Y',
260             strtotime($request->fecha));
261     }else{
262         $enfermedad_tratamiento = EnfermedadTratamiento::create([
263             'id_enfermedad' => $request->id_enfermedad,
264             'id_usuario' => $request->id_usuario,
265             'fecha' => $request->fecha,
266             'recomendaciones' => $request->recomendaciones
267         ]);
268
269         foreach ($request->data_medicina as $value) {
270             EnfermedadTratamientoDetalle::create([
271                 'id_enfermedad_tratamiento' => $enfermedad_tratamiento->id,
272                 'id_medicina' => $value['id_medicina'],
273                 'frecuencia' => $value['frecuencia'],
274                 'cantidad' => $value['cantidad'],
275                 'detalles' => $value['detalles']
276             ]);
277         }
278         Alert::success('Tratamiento agregado correctamente', '')->autoclose(5000);
279     }
280
281     return response()->json(
282         array(
283             'result' => $result,
284             'message' => $message
285         )
286     );
287 }
288

```

```
TratamientoController.php x
289 public function listaUser(Request $request)
290 {
291     $tratamientos = EnfermedadTratamiento::orderBy('fecha','asc')
292     ->where('id_usuario',$request->id_usuario)
293     ->where(function ($query) use ($request) {
294         if($request->fecha){
295             if($request->tipo == 1){
296                 $query = $query->where('fecha',$request->fecha);
297             }
298         }
299         if($request->id_enfermedad){
300             if($request->tipo == 2){
301                 $query = $query->where('id_enfermedad',$request->id_enfermedad);
302             }
303         }
304     })
305     ->whereNull('id_diagnostico')
306     ->get();
307
308     $data = collect();
309     foreach ($tratamientos as $key => $item) {
310         $route_edit = route('tratamiento.user.edit',$item->id);
311         $acciones='<a href="'.$route_edit.'" class="btn btn-warning" data-toggle="tooltip" title="Editar"><i
312             class="fas fa-pen"></i></a>
313         <button type="button" class="btn btn-danger" data-toggle="tooltip" title="Eliminar" onclick="deleteRegister(
314             '.$item->id.')"><i class="fas fa-trash-alt"></i></button>';
315         $data_tratamiento = array(
316             ($key+1),
317             $item->enfermedad->nombre,
318             date('d/m/Y',strtotime($item->fecha)),
319             $acciones
320         );
321         $data->push($data_tratamiento);
322     }
323     $results = array(
324         "draw" => 0,
325         "recordsTotal" => count($data),
326         "recordsFiltered" => count($data),
327         "data" => $data
328     );
329     return response()->json($results);
330 }
```

```

330
331 public function filtrosUser(Request $request)
332 {
333     $fechas = EnfermedadTratamiento::select('fecha')
334     ->orderBy('fecha', 'asc')
335     ->where('id_usuario', $request->id_usuario)
336     ->whereNull('id_diagnostico')
337     ->groupBy('fecha')
338     ->get();
339
340     $enfermedades = EnfermedadTratamiento::select('enfermedad_tratamiento.id_enfermedad', 'enfermedad.nombre')
341     ->join('enfermedad', 'enfermedad_tratamiento.id_enfermedad', '=', 'enfermedad.id')
342     ->orderBy('enfermedad.nombre', 'asc')
343     ->where('enfermedad_tratamiento.id_usuario', $request->id_usuario)
344     ->whereNull('enfermedad_tratamiento.id_diagnostico')
345     ->groupBy('enfermedad_tratamiento.id_enfermedad', 'enfermedad.nombre')
346     ->get();
347
348     $data = array(
349         'fechas' => $fechas,
350         'enfermedades' => $enfermedades
351     );
352
353     return response()->json(
354         array(
355             'result' => true,
356             'data' => $data
357         )
358     );
359 }
360
361 public function editUser($id)
362 {
363     $enfermedad_tratamiento = EnfermedadTratamiento::find($id);
364     if($enfermedad_tratamiento){
365         $medicina = Medicina::all()->pluck('nombre', 'id')->prepend('seleccionar,');
366         $enfermedades = Enfermedad::all()->pluck('nombre', 'id')->prepend('Todos,');
367         return view('tratamiento.edit', compact('enfermedad_tratamiento', 'enfermedades', 'medicina'));
368     }else{
369         abort(404);
370     }
371 }
372
373 public function updateUser($id, Request $request)
374 {
375     $result = true;
376     $message = "";
377
378     $validar_tratamiento = EnfermedadTratamiento::where('id_enfermedad', $request->id_enfermedad)
379     ->where('id', '!=', $id)
380     ->whereDate('fecha', $request->fecha)
381     ->where('id_usuario', $request->id_usuario)
382     ->whereNull('id_diagnostico')
383     ->first();
384
385     if($validar_tratamiento){
386         $result = false;
387         $enfermedad = Enfermedad::find($request->id_enfermedad);
388         $message = "Ya existe un tratamiento con la enfermedad ".$enfermedad->nombre." en la fecha ".$date('d/m/Y',
389             strtotime($request->fecha));
390     }else{
391         EnfermedadTratamiento::where('id', $id)->update($request->all());
392         Alert::success('Tratamiento actualizado correctamente', '')->autoClose(5000);
393     }
394     return response()->json(
395         array(
396             'result' => $result,
397             'message' => $message
398         )
399     );
}

```

```

TratamientoController.php x
401 public function eliminarUser($id)
402 {
403     EnfermedadTratamiento::where('id',$id)->delete();
404     return response()->json(
405         array(
406             'result' => true,
407             'message' => "Tratamiento eliminado correctamente"
408         )
409     );
410 }
411
412 public function tratamientoDetalle($id_enfermedad_tratamiento)
413 {
414     $data = EnfermedadTratamientoDetalle::select(
415         'enfermedad_tratamiento_detalle.*',
416         'medicina.nombre as nombre_medicina'
417     )
418     ->join('medicina','enfermedad_tratamiento_detalle.id_medicina','medicina.id')
419     ->where('enfermedad_tratamiento_detalle.id_enfermedad_tratamiento',$id_enfermedad_tratamiento)->get();
420     return response()->json(
421         array(
422             'result' => true,
423             'data' => $data
424         )
425     );
426 }
427
428 public function tratamientoDetallestore(Request $request)
429 {
430     EnfermedadTratamientoDetalle::create($request->all());
431     return $this->tratamientoDetalle($request->id_enfermedad_tratamiento);
432 }
433
434 public function deleteTratamientoDetalle($id)
435 {
436     $enfermedad_tratamiento_detalle = EnfermedadTratamientoDetalle::find($id);
437     EnfermedadTratamientoDetalle::where('id',$id)->delete();
438     return $this->tratamientoDetalle($enfermedad_tratamiento_detalle->id_enfermedad_tratamiento);
439 }
440

```

```

TratamientoController.php x
440
441 public function consultarMedicinas(Request $request)
442 {
443     $result = true;
444     $data = null;
445     $user = User::find($request->id_usuario);
446     $edad = $this->calculaEdad($user->fecha_nacimiento);
447
448     $tratamiento = Tratamiento::where('edad_min','<=', $edad)
449     ->where('edad_max','>=', $edad)
450     ->where('peso_min','<=', $user->peso)
451     ->where('peso_max','>=', $user->peso)
452     ->where('id_enfermedad',$request->id_enfermedad)
453     ->first();
454
455     if($tratamiento){
456         $data = TratamientoDetalle::select(
457             'tratamiento_detalle.id_medicina',
458             'medicina.nombre as nombre_medicina',
459             'tratamiento_detalle.frecuencia',
460             'tratamiento_detalle.cantidad',
461             'tratamiento_detalle.detalles'
462         )
463         ->join('medicina','tratamiento_detalle.id_medicina','-', 'medicina.id')
464         ->where('tratamiento_detalle.id_tratamiento',$tratamiento->id)
465         ->get();
466     }else{
467         $result = false;
468     }
469
470
471     return response()->json(
472         array(
473             'result' => $result,
474             'data' => $data
475         )
476     );
477 }
478
479 function calculaEdad($fechanacimiento){
480     list($ano,$mes,$dia) = explode("-",$fechanacimiento);
481     $ano_diferencia = date("Y") - $ano;
482     $mes_diferencia = date("m") - $mes;
483     $dia_diferencia = date("d") - $dia;
484     if ($dia_diferencia < 0 || $mes_diferencia < 0)
485         $ano_diferencia--;
486     return $ano_diferencia;
487 }
488
489 public function generarPdf($id_enfermedad_tratamiento)
490 {
491     $enfermedad_tratamiento = EnfermedadTratamiento::find($id_enfermedad_tratamiento);
492     $enfermedad_tratamiento_detalle = EnfermedadTratamientoDetalle::where('id_enfermedad_tratamiento',$
493     enfermedad_tratamiento->id)
494     ->get();
495     $pdf = PDF::loadView('pdf.tratamiento',compact('enfermedad_tratamiento','enfermedad_tratamiento_detalle'));
496     return $pdf->stream('Tratamiento '.$enfermedad_tratamiento->enfermedad->nombre.' - '.$enfermedad_tratamiento->
497     usuario->nombr.' '.$enfermedad_tratamiento->usuario->apellidos.'.pdf');
498 }
499

```

Figura\_ 84: Modelo de Gestión de tratamientos

```

1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7 use Illuminate\Database\Eloquent\SoftDeletes;
8
9 class Tratamiento extends Model
10 {
11     use HasFactory, SoftDeletes;
12     public $table = 'tratamiento';
13     protected $dates = ['deleted_at'];
14
15     public $fillable = [
16         'id_enfermedad',
17         'edad_min',
18         'edad_max',
19         'peso_min',
20         'peso_max',
21         'recomendaciones'
22     ];
23
24     function enfermedad(){
25         return $this->belongsTo('App\Models\Enfermedad', 'id_enfermedad', 'id');
26     }
27 }
28

```

Figura\_ 85: Vista de Gestión de tratamientos

```

1 @extends('layouts.app')
2 @section('content')
3 @include('sweetalert:alert')
4 <div class="row mb-2 mb-md-3">
5     <div class="col-auto d-none d-sm-block">
6         <h3><strong>Tratamientos</strong></h3>
7     </div>
8 </div>
9 <div class="card">
10     <div class="card-body">
11         <div class="row">
12             <form class="d-sm-inline-block" id="form-paciente">
13                 <div class="col-sm-12 mb-4">
14                     <div class="input-group input-group-navbar">
15                         <input type="text" class="form-control" id="dni" placeholder="Ingrese DNI del paciente"
16                             aria-label="Search" maxlength="8" onkeypress="return validaNumericos(event)">
17                         <button class="btn" type="button">
18                             <i class="align-middle" data-feather="search"></i>
19                     </button>
20                 </div>
21             </form>
22         </div>
23         <div id="resultados" class="row d-none">
24             <div class="col-sm-8 mb-4 pt-2 ps-3">
25                 <strong></strong>
26             </div>
27             <div class="col-sm-4 mb-4 text-sm-end">
28                 <a href="#" class="btn btn-primary">Nuevo Tratamiento</a>
29             </div>
30         </div>
31         <div class="row">
32             <div class="col-sm-3 mb-4">
33                 <select id="tipo" class="form-select">
34                     <option value="">Todos los tratamientos</option>
35                     <option value="1">Por Fecha</option>
36                     <option value="2">Por Enfermedad</option>
37                 </select>
38             <div class="col-sm-3 mb-4 fecha d-none">
39                 <select id="fecha" class="form-control select2"></select>
40             </div>
41             <div class="col-sm-3 mb-4 enfermedad d-none">
42                 <select id="enfermedades" class="form-control select2"></select>
43             </div>
44         </div>
45     </div>
46     <div class="table-responsive">
47         <table id="paciente" class="table table-striped text-center" style="width:100%">
48             <thead>
49                 <tr>
50                     <th>#</th>
51                     <th>Enfermedad</th>
52                     <th>Fecha</th>
53                     <th class="no-export">Acciones</th>
54                 </tr>
55             </thead>
56             <tbody>
57                 <table>
58             </tbody>
59         </table>
60     </div>
61 @endsection

```

```

index.blade.php
61
62 @section('js')
63 <script>
64     var id_usuario = null;
65     var fecha = null;
66     var enfermedades = null;
67     var tipo = 1;
68     var nombre_usuario = null;
69
70     function lista(){
71         listaDataTable();
72         $.post('{{ route('tratamiento.user.filtros') }}', {id_usuario: id_usuario}, function(res) {
73             if(res.result){
74                 $('#fecha').empty();
75                 $('#fecha').prepend("<option value>Todas las fechas</option>");
76                 $.each(res.data.fecha, function (i, item) {
77                     $('#fecha').append("<option>", {
78                         value: item.fecha,
79                         text : formato(item.fecha)
80                     });
81                 });
82
83                 $('#enfermedades').empty();
84                 $('#enfermedades').prepend("<option value>Todas las enfermedades</option>");
85                 $.each(res.data.enfermedades, function (i, item) {
86                     $('#enfermedades').append("<option>", {
87                         value: item.id_enfermedad,
88                         text : item.nombre
89                     });
90                 });
91                 $(".select2").select2();
92             }
93         });
94     }
95
96     function listaDataTable(){
97         $('#paciente').dataTable( {
98             "language": {
99                 "lengthMenu": "Ver los_MENU_Primeros Registros",
100                 "info": "Elm de_TOTAL registros",
101                 "infoEmpty": "No se encontraron registros",
102                 "infoFiltered": "(filtrado de_MAX_total entradas)",
103                 "loadingRecords": "Cargando...",
104                 "processing": "Procesando...",
105                 "sSearch": "BUSCAR:",
106                 "sZeroRecords": "No se encontraron resultados",
107                 "sEmptyTable": "Ningún dato disponible en esta tabla",
108                 "oPaginate": {
109                     "sFirst": "Primer",
110                     "sLast": "Último",
111                     "sNext": "Siguiente",
112                     "sPrevious": "Anterior"
113                 },
114                 "fnInfoCallback": null
115             },
116             "aProcessing": true,
117             "aServerSide": true,
118             "destroy": true,
119             "ajax": {
120                 "method": "POST",
121                 "url": "{{ route('tratamiento.user.lista') }}",
122                 "data": {
123                     id_usuario: id_usuario,
124                     fecha: fecha,
125                     id_enfermedad: enfermedades,
126                     tipo: tipo
127                 }
128             },

```

```

index.blade.php
189
190 $('#form-paciente').submit(function(e) {
191     e.preventDefault();
192     id_usuario = null;
193     consultarUsuario();
194 });
195
196 function consultarUsuario()
197 {
198     let dni = $('#dni').val();
199     loading("show");
200     $.post('{{ route('diagnostico.search') }}', {dni: dni, id_usuario: id_usuario}, function(res) {
201         loading("hide");
202         if(res.result){
203             $('#resultados').removeClass('d-none');
204             nombre_usuario = res.data.nombres + " " + res.data.apellidos;
205             $('#resultados strong').html(nombre_usuario);
206             id_usuario = res.data.id;
207             let url = "{{url::to('/tratamiento-user')}}" + "/" + id_usuario + "/create";
208             $('#resultados a').attr("href", url);
209             lista();
210         }else{
211             $('#resultados').addClass('d-none');
212             Swal.fire(res.message, "", 'error');
213         }
214     });
215 }
216
217 $('#tipo').change(function() {
218     let tipo_search = $(this).val();
219     if(tipo_search == 1){
220         $('#fecha').val("");
221         $('#fecha').removeClass('d-none');
222         $('#enfermedad').addClass('d-none');
223         $(".select2").select2();
224         fecha = null;
225         tipo = 1;
226     }else if(tipo_search == 2){
227         $('#enfermedades').val("");
228         $('#fecha').addClass('d-none');
229         $('#enfermedad').removeClass('d-none');
230         $(".select2").select2();
231         enfermedades = null;
232         tipo = 2;
233     }else{
234         $('#fecha').addClass('d-none');
235         $('#enfermedad').addClass('d-none');
236     }
237     listaDataTable();
238 });
239
240 $('#fecha').change(function() {
241     fecha = $(this).val();
242     tipo = 1;
243     listaDataTable();
244 });
245
246 $('#enfermedades').change(function() {
247     enfermedades = $(this).val();
248     tipo = 2;
249     listaDataTable();
250 });
251 </script>
252 @endsection

```

```

118         "destroy":true,
119         "ajax":{
120             "method":"POST",
121             "url": "{ route('tratamiento.user.lista') }",
122             "data": {
123                 id_usuario: id_usuario,
124                 fecha: fecha,
125                 id_enfermedad: enfermedades,
126                 tipo: tipo
127             }
128         },
129         "idDisplayLength":10,
130         dom: 'BfRtip',
131         buttons: [
132             {
133                 extend: 'excelHtml5',
134                 title: 'Sistema experto - Tratamiento ({nombre_usuario})',
135                 className: 'btn btn-success',
136                 text: '<i class="far fa-file-excel"></i> EXCEL',
137                 exportOptions: { columns: ":not(.no-export)" }
138             },
139             {
140                 extend: 'pdfHtml5',
141                 title: 'Sistema experto - Tratamiento ({nombre_usuario})',
142                 orientation: 'landscape',
143                 className: 'btn btn-danger',
144                 text: '<i class="far fa-file-pdf"></i> PDF',
145                 exportOptions: { columns: ":not(.no-export)" }
146             },
147         ],
148     });
149 }
150
151 function formato(texto){
152     return texto.replace(/(\d{4})-(\d{2})-(\d{2})$/g, '$3/$2/$1');
153 }
154
155 var params = new window.URLSearchParams(window.location.search);
156 var id_usuario = params.get('id_user')
157 if(id_usuario){
158     consultarUsuario();
159 }
160
161 function deleteRegister(id){
162     Swal.fire({
163         title: 'Esta seguro de eliminar este registro?',
164         text: "",
165         icon: 'warning',
166         showCancelButton: true,
167         confirmButtonColor: '#3085d6',
168         cancelButtonColor: '#d33',
169         confirmButtonText: 'Confirmar',
170         cancelButtonText: 'Cancelar'
171     }).then((result) => {
172         if (result.value) {
173             $.ajax({
174                 url: "{{URL::to('/tratamiento-user')}}" + "/" + id,
175                 type: 'DELETE'
176             })
177             .done(function(res) {
178                 if(res.result){
179                     lista();
180                     Swal.fire(res.message, "", "success");
181                 }else{
182                     Swal.fire("Ocurrio un error", "", "error");
183                 }
184             });
185         }
186     })
187 }
188 }

```

## IMPLEMENTACIÓN

En la siguiente figura se presenta la interfaz de este requerimiento, la cual fue definida anteriormente para proceder con el desarrollo.

Figura\_ 86: Implementación de Gestión de tratamientos

Inicio

Pacientes

Médicos

Síntomas

Medicina

Tipo de enfermedad

Enfermedades

Tratamiento por usuario

Tratamiento General

Diagnostico

Síntomas Paciente

72694847

Diego Simon Samaniego

Nuevo Tratamiento

Todos los tratamientos

EXCEL PDF

BUSCAR:

#	Enfermedad	Fecha	Acciones
1	Covid 19	03/12/2021	
2	Hepatitis	07/12/2021	
3	Covid 19	10/12/2021	
4	Resfriado	10/12/2021	
5	Hepatitis	29/12/2021	

5 de 5 registros

Anterior 1 Siguiete

Sistema Experto ©

https://diagnostico-pulmonares.com/user/paciente



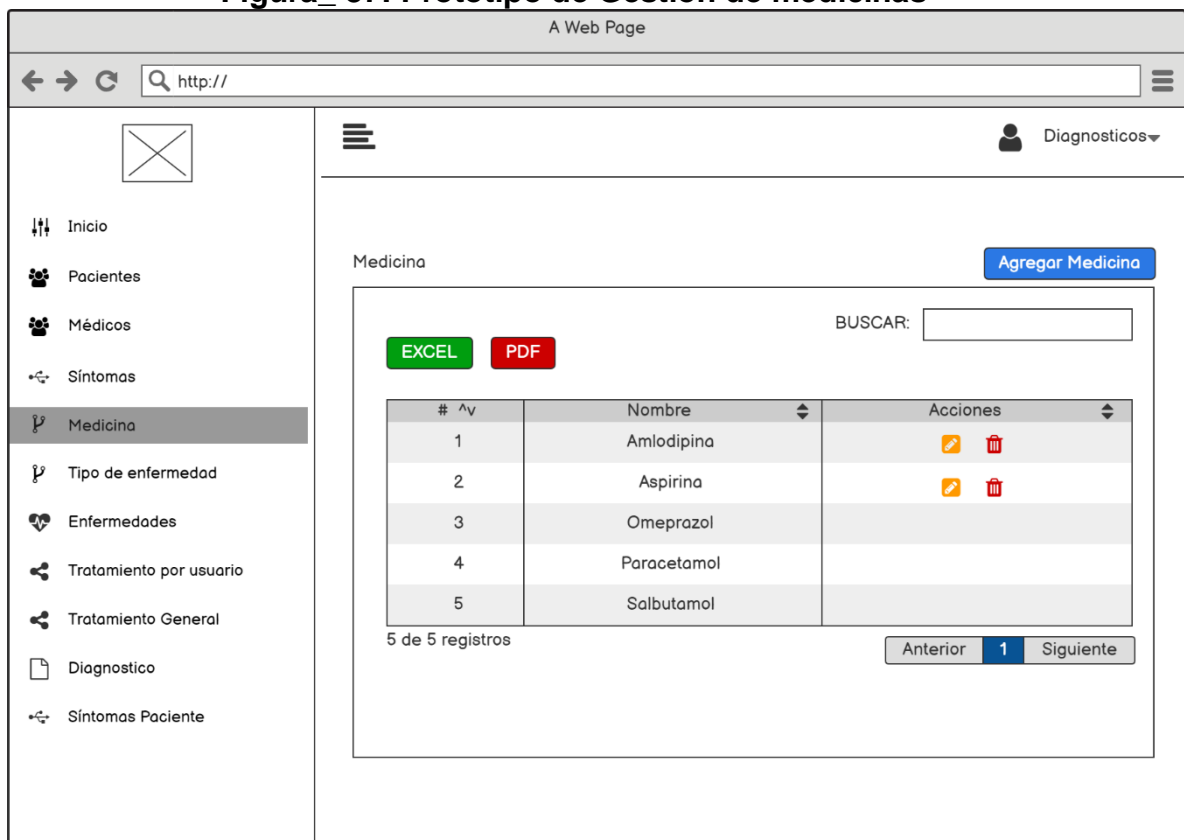
RF14: El sistema permitirá gestionar la lista de medicinas.

## DISEÑO

### PROTOTIPO

De la misma manera a continuación se muestra el prototipo desarrollado para este requerimiento, la cual fue presentado mediante el producto owner para poder continuar con la parte del desarrollo, el programa en el que se desarrollo fue el Balsamiq mockups.

Figura\_ 87: Prototipo de Gestión de medicinas



## DESARROLLO

En las siguientes figuras se pueden observar la parte del desarrollo del sistema, la cual va comenzando por el controlador, modelo y vista.

Figura\_ 88: Controlador de Gestión de medicinas

```

1  }?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6  use App\Models\Medicina;
7  use App\Models\TratamientoDetalle;
8  use App\Models\EnfermedadTratamientoDetalle;
9  use Alert;
10
11 class MedicinaController extends Controller
12 {
13     public function index()
14     {
15         return view('medicina.index');
16     }
17
18     public function lista(Request $request)
19     {
20         $medicina = Medicina::orderBy('nombre','asc')->get();
21
22         $data = collect();
23         foreach ($medicina as $key => $item) {
24             $route_edit = route('medicina.edit',$item->id);
25             $acciones = '<a href="'.$route_edit.'" class="btn btn-warning data-toggle="tooltip" title="Editar"><i
26                 class="fas fa-pen"></i></a>';
27             $acciones .= '<button type="button" class="btn btn-danger data-toggle="tooltip" title="Eliminar" onclick="deleteRegister
28                 ($item->id) "><i class="fas fa-trash-alt"></i></button>';
29
30             $data_Medicina = array(
31                 ($key->id),
32                 $item->nombre,
33                 $acciones
34             );
35             $data->push($data_Medicina);
36         }
37         $results = array(
38             "draw" => 0,
39             "recordsTotal" => count($data),
40             "recordsFiltered" => count($data),
41             "data" => $data
42         );
43         return response()->json($results);
44     }
45
46     public function create()
47     {
48         return view('medicina.create');
49     }
50
51     public function store(Request $request)
52     {
53         Medicina::create($request->all());
54         Alert::success('Medicina agregado', 'Registro guardado correctamente')->autoclose(5000);
55         return redirect(route('medicina.index'));
56     }
57
58     public function edit($id)
59     {
60         $medicina = Medicina::find($id);
61         if($medicina){
62             return view('medicina.edit',compact('medicina'));
63         }else{
64             abort(404);
65         }
66     }

```

```

67     public function update($id, Request $request)
68     {
69         $input = $request->all();
70
71         unset($input['_method']);
72         unset($input['_token']);
73
74         Medicina::where('id',$id)->update($input);
75         Alert::success('Medicina actualizado', 'Registro actualizado correctamente')->autoclose(5000);
76         return redirect(route('medicina.index'));
77     }
78
79     public function destroy($id)
80     {
81         $result = true;
82         $message = "";
83         $tratamiento_detalle = TratamientoDetalle::where('id_medicina',$id)->first();
84         $enfermedad_tratamiento_detalle = EnfermedadTratamientoDetalle::where('id_medicina',$id)->first();
85         if($tratamiento_detalle || $enfermedad_tratamiento_detalle){
86             $result = false;
87             $message = "Medicina en uso";
88         }else{
89             Medicina::where('id',$id)->delete();
90             $message = "Medicina eliminado correctamente";
91         }
92         return response()->json(
93             array(
94                 'result' => $result,
95                 'message' => $message
96             )
97         );
98     }
99
100 }

```

Figura\_ 89: Modelo de Gestión de medicinas

```
Medicina.php x
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7
8 class Medicina extends Model
9 {
10     use HasFactory;
11     public $table = 'medicina';
12     public $timestamps = false;
13
14     public $fillable = [
15         'nombre'
16     ];
17 }
18
```

Figura\_ 90: Vista de Gestión de medicinas

```
index.blade.php x
1 @extends('layouts.app')
2 @section('content')
3 @include('sweetalert:alert')
4 <div class="row mb-2">
5 <div class="col-auto d-none d-sm-block">
6 <strong>Medicina</strong>
7 </div>
8
9 <div class="col-auto ml-auto text-end mt-1">
10 <a href="{{ route('medicina.create') }}" class="btn btn-primary">Agregar Medicina</a>
11 </div>
12 </div>
13 <div class="card">
14 <div class="card-body">
15 <table id="medicinas" class="table table-striped text-center" style="width:100%">
16 <thead>
17 <tr>
18 <th></th>
19 <th>Nombre</th>
20 <th class="no-export">Acciones</th>
21 </tr>
22 </thead>
23 </table>
24 </div>
25 </div>
26 @endsection
27
28 @section('js')
29 <script>
30     var table = $('#medicinas').dataTable( {
31         "language": {
32             "lengthMenu": "Ver los {MENU} primeros registros",
33             "info": "_END_ de _TOTAL_ registros",
34             "infoEmpty": "No se encontraron registros",
35             "infoFiltered": "(Filtro de _PAG_ total entradas)",
36             "loadingRecords": "Cargando...",
37             "processing": "Procesando...",
38             "search": "BUSCAR:",
39             "zeroRecords": "No se encontraron resultados",
40             "emptyTable": "Ningún dato disponible en esta tabla",
41             "paginate": {
42                 "first": "Primero",
43                 "last": "Último",
44                 "next": "Siguiente",
45                 "previous": "Anterior"
46             },
47             "infoCallback": null
48         },
49         "processing": true,
50         "serverSide": true,
51         "destroy": true,
52         "ajax": {
53             "method": "GET",
54             "url": "{{ route('medicina.lista') }}"
55         },
56         "iDisplayLength": 10,
57         dom: 'BtFtIP',
58         buttons: [
59             {
60                 extend: 'excelHtml5',
61                 title: 'Sistema experto - Medicinas',
62                 className: 'btn btn-success',
63                 text: '<i class="far fa-file-excel"></i> EXCEL',
64                 exportOptions: { columns: "not(.no-export)" }
65             }
66         ],
67     });
68 </script>
69 </div>
```

```

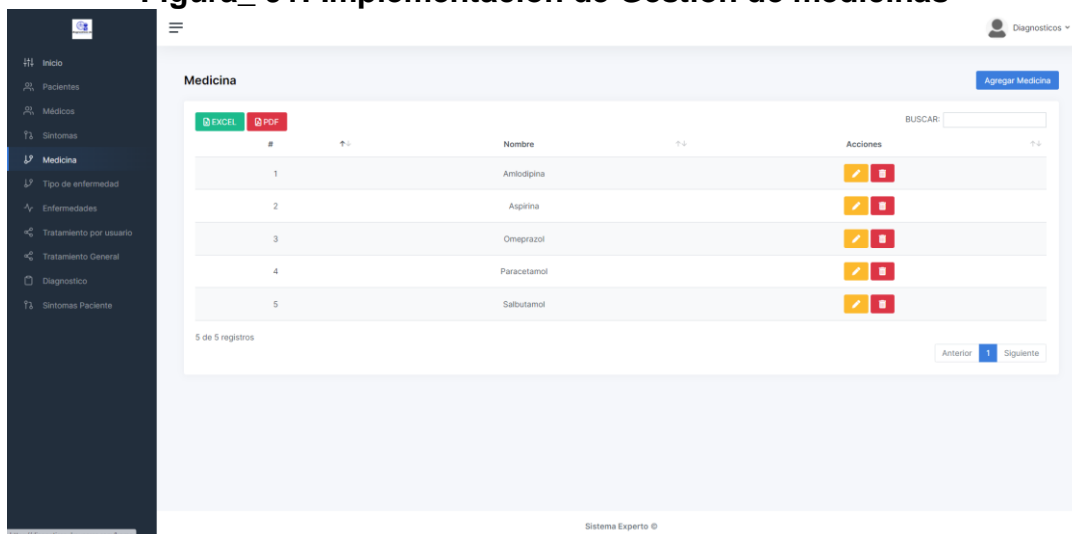
index.blade.php
62     className: 'btn btn-success',
63     text: ' <i class="fa fa-file-excel"></i> EXCEL',
64     exportOptions: { columns: ":not(.no-export)" }
65   },
66   {
67     extend: 'pdfhtml5',
68     title: 'Sistema experto - Medicinas',
69     orientation: 'landscape',
70     className: 'btn btn-danger',
71     text: ' <i class="fa fa-file-pdf"></i> PDF',
72     exportOptions: { columns: ":not(.no-export)" }
73   },
74   ],
75 });
76
77 function deleteRegister(id){
78   Swal.fire({
79     title: 'Esta seguro de eliminar este registro?',
80     text: '',
81     icon: 'warning',
82     showCancelButton: true,
83     confirmButtonColor: '#3085d6',
84     cancelButtonColor: '#d33',
85     confirmButtonText: 'Confirmar',
86     cancelButtonText: 'Cancelar'
87   }).then((result) => {
88     if (result.value) {
89       $.ajax({
90         url: "{URL::to('/medicina')}" + "/" + id,
91         type: 'DELETE'
92       })
93       .done(function(res) {
94         if (res.result){
95           table.ajax.reload();
96           Swal.fire(res.message, "", "success");
97         } else {
98           Swal.fire(res.message, "", "error");
99         }
100       });
101     }
102   })
103 }
104
105 </script>
106 @endsection

```

## IMPLEMENTACIÓN

En la siguiente figura se presenta la interfaz de este requerimiento, la cual fue definida anteriormente para proceder con el desarrollo.

**Figura\_ 91: Implementación de Gestión de medicinas**



**RF15: El sistema permitirá registrar los tratamientos para cada enfermedad de igual manera se cargan las medicinas por defecto.**

## DISEÑO PROTOTIPO

De la misma manera a continuación se muestra el prototipo desarrollado para este requerimiento, la cual fue presentado mediante el producto owner para

poder continuar con la parte del desarrollo, el programa en el que se desarrollo fue el Balsamiq mockups.

**Figura\_ 92: Prototipo de Registro de tratamientos**

El prototipo muestra una interfaz de usuario para un sistema experto. En la parte superior, hay un navegador con botones de navegación y un campo de búsqueda. A la izquierda, un menú de navegación muestra opciones como Inicio, Pacientes, Médicos, Síntomas, Enfermedades, Tratamiento y Diagnóstico. El usuario actual es Gabriel Lozada, Médico. El título de la página es 'Lista de tratamiento > Registro de tratamiento' y el paciente es Juan Lopez Perez. Se pueden ingresar la enfermedad (Covid 19) y la fecha. Una sección 'AGREGAR MEDICINA' permite seleccionar la medicina (Paracetamol 500gr), la frecuencia (1 Tableta | 1/2. tablet | 1 Cucharada) y la cantidad total (10). Se puede agregar un detalle de administración: 'Tomarlo con el estómago lleno'. Una tabla muestra los medicamentos agregados:

Item	Medicina	Frecuencia	Cantidad	Acciones
1	Paracetamol 500gb	1 tableta cada 5 horas	10	[Editar] [Eliminar]
2	Desametaxona 500m	1 cucharada cada 8 horas	10	[Editar] [Eliminar]
3	Amoxicilina	1 Inyectada cada 3 días	5	[Editar] [Eliminar]

Debajo de la tabla hay un campo para 'Recomendaciones' y botones para 'Imprimir' y 'Guardar'.

## DESARROLLO

En las siguientes figuras se pueden observar la parte del desarrollo del sistema, la cual va comenzando por el controlador, modelo y vista.

Figura\_93: Controlador de Registro de tratamientos

```
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161

public function tratamientoCreate($id_diagnostico_enfermedad)
{
    $diagnostico_enfermedad = DiagnosticoEnfermedad::find($id_diagnostico_enfermedad);
    $diagnostico = Diagnostico::find($diagnostico_enfermedad->id_diagnostico);
    $enfermedades = Enfermedad::all()->pluck('nombre','id')->prepend('Todos,');
    $medicina = Medicina::all()->pluck('nombre','id')->prepend('Seleccionar,');

    $enfermedad_tratamiento = EnfermedadTratamiento::where('id_diagnostico',$diagnostico->id)
->where('id_enfermedad',$diagnostico_enfermedad->id_enfermedad)
->where('id_usuario',$diagnostico->paciente->id)
->whereDate('fecha',$diagnostico->created_at)
->first();

    if($enfermedad_tratamiento){
        return view('diagnostico.tratamiento-edit',compact('diagnostico_enfermedad','diagnostico','enfermedades','
medicina','enfermedad_tratamiento'));
    }else{
        return view('diagnostico.tratamiento-create',compact('diagnostico','diagnostico_enfermedad','enfermedades','
medicina'));
    }
}

public function tratamientoStore(Request $request)
{
    $enfermedad_tratamiento = EnfermedadTratamiento::create($request->all());

    foreach ($request->data_medicina as $value) {
        EnfermedadTratamientoDetalle::create([
            'id_enfermedad_tratamiento' => $enfermedad_tratamiento->id,
            'id_medicina' => $value['id_medicina'],
            'frecuencia' => $value['frecuencia'],
            'cantidad' => $value['cantidad'],
            'detalles' => $value['detalles']
        ]);
    }

    Alert::success('Tratamiento agregado correctamente', '')->autoclose(5000);

    return response()->json(
        array(
            'result' => true,
            'data' => $enfermedad_tratamiento->id
        )
    );
}

public function tratamientoUpdate($id,Request $request)
{
    EnfermedadTratamiento::where('id',$id)->update($request->all());

    return response()->json(
        array(
            'result' => true,
            'message' => "Tratamiento actualizado correctamente"
        )
    );
}
}
```

Figura\_ 94: Modelo de Registro de tratamientos

```
EnfermedadTratamiento.php x
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7 use Illuminate\Database\Eloquent\SoftDeletes;
8
9 class EnfermedadTratamiento extends Model
10 {
11     use HasFactory,SoftDeletes;
12
13     public $table = 'enfermedad_tratamiento';
14     protected $dates = ['deleted_at'];
15
16     public $fillable = [
17         'id_diagnostico',
18         'id_enfermedad',
19         'id_usuario',
20         'fecha',
21         'recomendaciones'
22     ];
23
24     function diagnostico(){
25         return $this->belongsTo('App\Models\Diagnostico','id_diagnostico','id');
26     }
27
28     function enfermedad(){
29         return $this->belongsTo('App\Models\Enfermedad','id_enfermedad','id');
30     }
31
32     function usuario(){
33         return $this->belongsTo('App\Models\User','id_usuario','id');
34     }
35 }
36
```

Figura\_ 95: Vista de Registro de tratamientos

```

tratamiento-create.blade.php x
1 @extends('layouts.app')
2
3 @section('content')
4 <div class="row mb-2 mb-xl-3">
5 <div class="col-auto d-none d-sm-block">
6 <h3><strong>Registro de tratamientos</strong></h3>
7 </div>
8 </div>
9 <div class="card">
10 <div class="card-body">
11 <form id="form-tratamiento">
12 <div class="mb-3 row">
13 <h4 class="card-title">Paciente: {{ $diagnostico->paciente->nombr
14 " . $diagnostico->paciente->apellidos }}</h4>
15 </div>
16 <div class="mb-3 row">
17 <label class="col-form-label col-6 col-sm-2">Enfermedad:</label>
18 <label class="col-form-label col-6 col-sm-7 text-end text-sm-start">{{
19 $diagnostico_enfermedad->enfermedad->nombre }}</label>
20 <label class="col-form-label col-6 col-sm-1">Fecha:</label>
21 <label class="col-form-label col-6 col-sm-2 text-end">{{ date('d/m/
22 Y', strtotime($diagnostico->created_at)) }}</label>
23 </div>
24 <div class="mb-3 row">
25 <h4 class="card-title">Agregar Medicina</h4>
26 </div>
27 <div class="mb-3 row">
28 <label class="col-form-label col-sm-2">Medicina</label>
29 <div class="col-sm-10 mb-3">
30 {!! Form::select('medicina',$medicina, null, ['class' => 'form-control select2', 'id' =>
31 'id_medicina']) !!}
32 </div>
33 <label class="col-form-label col-sm-2">Frecuencia</label>
34 <div class="col-sm-5">
35 {!! Form::select('frecuencia',['' => 'seleccionar','1 Tableta','1/2 Tableta','1 Cucharada'], null,
36 ['class' => 'form-control', 'id' => 'frecuencia']) !!}
37 </div>
38 <label class="col-form-label col-sm-2">Repetición</label>
39 <div class="col-sm-5">
40 {!! Form::select('repeticion',['' => 'seleccionar','Una vez al día','cada 5 horas','3 veces al
41 día'], null, ['class' => 'form-control', 'id' => 'repeticion']) !!}
42 </div>
43 <div class="mb-3 row">
44 <label class="col-form-label col-sm-2">Cantidad Total</label>
45 <div class="col-sm-10 mb-3">
46 {!! Form::number('cantidad_total', null, ['class' => 'form-control', 'onkeypress' => 'return
47 validaNumericos(event)', 'id' => 'cantidad_total']) !!}
48 </div>
49 </div>
50 <div class="mb-3 row">
51 <label class="col-form-label col-sm-2">Detalles</label>
52 <div class="col-sm-10 mb-3">
53 {!! Form::textarea('detalles', null, ['class' => 'form-control', 'rows' => '3', 'id' => 'detalles'])
54 !!}
55 </div>
56 </div>
57 <div class="mb-3 row">
58 <div class="col-sm-12 mb-3 text-end">
59 <button type="button" class="btn btn-primary" id="agregar"><i class="fas fa-plus"></i> Agregar</
60 button>
61 </div>
62 </div>
63 <div class="mb-3 row">
64 <div class="table-responsive">
65 <table class="table text-center">
66 <thead>
67 <tr>
68 <th></th>
69 <th>Medicina</th>
70 <th>Frecuencia</th>
71 <th>Cantidad</th>
72 <th>Acciones</th>
73 </tr>

```



```

57 <tbody>
58 <tr>
59 <th>#</th>
60 <th>Medicina</th>
61 <th>Frecuencia</th>
62 <th>Cantidad</th>
63 <th>Acciones</th>
64 </tr>
65 </thead>
66 <tbody id="data_medicina"></tbody>
67 </table>
68 </div>
69 </div>
70 <div class="mb-3">
71 <label class="form-label">Recomendaciones</label>
72 <input type="text" class="form-control" value="" />
73 </div>
74 <div class="mb-3 row">
75 <div class="col-sm-12 d-flex justify-content-between">
76 <a href="{{ route('diagnostico.resultados', diagnostico->id) }}" class="btn btn-primary">
77 <i class="align-middle" data-feather="chevrons-left"></i>
78 Atras
79 </a>
80 <button type="submit" class="btn btn-success">
81 <i class="align-middle" data-feather="save"></i>
82 Guardar
83 </button>
84 </div>
85 </div>
86 </form>
87 </div>
88 </div>
89 @endsection
90 @section('js')
91 <script>
92 let arreglo_medicina = [];
93 consultarTratamientoDetalle();
94
95 function consultarTratamientoDetalle(){
96 let id_usuario = "{{ diagnostico->paciente->id }}";
97 let id_enfermedad = "{{ diagnostico->enfermedad->id }}";
98 let url = "{{ route('enfermedad.tratamiento.consultar.medicina') }}";
99 loading("show");
100 $.post(url, {id_usuario: id_usuario, id_enfermedad: id_enfermedad}, function(res) {
101 loading("hide");
102 if(res.result){
103 arreglo_medicina = res.data;
104 listarMedicinas();
105 }
106 });
107
108
109
110 $("#agregar").click(function() {
111 let id_medicina = $("#id_medicina").val();
112 let frecuencia = $("#frecuencia option:selected").text();
113 let repeticion = $("#repeticion option:selected").text();
114 let cantidad_total = $("#cantidad_total").val();
115 let detalles = $("#detalles").val();
116
117 if(id_medicina && frecuencia && repeticion && cantidad_total){
118 let validar = true;
119 $.each(arreglo_medicina, function(index, val) {
120 if(val.id_medicina == id_medicina){
121 validar=false;
122 }
123 });
124 if(validar){
125 frecuencia = frecuencia + " " + repeticion;
126 let nombre_medicina = $("#id_medicina option:selected").text();
127 let data = {
128 id_medicina: id_medicina,
129 frecuencia: frecuencia,
130 cantidad: cantidad_total,
131 detalles: detalles
132 };
133 $.post(url, data, function(res) {
134 if(res.result){
135 arreglo_medicina = arreglo_medicina.concat(data);
136 listarMedicinas();
137 }
138 });
139 }
140 });
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

```

115 let detalles = $('#detalles').val();
116
117
118 if(id_medicina && frecuencia && repeticion && cantidad_total){
119     let validar = true;
120     $.each(arreglo_medicina, function(index, val) {
121         if(val.id_medicina == id_medicina){
122             validar=false;
123         }
124     });
125     if(validar){
126         frecuencia = frecuencia+" "+repeticion;
127         let nombre_medicina = $('#id_medicina option:selected').text();
128         let data = {
129             id_medicina: id_medicina,
130             nombre_medicina: nombre_medicina,
131             frecuencia: frecuencia,
132             cantidad: cantidad_total,
133             detalles: detalles
134         };
135         arreglo_medicina.push(data);
136         listarMedicinas();
137         limpiarCampos();
138     }else{
139         Swal.fire("Medicina ya está registrado","", "error");
140     }
141 }else{
142     Swal.fire("Campos incompletos","", "error");
143 }
144 });
145
146 function listarMedicinas()
147 {
148     data="";
149     var c = 1;
150     $.each(arreglo_medicina, function(index, val) {
151         data += '<tr>';
152         data += '<td>${c++}</td>';
153         data += '<td>${val.nombre_medicina}</td>';
154         data += '<td>${val.frecuencia}</td>';
155         data += '<td>${val.cantidad}</td>';
156         data += '<td><button type="button" onclick="editarMedicina(${index})" class="btn btn-warning" data-toggle="tooltip" title="editar"><i class="fas fa-pen"></i></button><button type="button" onclick="deleteMedicina(${index})" class="btn btn-danger"><i class="fas fa-trash-alt"></i></button></td>';
157         data += '</tr>';
158     });
159     $('#data_medicina').html(data);
160 }
161
162 function deleteMedicina(i)
163 {
164     arreglo_medicina.splice(i,1);
165     listarMedicinas();
166     limpiarCampos();
167 }
168
169 function limpiarCampos(){
170     $('#id_medicina').val("");
171     $('#frecuencia').val("");
172     $('#repeticion').val("");
173     $('#cantidad_total').val("");
174     $('#detalles').val("");
175     $('#select2').select2();
176 }
177
178 $('#form-tratamiento').submit(function(e) {
179     e.preventDefault();
180
181     var data_form = $('#form-tratamiento').serializeArray();

```

```

164         arreglo_medicina.splice(1,1);
165         listarMedicinas();
166         limpiarCampos();
167     }
168
169     function limpiarCampos(){
170         $("#id_medicina").val("");
171         $("#frecuencia").val("");
172         $("#repeticion").val("");
173         $("#cantidad_total").val("");
174         $("#detalles").val("");
175         $(".select2").select2();
176     }
177
178     $("#form-tratamiento").submit(function(e) {
179         e.preventDefault();
180
181         var data_form = $(this).serializeArray();
182
183         let id_usuario = "{{ $diagnostico->paciente->id }}";
184         let id_diagnostico = "{{ $diagnostico->id }}";
185         let id_enfermedad = "{{ $diagnostico_enfermedad->id_enfermedad }}";
186         let fecha = "{{ date('Y-m-d',strtotime($diagnostico->created_at)) }}";
187         var data = {
188             id_diagnostico: id_diagnostico,
189             id_enfermedad: id_enfermedad,
190             id_usuario: id_usuario,
191             fecha: fecha,
192             recomendaciones: data_form[5]["value"],
193             data_medicina: arreglo_medicina
194         };
195
196         loading("show");
197         $.post('{{ route('diagnostico.tratamiento.store') }}', data, function(res) {
198             loading("hide");
199             if(res.result){
200                 location.reload();
201             }else{
202                 Swal.fire("Ocurrio un error intentelo mas tarde","", "error");
203             }
204         });
205     });
206 </script>
207 @endsection

```

## IMPLEMENTACIÓN

En la siguiente figura se presenta la interfaz de este requerimiento, la cual fue definida anteriormente para proceder con el desarrollo.

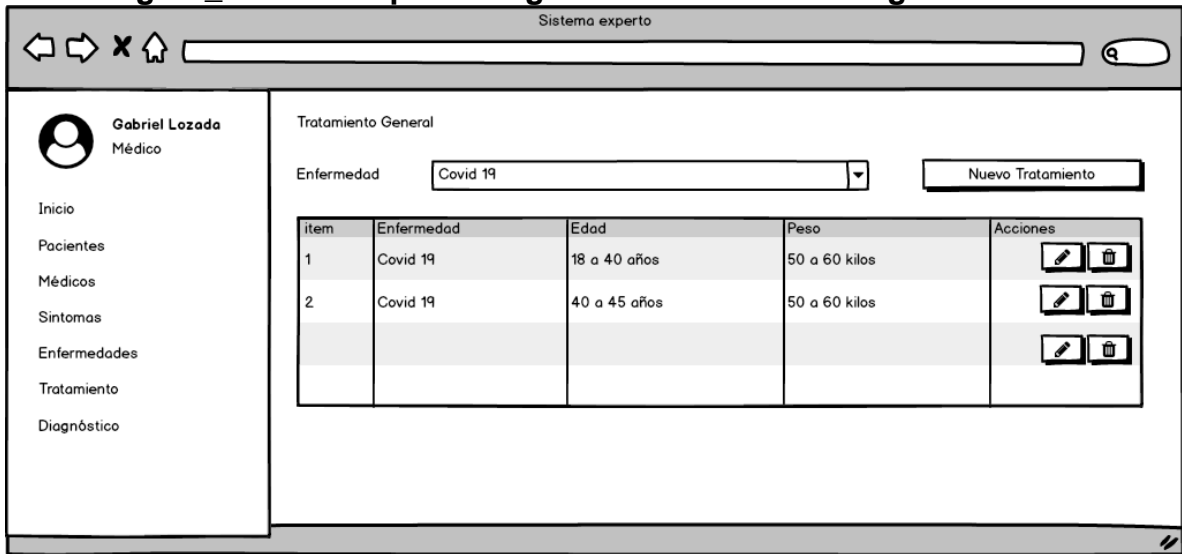
**Figura\_ 96: Implementación de Registro de tratamientos**

**RF16: El sistema permitirá registrar tratamientos generales, respecto a la edad y peso de las personas, para de esta manera se le asigne automáticamente.**

## DISEÑO PROTOTIPO

De la misma manera a continuación se muestra el prototipo desarrollado para este requerimiento, la cual fue presentado mediante el producto owner para poder continuar con la parte del desarrollo, el programa en el que se desarrollo fue el Balsamiq mockups.

**Figura\_ 97: Prototipo de Registros de tratamientos generales**



Sistema experto

← → × ↶

🔍

**Gabriel Lozada**  
Médico

- Inicio
- Pacientes
- Médicos
- Sintomas
- Enfermedades
- Tratamiento
- Diagnóstico

### Tratamiento General

Enfermedad:

Rango de edad:

Rango de peso:

**AGREGAR MEDICINA**

Medicina:

Frecuencia:

Cantidad Total:

Detalles:

item	Medicina	Frecuencia	Cantidad	Acciones
1	Paracetamol 500gb	1 tableta cada 5 horas	10	<input type="button" value="✎"/> <input type="button" value="🗑"/>
2	Desametaxona 500m	1 cucharada cada 8 horas	10	<input type="button" value="✎"/> <input type="button" value="🗑"/>
3	Amoxicilina	1 Inyectada cada 3 días	5	<input type="button" value="✎"/> <input type="button" value="🗑"/>

Recomendaciones:

## DESARROLLO

En las siguientes figuras se pueden observar la parte del desarrollo del sistema, la cual va comenzando por el controlador, modelo y vista.

Figura\_ 98: Controlador de Registros de tratamientos generales

```
TratamientoController.php x
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use App\Models\Enfermedad;
7 use App\Models\Tratamiento;
8 use App\Models\TratamientoDetalle;
9 use App\Models\Medicina;
10 use App\Models\User;
11 use App\Models\EnfermedadTratamiento;
12 use App\Models\EnfermedadTratamientoDetalle;
13 use Alert;
14 use PDF;
15
16 class TratamientoController extends Controller
17 {
18     public function index()
19     {
20         $enfermedades = Enfermedad::all()->pluck('nombre','id')->prepend('Todos,');
21         return view('tratamiento-general.index',compact('enfermedades'));
22     }
23
24     public function lista(Request $request)
25     {
26         if($request->id_enfermedad){
27             $tratamientos = Tratamiento::where('id_enfermedad',$request->id_enfermedad)
28                 ->orderByDesc('created_at')
29                 ->get();
30         }else{
31             $tratamientos = Tratamiento::orderByDesc('created_at')->get();
32         }
33
34         $data = collect();
35         foreach ($tratamientos as $key => $item) {
36             $route_edit = route('tratamiento.edit',$item->id);
37             $acciones = <a href="$route_edit" class="btn btn-warning" data-toggle="tooltip" title="Editar">i
38                 class="fas fa-pen"</i></a>
39             <button type="button" class="btn btn-danger" data-toggle="tooltip" title="Eliminar" onclick="deleteRegister(
40                 '$item->id')">i class="fas fa-trash-alt"</i></button>;
41             $data_tratamiento = array(
42                 ($key->id),
43                 $item->enfermedad->nombre,
44                 $item->edad_min." a ".$item->edad_max." años",
45                 $item->peso_min." a ".$item->peso_max." kilos",
46                 $acciones
47             );
48             $data->push($data_tratamiento);
49         }
50         $results = array(
51             "draw" => 0,
52             "recordsTotal" => count($data),
53             "recordsFiltered" => count($data),
54             "data" => $data
55         );
56         return response()->json($results);
57     }
58
59     public function create()
60     {
61         $enfermedades = Enfermedad::all()->pluck('nombre','id')->prepend('Seleccionar,');
62         $medicina = Medicina::all()->pluck('nombre','id')->prepend('Seleccionar,');
63         return view('tratamiento-general.create',compact('medicina','enfermedades'));
64     }
65 }
```

```
TratamientoController.php x
64
65 public function store(Request $request)
66 {
67     $result = true;
68     $message = "";
69
70     $validar_tratamiento = Tratamiento::where('id_enfermedad',$request->id_enfermedad)->first();
71
72     if($validar_tratamiento){
73         $result = false;
74         $enfermedad = Enfermedad::find($request->id_enfermedad);
75         $message = "Ya existe un tratamiento con la enfermedad ".$enfermedad->nombre;
76     }else{
77         $tratamiento = Tratamiento::create([
78             'id_enfermedad' => $request->id_enfermedad,
79             'edad_min' => $request->edad_min,
80             'edad_max' => $request->edad_max,
81             'peso_min' => $request->peso_min,
82             'peso_max' => $request->peso_max,
83             'recomendaciones' => $request->recomendaciones
84         ]);
85
86         foreach ($request->data_medicina as $value) {
87             TratamientoDetalle::create([
88                 'id_tratamiento' => $tratamiento->id,
89                 'id_medicina' => $value['id_medicina'],
90                 'frecuencia' => $value['frecuencia'],
91                 'cantidad' => $value['cantidad'],
92                 'detalles' => $value['detalles']
93             ]);
94         }
95         Alert::success('Tratamiento agregado', 'Registro guardado correctamente')->autoclose(5000);
96     }
97
98
99     return response()->json(
100         array(
101             'result' => $result,
102             'message' => $message
103         )
104     );
105 }
106
107 public function edit($id)
108 {
109     $tratamiento = Tratamiento::find($id);
110     if($tratamiento){
111         $enfermedades = Enfermedad::all()->pluck('nombre','id')->prepend('Seleccionar','');
112         $medicina = Medicina::all()->pluck('nombre','id')->prepend('Seleccionar','');
113         return view('tratamiento-general.edit',compact('tratamiento','enfermedades','medicina'));
114     }else{
115         abort(404);
116     }
117 }
118 }
```

```

TratamientoController.php x
119 public function update($id, Request $request)
120 {
121     $result = true;
122     $message = "";
123
124     $validar_tratamiento = Tratamiento::where('id_enfermedad', $request->id_enfermedad)
125     ->where('id', '!=', $id)
126     ->first();
127
128     if($validar_tratamiento){
129         $result = false;
130         $enfermedad = Enfermedad::find($request->id_enfermedad);
131         $message = "Ya existe un tratamiento con la enfermedad ".$enfermedad->nombre;
132     }else{
133         Tratamiento::where('id', $id)->update($request->all());
134         Alert::success('Tratamiento actualizado correctamente', '')->autoclose(5000);
135     }
136
137     return response()->json(
138         array(
139             'result' => $result,
140             'message' => $message
141         )
142     );
143 }
144
145 public function listaDetalle($id_tratamiento)
146 {
147     $data = TratamientoDetalle::where('id_tratamiento', $id_tratamiento)
148     ->with('medicina')
149     ->get();
150     return response()->json(
151         array(
152             'result' => true,
153             'data' => $data
154         )
155     );
156 }
157
158 public function agregarDetalle(Request $request)
159 {
160     $result = true;
161     $message = "";
162     $data = null;
163
164     $tratamiento_detalle = TratamientoDetalle::where('id_tratamiento', $request->id_tratamiento)
165     ->where('id_medicina', $request->id_medicina)
166     ->first();
167
168     if($tratamiento_detalle){
169         $result = false;
170         $message = "Medicina ya está registrado";
171     }else{
172         TratamientoDetalle::create($request->all());
173         $message = "Medicina agregado correctamente";
174         $data = TratamientoDetalle::where('id_tratamiento', $request->id_tratamiento)
175         ->with('medicina')
176         ->get();
177     }
178
179     return response()->json(
180         array(
181             'result' => $result,
182             'data' => $data,
183             'message' => $message
184         )
185     );
186 }
187

```



```

188     public function eliminarDetalle($id)
189     {
190         $result = true;
191         $message = "";
192         $data = null;
193
194         $tratamiento_detalle = TratamientoDetalle::where('id',$id)->first();
195         if($tratamiento_detalle){
196             $id_tratamiento = $tratamiento_detalle->id_tratamiento;
197             $tratamiento_detalle->delete();
198             $message = "Medicina eliminado correctamente";
199             $data = TratamientoDetalle::where('id_tratamiento',$id_tratamiento)
200                 ->with('medicina')
201                 ->get();
202         }else{
203             $result = false;
204             $message = "Registro no existe";
205         }
206     }
207
208     return response()->json(
209         array(
210             'result' => $result,
211             'data' => $data,
212             'message' => $message
213         )
214     );
215 }
216
217 public function eliminar($id)
218 {
219     Tratamiento::where('id',$id)->delete();
220     return response()->json(
221         array(
222             'result' => true,
223             'message' => "Tratamiento eliminado correctamente"
224         )
225     );
226 }
227
228 public function indexUser()
229 {
230     return view('tratamiento.index');
231 }
232
233 public function createUser($id_user)
234 {
235     $usuario = User::find($id_user);
236     if($usuario){
237         $enfermedades = Enfermedad::all()->pluck('nombre','id')->prepend('Todos,');
238         $medicina = Medicina::all()->pluck('nombre','id')->prepend('Seleccionar,');
239         return view('tratamiento.create',compact('usuario','enfermedades','medicina'));
240     }else{
241         abort(404);
242     }
243 }
244

```

```

244
245 public function storeUser(Request $request)
246 {
247     $result = true;
248     $message = "";
249
250     $validar_tratamiento = EnfermedadTratamiento::where('id_enfermedad',$request->id_enfermedad)
251         ->whereDate('fecha',$request->fecha)
252         ->where('id_usuario',$request->id_usuario)
253         ->whereNull('id diagnostico')
254         ->first();
255
256     if($validar_tratamiento){
257         $result = false;
258         $enfermedad = Enfermedad::find($request->id_enfermedad);
259         $message = "Ya existe un tratamiento con la enfermedad ".$enfermedad->nombre." en la fecha ".date('d/m/Y',
260             strtotime($request->fecha));
261     }else{
262         $enfermedad_tratamiento = EnfermedadTratamiento::create([
263             'id_enfermedad' => $request->id_enfermedad,
264             'id_usuario' => $request->id_usuario,
265             'fecha' => $request->fecha,
266             'recomendaciones' => $request->recomendaciones
267         ]);
268
269         foreach ($request->data_medicina as $value) {
270             EnfermedadTratamientoDetalle::create([
271                 'id_enfermedad_tratamiento' => $enfermedad_tratamiento->id,
272                 'id_medicina' => $value['id_medicina'],
273                 'frecuencia' => $value['frecuencia'],
274                 'cantidad' => $value['cantidad'],
275                 'detalles' => $value['detalles']
276             ]);
277         }
278         Alert::success('Tratamiento agregado correctamente', '')->autoclose(5000);
279     }
280
281     return response()->json(
282         array(
283             'result' => $result,
284             'message' => $message
285         )
286     );
287 }
288

```

```
TratamientoController.php x
289 public function listaUser(Request $request)
290 {
291     $tratamientos = EnfermedadTratamiento::orderBy('fecha','asc')
292     ->where('id_usuario',$request->id_usuario)
293     ->where(function ($query) use ($request) {
294         if($request->fecha){
295             if($request->tipo == 1){
296                 $query = $query->where('fecha',$request->fecha);
297             }
298         }
299         if($request->id_enfermedad){
300             if($request->tipo == 2){
301                 $query = $query->where('id_enfermedad',$request->id_enfermedad);
302             }
303         }
304     })
305     ->whereNull('id_diagnostico')
306     ->get();
307
308     $data = collect();
309     foreach ($tratamientos as $key => $item) {
310         $route_edit = route('tratamiento.user.edit',$item->id);
311         $acciones='<a href="'.$route_edit.'" class="btn btn-warning" data-toggle="tooltip" title="Editar"><i class="fas fa-pen"></i></a>
312         <button type="button" class="btn btn-danger" data-toggle="tooltip" title="Eliminar" onclick="deleteRegister(
313             '.$item->id.')"><i class="fas fa-trash-alt"></i></button>';
314         $data_tratamiento = array(
315             ($key+1),
316             $item->enfermedad->nombre,
317             date('d/m/Y',strtotime($item->fecha)),
318             $acciones
319         );
320         $data->push($data_tratamiento);
321     }
322     $results = array(
323         "draw" => 0,
324         "recordsTotal" => count($data),
325         "recordsFiltered" => count($data),
326         "data" => $data
327     );
328     return response()->json($results);
329 }
330 }
```

```

330
331 public function filtrosUser(Request $request)
332 {
333     $fechas = EnfermedadTratamiento::select('fecha')
334     ->orderBy('fecha', 'asc')
335     ->where('id_usuario', $request->id_usuario)
336     ->whereNull('id_diagnostico')
337     ->groupBy('fecha')
338     ->get();
339
340     $enfermedades = EnfermedadTratamiento::select('enfermedad_tratamiento.id_enfermedad', 'enfermedad.nombre')
341     ->join('enfermedad', 'enfermedad_tratamiento.id_enfermedad', '=', 'enfermedad.id')
342     ->orderBy('enfermedad.nombre', 'asc')
343     ->where('enfermedad_tratamiento.id_usuario', $request->id_usuario)
344     ->whereNull('enfermedad_tratamiento.id_diagnostico')
345     ->groupBy('enfermedad_tratamiento.id_enfermedad', 'enfermedad.nombre')
346     ->get();
347
348     $data = array(
349         'fechas' => $fechas,
350         'enfermedades' => $enfermedades
351     );
352
353     return response()->json(
354         array(
355             'result' => true,
356             'data' => $data
357         )
358     );
359 }
360
361 public function editUser($id)
362 {
363     $enfermedad_tratamiento = EnfermedadTratamiento::find($id);
364     if($enfermedad_tratamiento){
365         $medicina = Medicina::all()->pluck('nombre', 'id')->prepend('seleccionar,');
366         $enfermedades = Enfermedad::all()->pluck('nombre', 'id')->prepend('Todos,');
367         return view('tratamiento.edit', compact('enfermedad_tratamiento', 'enfermedades', 'medicina'));
368     }else{
369         abort(404);
370     }
371 }
372
373 public function updateUser($id, Request $request)
374 {
375     $result = true;
376     $message = "";
377
378     $validar_tratamiento = EnfermedadTratamiento::where('id_enfermedad', $request->id_enfermedad)
379     ->where('id', '!=', $id)
380     ->whereDate('fecha', $request->fecha)
381     ->where('id_usuario', $request->id_usuario)
382     ->whereNull('id_diagnostico')
383     ->first();
384
385     if($validar_tratamiento){
386         $result = false;
387         $enfermedad = Enfermedad::find($request->id_enfermedad);
388         $message = "Ya existe un tratamiento con la enfermedad ".$enfermedad->nombre." en la fecha ".$date('d/m/Y',
389             strtotime($request->fecha));
390     }else{
391         EnfermedadTratamiento::where('id', $id)->update($request->all());
392         Alert::success('Tratamiento actualizado correctamente', '')->autoClose(5000);
393     }
394     return response()->json(
395         array(
396             'result' => $result,
397             'message' => $message
398         )
399     );
}

```

```

401 public function eliminarUser($id)
402 {
403     EnfermedadTratamiento::where('id',$id)->delete();
404     return response()->json(
405         array(
406             'result' => true,
407             'message' => "Tratamiento eliminado correctamente"
408         )
409     );
410 }
411
412 public function tratamientoDetalle($id_enfermedad_tratamiento)
413 {
414     $data = EnfermedadTratamientoDetalle::select(
415         'enfermedad_tratamiento_detalle.*',
416         'medicina.nombre as nombre_medicina'
417     )
418     ->join('medicina','enfermedad_tratamiento_detalle.id_medicina','medicina.id')
419     ->where('enfermedad_tratamiento_detalle.id_enfermedad_tratamiento',$id_enfermedad_tratamiento)->get();
420     return response()->json(
421         array(
422             'result' => true,
423             'data' => $data
424         )
425     );
426 }
427
428 public function tratamientoDetallestore(Request $request)
429 {
430     EnfermedadTratamientoDetalle::create($request->all());
431     return $this->tratamientoDetalle($request->id_enfermedad_tratamiento);
432 }
433
434 public function deleteTratamientoDetalle($id)
435 {
436     $enfermedad_tratamiento_detalle = EnfermedadTratamientoDetalle::find($id);
437     EnfermedadTratamientoDetalle::where('id',$id)->delete();
438     return $this->tratamientoDetalle($enfermedad_tratamiento_detalle->id_enfermedad_tratamiento);
439 }
440
441
442
443 public function consultarMedicinas(Request $request)
444 {
445     $result = true;
446     $data = null;
447     $user = User::find($request->id_usuario);
448     $edad = $this->calculaEdad($user->fecha_nacimiento);
449
450     $tratamiento = Tratamiento::where('edad_min','<=', $edad)
451     ->where('edad_max','>=', $edad)
452     ->where('peso_min','<=', $user->peso)
453     ->where('peso_max','>=', $user->peso)
454     ->where('id_enfermedad',$request->id_enfermedad)
455     ->first();
456
457     if($tratamiento){
458         $data = TratamientoDetalle::select(
459             'tratamiento_detalle.id_medicina',
460             'medicina.nombre as nombre_medicina',
461             'tratamiento_detalle.frecuencia',
462             'tratamiento_detalle.cantidad',
463             'tratamiento_detalle.detalles'
464         )
465         ->join('medicina','tratamiento_detalle.id_medicina',' ','medicina.id')
466         ->where('tratamiento_detalle.id_tratamiento',$tratamiento->id)
467         ->get();
468     }else{
469         $result = false;
470     }
471
472     return response()->json(
473         array(
474             'result' => $result,
475             'data' => $data
476         )
477     );
478
479     function calculaEdad($fechanacimiento){
480         list($ano,$mes,$dia) = explode("-",$fechanacimiento);
481         $ano_diferencia = date("Y") - $ano;
482         $mes_diferencia = date("m") - $mes;
483         $dia_diferencia = date("d") - $dia;
484         if ($dia_diferencia < 0 || $mes_diferencia < 0)
485             $ano_diferencia--;
486         return $ano_diferencia;
487     }
488
489     public function generarPdf($id_enfermedad_tratamiento)
490     {
491         $enfermedad_tratamiento = EnfermedadTratamiento::find($id_enfermedad_tratamiento);
492         $enfermedad_tratamiento_detalle = EnfermedadTratamientoDetalle::where('id_enfermedad_tratamiento',$id_enfermedad_tratamiento->id)
493         ->get();
494         $pdf = PDF::loadView('pdf.tratamiento',compact('enfermedad_tratamiento','enfermedad_tratamiento_detalle'));
495         return $pdf->stream('Tratamiento '.$enfermedad_tratamiento->enfermedad->nombre.' - '.$enfermedad_tratamiento->usuario->nombr.' '.$enfermedad_tratamiento->usuario->apellidos.'.pdf');
496     }
497 }
498

```

Figura\_ 99: Modelo de Registros de tratamientos generales

```

1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7 use Illuminate\Database\Eloquent\SoftDeletes;
8
9 class Tratamiento extends Model
10 {
11     use HasFactory, SoftDeletes;
12     public $table = 'tratamiento';
13     protected $dates = ['deleted_at'];
14
15     public $fillable = [
16         'id_enfermedad',
17         'edad_min',
18         'edad_max',
19         'peso_min',
20         'peso_max',
21         'recomendaciones'
22     ];
23
24     function enfermedad(){
25         return $this->belongsTo('App\Models\Enfermedad', 'id_enfermedad', 'id');
26     }
27
28 }
    
```

Figura\_ 100: Vista de Registros de tratamientos generales

```

1 @extends('layouts.app')
2 @section('content')
3 @include('sweetalert::alert')
4 <div class="row mb-2 mb-xl-3">
5     <div class="col-auto d-none d-sm-block">
6         <strong>Tratamientos</strong></div>
7     </div>
8 </div>
9 <div class="card">
10     <div class="card-body">
11         <div class="row">
12             <form class="d-sm-inline-block" id="form-paciente">
13                 <div class="col-sm-12 mb-4">
14                     <div class="input-group input-group-navbar">
15                         <input type="text" class="form-control" id="dni" placeholder="Ingrese DNI del paciente"
16                             aria-label="search" maxlength="8" onkeypress="return validaNumericos(event)">
17                         <button class="btn" type="button">
18                             <i class="align-middle" data-feather="search"></i>
19                         </button>
20                     </div>
21                 </div>
22             </form>
23         </div>
24         <div id="resultados" class="row d-none">
25             <div class="col-sm-6 mb-4 pt-2 ps-3">
26                 <strong></strong>
27             </div>
28             <div class="col-sm-4 text-sm-end">
29                 <a href="#" class="btn btn-primary">Nuevo Tratamiento</a>
30             </div>
31         </div>
32         <div class="row">
33             <div class="col-sm-3 mb-4">
34                 <select id="tipo" class="form-select">
35                     <option value="">Todos los tratamientos</option>
36                     <option value="1">por Fecha</option>
37                     <option value="2">por Enfermedad</option>
38                 </select>
39             </div>
40             <div class="col-sm-3 mb-4 fecha d-none">
41                 <select id="fecha" class="form-control select2"></select>
42             </div>
43             <div class="col-sm-3 mb-4 enfermedad d-none">
44                 <select id="enfermedades" class="form-control select2"></select>
45             </div>
46         </div>
47         <div class="table-responsive">
48             <table id="paciente" class="table table-striped text-center" style="width:100%">
49                 <thead>
50                     <tr>
51                         <th></th>
52                         <th>Enfermedad</th>
53                         <th>Fecha</th>
54                         <th class="no-export">Acciones</th>
55                     </tr>
56                 </thead>
57             </table>
58         </div>
59 @endsection
    
```

```

61
62 @section('js')
63 <script>
64     var id_usuario = null;
65     var fecha = null;
66     var enfermedades = null;
67     var tipo = 1;
68     var nombre_usuario = null;
69
70     function lista(){
71         listaDataTable();
72         $.post("{ route('tratamiento.user.filtros') }", {id_usuario: id_usuario}, function(res) {
73             if(res.result){
74                 $('#fecha').empty();
75                 $('#fecha').prepend("<option value>Todas las fechas</option>");
76                 $.each(res.data.fecha, function (i, item) {
77                     $('#fecha').append("<option>", {
78                         value: item.fecha,
79                         text : formato(item.fecha)
80                     });
81                 });
82                 $('#enfermedades').empty();
83                 $('#enfermedades').prepend("<option value>Todas las enfermedades</option>");
84                 $.each(res.data.enfermedades, function (i, item) {
85                     $('#enfermedades').append("<option>", {
86                         value: item.id_enfermedad,
87                         text : item.nombre
88                     });
89                 });
90                 $(".select2").select2();
91             }
92         });
93     }
94
95     function listaDataTable(){
96         $('#paciente').dataTable( {
97             "language": {
98                 "lengthMenu": "Ver los _MENU_ Primeros Registros",
99                 "info": "Elm de _TOTAL_ registros",
100                 "infoEmpty": "No se encontraron registros",
101                 "infoFiltered": "(filtrado de _MAX_ total entradas)",
102                 "loadingRecords": "Cargando...",
103                 "processing": "Procesando...",
104                 "sSearch": "BUSCAR:",
105                 "sZeroRecords": "No se encontraron resultados",
106                 "emptyTable": "Ningún dato disponible en esta tabla",
107                 "oPaginate": {
108                     "sFirst": "Primero",
109                     "sLast": "Último",
110                     "sNext": "Siguiente",
111                     "sPrevious": "Anterior"
112                 },
113                 "fnInfoCallback": null
114             },
115             "aProcessing": true,
116             "aServerSide": true,
117             "destroy": true,
118             "ajax": {
119                 "method": "POST",
120                 "url": "{ route('tratamiento.user.lista') }",
121                 "data": {
122                     id_usuario: id_usuario,
123                     fecha: fecha,
124                     id_enfermedad: enfermedades,
125                     tipo: tipo
126                 }
127             },
128         });

```

```

189
190     $('#form-paciente').submit(function(e) {
191         e.preventDefault();
192         id_usuario = null;
193         consultarUsuario();
194     });
195
196     function consultarUsuario()
197     {
198         let dni = $("#dni").val();
199         loading("show");
200         $.post("{ route('diagnostico.search') }", {dni: dni, id_usuario: id_usuario}, function(res) {
201             loading("hide");
202             if(res.result){
203                 $('#resultados').removeClass('d-none');
204                 nombre_usuario = res.data.nombres + " " + res.data.apellidos;
205                 $('#resultados strong').html(nombre_usuario);
206                 id_usuario = res.data.id;
207                 let url = "{ url: '{ route('tratamiento.user') }' }" + "/" + id_usuario + "/create";
208                 $('#resultados a').attr("href", url);
209                 lista();
210             }else{
211                 $('#resultados').addClass('d-none');
212                 Swal.fire(res.message, 'error');
213             }
214         });
215     }
216
217     $('#tipo').change(function() {
218         let tipo_search = $(this).val();
219         if(tipo_search == 1){
220             $('#fecha').val("");
221             $('#fecha').removeClass('d-none');
222             $('#enfermedad').addClass('d-none');
223             $(".select2").select2();
224             fecha = null;
225             tipo = 1;
226         }else if(tipo_search == 2){
227             $('#enfermedades').val("");
228             $('#fecha').addClass('d-none');
229             $('#enfermedad').removeClass('d-none');
230             $(".select2").select2();
231             enfermedades = null;
232             tipo = 2;
233         }else{
234             $('#fecha').addClass('d-none');
235             $('#enfermedad').addClass('d-none');
236         }
237         listaDataTable();
238     });
239
240     $('#fecha').change(function() {
241         fecha = $(this).val();
242         tipo = 1;
243         listaDataTable();
244     });
245
246     $('#enfermedades').change(function() {
247         enfermedades = $(this).val();
248         tipo = 2;
249         listaDataTable();
250     });
251 </script>
252 @endsection

```

```

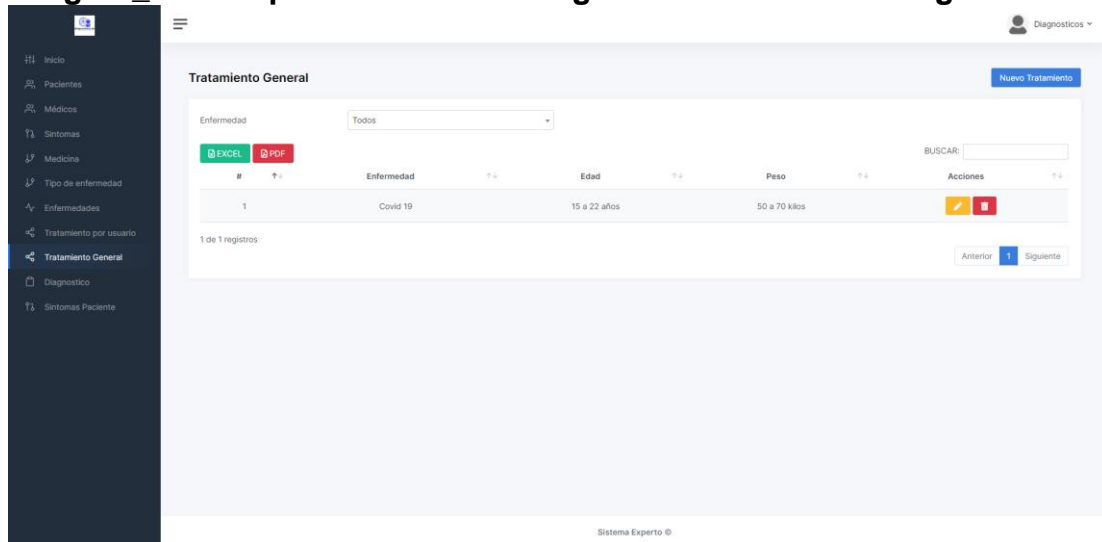
118         "destroy":true,
119         "ajax":{
120             "method":"POST",
121             "url": "{{ route('tratamiento.user.lista') }}",
122             "data": {
123                 id_usuario: id_usuario,
124                 fecha: fecha,
125                 id_enfermedad: enfermedades,
126                 tipo: tipo
127             }
128         },
129         "idDisplayLength":10,
130         dom: 'Bfirtip',
131         buttons: [
132             {
133                 extend: 'excelHtml5',
134                 title: 'Sistema experto - Tratamiento (${nombre_usuario}}',
135                 className: 'btn btn-success',
136                 text: '<i class="far fa-file-excel"></i> EXCEL',
137                 exportOptions: { columns: ":not(.no-export)" }
138             },
139             {
140                 extend: 'pdfHtml5',
141                 title: 'Sistema experto - Tratamiento (${nombre_usuario}}',
142                 orientation: 'landscape',
143                 className: 'btn btn-danger',
144                 text: '<i class="far fa-file-pdf"></i> PDF',
145                 exportOptions: { columns: ":not(.no-export)" }
146             }
147         ],
148     });
149 }
150
151 function formato(texto){
152     return texto.replace(/-(\d{4})-(\d{2})-(\d{2})$/g, '$3/$2/$1');
153 }
154
155 var params = new window.URLSearchParams(window.location.search);
156 var id_usuario = params.get('id_user')
157 if(id_usuario){
158     consultarUsuario();
159 }
160
161 function deleteRegister(id){
162     Swal.fire({
163         title: 'Esta seguro de eliminar este registro?',
164         text: "",
165         icon: 'warning',
166         showCancelButton: true,
167         confirmButtonColor: '#3085d6',
168         cancelButtonColor: '#d33',
169         confirmButtonText: 'Confirmar',
170         cancelButtonText: 'Cancelar'
171     }).then((result) => {
172         if (result.value) {
173             $.ajax({
174                 url: "{{URL::to('tratamiento-user')}}" + "/" + id,
175                 type: 'DELETE'
176             })
177             .done(function(res) {
178                 if(res.result){
179                     lista();
180                     Swal.fire(res.message, "", "success");
181                 }else{
182                     Swal.fire("Ocurrio un error", "", "error");
183                 }
184             });
185         }
186     })
187 }
188 }

```

## IMPLEMENTACIÓN

En la siguiente figura se presenta la interfaz de este requerimiento, la cual fue definida anteriormente para proceder con el desarrollo.

**Figura\_ 101: Implementación de Registros de tratamientos generales**



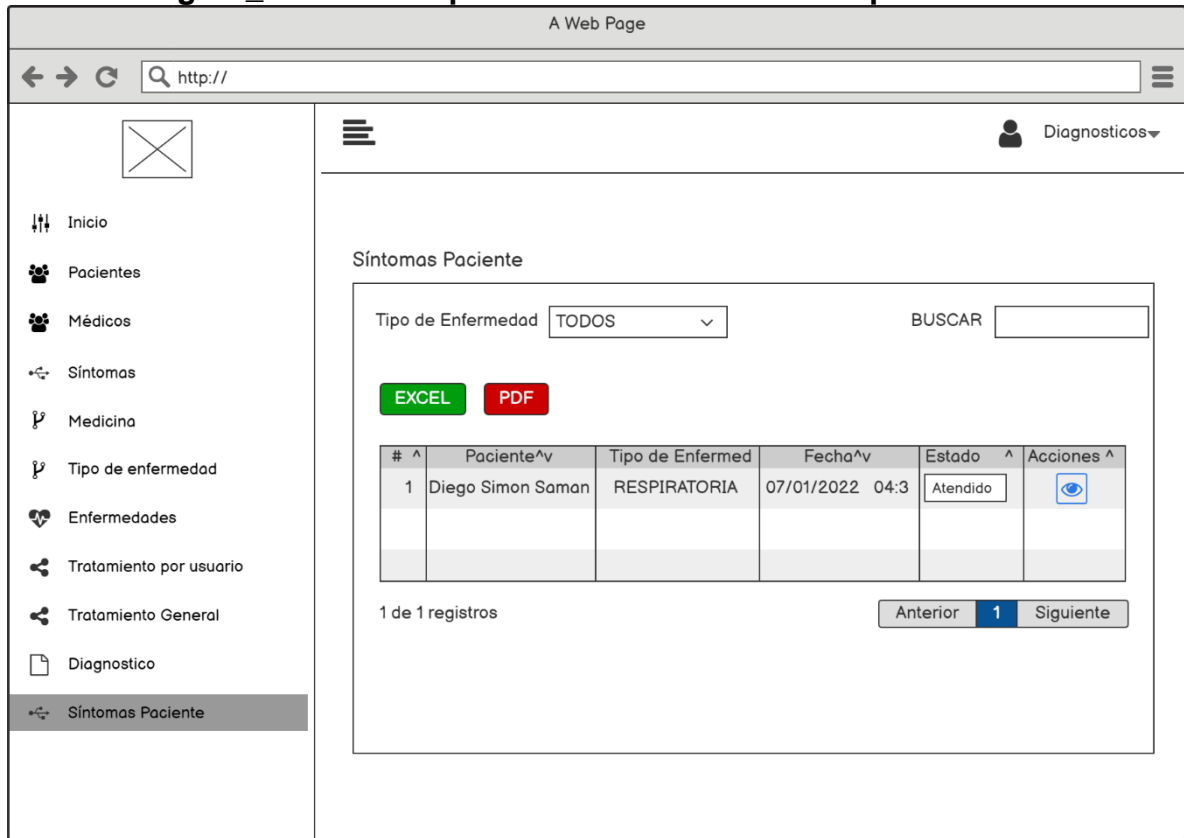
**RF17: El sistema web permitirá gestionar y mostrar los síntomas de cada paciente.**

## **DISEÑO PROTOTIPO**

De la misma manera a continuación se muestra el prototipo desarrollado para este requerimiento, la cual fue presentado mediante el producto owner para poder continuar con la parte del desarrollo, el programa en el que se desarrollo fue el Balsamiq mockups.



**Figura\_ 102: Prototipo de Gestión de Síntomas paciente**



## DESARROLLO

En las siguientes figuras se pueden observar la parte del desarrollo del sistema, la cual va comenzando por el controlador, modelo y vista.

Figura\_ 103: Controlador de Gestión de Síntomas paciente

```
DiagnosticoController.php x
245
246
247 public function indexUsers Sintomas()
248 {
249     $tipo_enfermedad = TipoEnfermedad::all()->pluck('nombre','id')->prepend('TODOS','');
250     return view('sintomas-paciente.index',compact('tipo_enfermedad'));
251 }
252
253 public function listaUsersintomas(Request $request)
254 {
255     $usuario_diagnostico = UsuarioDiagnostico::where(function ($query) use ($request) {
256         if($request->tipo_enfermedad){
257             $query = $query->where('id_tipo_enfermedad',$request->tipo_enfermedad);
258         }
259     })
260     ->get();
261
262     $data = collect();
263     foreach ($usuario_diagnostico as $key => $item) {
264         $route_edit = route('sintomas.paciente.detalle',$item->id);
265         $acciones = '<a href="'.$route_edit.'" class="btn btn-primary" data-toggle="tooltip" title="Ver Síntomas"><i
266             class="fas fa-eye"></i></a>';
267         $estado = "";
268         if($item->estado == 0){
269             $estado = '<span class="badge bg-warning">PENDIENTE</span>';
270         }else if($item->estado == 1){
271             $estado = '<span class="badge bg-success">ATENDIDO</span>';
272         }
273
274         $data_diagnostico_sintomas = array(
275             ($key+1),
276             $item->usuario->nombres." ".$item->usuario->apellidos,
277             $item->tipo_enfermedad->nombre,
278             date('d/m/Y h:i a',strtotime($item->created_at)),
279             $estado,
280             $acciones
281         );
282         $data->push($data_diagnostico_sintomas);
283     }
284     $results = array(
285         "draw" => 0,
286         "recordsTotal" => count($data),
287         "recordsFiltered" => count($data),
288         "data" => $data
289     );
290     return response()->json($results);
291 }
292
293 public function sintomasDetallePaciente($id_usuario_diagnostico)
294 {
295     $usuario_diagnostico = UsuarioDiagnostico::find($id_usuario_diagnostico);
296     return view('sintomas-paciente.create',compact('usuario_diagnostico'));
297 }
```

```
DiagnosticoController.php x
297
298 public function listarUsuarioDiagnosticoSintoma(Request $request)
299 {
300     $usuario_diagnostico = UsuarioDiagnostico::find($request->id_usuario_diagnostico);
301     $usuario_diagnostico_sintomas = UsuarioDiagnosticoSintomas::where('id_usuario_diagnostico',$request->
        id_usuario_diagnostico)->get();
302
303     $data = collect();
304     foreach ($usuario_diagnostico_sintomas as $key => $item) {
305         $acciones=<button type="button" class="btn btn-danger" data-toggle="tooltip" title="Eliminar Síntomas"
            onclick="eliminarSintoma('$item->id.')"><i class="fas fa-trash-alt"></i></button>;
306
307         if($usuario_diagnostico->tipo == 0){
308             $data_diagnostico_sintomas = array(
309                 ($key+1),
310                 $item->sintoma->nombre,
311                 $item->value,
312                 $acciones
313             );
314         }else if($usuario_diagnostico->tipo == 1){
315             $data_diagnostico_sintomas = array(
316                 ($key+1),
317                 $item->sintoma->nombre,
318                 $item->value
319             );
320         }
321         $data->push($data_diagnostico_sintomas);
322     }
323     $results = array(
324         "draw" => 0,
325         "recordsTotal" => count($data),
326         "recordsFiltered" => count($data),
327         "data" => $data
328     );
329
330     return response()->json($results);
331 }
332
333 public function eliminarUsuarioDiagnosticoSintoma($id_usuario_diagnostico_sintomas)
334 {
335     UsuarioDiagnosticoSintomas::where('id',$id_usuario_diagnostico_sintomas)->delete();
336     return response()->json(
337         array(
338             'result' => true,
339             'message' => 'Síntoma eliminado correctamente'
340         )
341     );
342 }
343
```

```

343
344 public function procesarSintomas($id_usuario_diagnostico)
345 {
346     $usuario_diagnostico = UsuarioDiagnostico::find($id_usuario_diagnostico);
347     $usuario_diagnostico_sintomas = UsuarioDiagnosticoSintomas::where('id_usuario_diagnostico',$
348         id_usuario_diagnostico)->get();
349
350     $diagnostico = Diagnostico::create([
351         'id_paciente' => $usuario_diagnostico->id_usuario,
352         'id_medico' => Auth::user()->id,
353         'id_tipo_enfermedad' => $usuario_diagnostico->id_tipo_enfermedad
354     ]);
355
356     $enfermedades = Enfermedad::where('id_tipo_enfermedad',$usuario_diagnostico->id_tipo_enfermedad)->get();
357     foreach ($enfermedades as $value_enfermedad) {
358         $validar = false;
359         $puntaje = 0;
360         $enfermedad_sintomas = EnfermedadSintomas::where('id_enfermedad',$value_enfermedad->id)->get();
361         foreach ($usuario_diagnostico_sintomas as $value_sintomas_input) {
362             foreach ($enfermedad_sintomas as $value_sintomas) {
363                 if($value_sintomas_input->id_sintoma == $value_sintomas->id_sintoma){
364                     $enfermedad_sd = EnfermedadSintomasDetalle::where('id_enfermedad_sintomas',$value_sintomas->id)
365                         ->get();
366                     foreach ($enfermedad_sd as $value_esd) {
367                         if($value_sintomas_input->sintoma->tipo == 1 || $value_sintomas_input->sintoma->tipo == 2){
368                             if($value_sintomas_input->value == $value_esd->nombre){
369                                 $validar = true;
370                                 $puntaje += $value_esd->puntaje;
371                             }
372                         }else if($value_sintomas_input->sintoma->tipo == 3){
373                             $rango = explode(" ", $value_esd->nombre);
374                             if(isset($value_sintomas_input->value)){
375                                 if($rango[1] <= $value_sintomas_input->value && $rango[3] >= $value_sintomas_input->
376                                     value){
377                                     $validar = true;
378                                     $puntaje += $value_esd->puntaje;
379                                 }
380                             }
381                         }
382                     }
383                 }
384             }
385         }
386     }
387     if($validar){
388         DiagnosticoEnfermedad::create([
389             'id_diagnostico' => $diagnostico->id,
390             'id_enfermedad' => $value_enfermedad->id,
391             'probabilidad' => $puntaje
392         ]);
393     }
394
395     UsuarioDiagnostico::where('id',$id_usuario_diagnostico)->update([
396         'estado' => 1
397     ]);
398
399     Alert::success('Síntomas procesado correctamente', '')->autoclose(5000);
400     return redirect(route('sintomas.paciente.detalle',$id_usuario_diagnostico));
}

```

Figura\_ 104: Modelo de Gestión de Síntomas paciente

```

1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Model;
6 use Illuminate\Database\Eloquent\SoftDeletes;
7
8 class UsuarioDiagnostico extends Model
9 {
10     use SoftDeletes;
11     public $table = 'usuario_diagnostico';
12     protected $dates = ['deleted_at'];
13
14     public $fillable = [
15         'id_usuario',
16         'id_tipo_enfermedad',
17         'estado'
18     ];
19
20     function usuario(){
21         return $this->belongsTo('App\Models\User','id_usuario','id');
22     }
23     function tipo_enfermedad(){
24         return $this->belongsTo('App\Models\TipoEnfermedad','id_tipo_enfermedad','id');
25     }
26 }
27

```

Figura\_ 105: Vista de Gestión de Síntomas paciente

```

1  @extends('layouts.app')
2  @section('content')
3  @include('sweetalert:alert')
4  <div class="row mb-2 mb-xl-3">
5    <div class="col-auto d-none d-sm-block">
6      <h3><strong>Síntomas Paciente</strong></h3>
7    </div>
8  </div>
9  <div class="card">
10   <div class="card-body">
11     <div class="row">
12       <label class="col-form-label col-sm-2">Tipo de Enfermedad</label>
13       <div class="col-sm-3 mb-4">
14         {!! Form::select('tipo',$tipo_enfermedad, null, ['class' => 'form-select','id' => 'tipo']) !!}
15       </div>
16     </div>
17     <div class="table-responsive">
18       <table id="sintomas" class="table table-striped text-center" style="width:100%">
19         <thead>
20           <tr>
21             <th>#</th>
22             <th>Paciente</th>
23             <th>Tipo de Enfermedad</th>
24             <th>Fecha</th>
25             <th>Estado</th>
26             <th class="no-export">Acciones</th>
27           </tr>
28         </thead>
29       </table>
30     </div>
31   </div>
32 </div>
33 @endsection
34
35 @section('js')
36 <script>
37   listar();
38
39   $("#tipo").change(function() {
40     listar();
41   });
42
43   function listar(){
44     var tipo_enfermedad = $("#tipo").val();
45
46     $('#sintomas').DataTable( {
47       "language": {
48         "lengthMenu": "Ver los _MENU_ Primeros Registros",
49         "info": " _END_ de _TOTAL_ registros",
50         "infoEmpty": "No se encontraron registros",
51         "infoFiltered": "(filtrado de _MAX_ total entradas)",
52         "loadingRecords": "Cargando...",
53         "processing": "Procesando...",
54         "sSearch": "BUSCAR:",
55         "sZeroRecords": "No se encontraron resultados",
56         "sEmptyTable": "Ningún dato disponible en esta tabla",
57         "oPaginate": {
58           "sFirst": "Primero",
59           "sLast": "Último",
60           "sNext": "siguiente",
61           "sPrevious": "Anterior"
62         },
63         "fnInfoCallback": null
64       },
65       "aProcessing": true,
66       "aServerSide": true,
67       "destroy": true,
68       "ajax": {
69         "method": "POST",
70         "url": "{ route('sintomas.paciente.lista') }",
71         "data": {tipo_enfermedad: tipo_enfermedad}
72       },

```

```

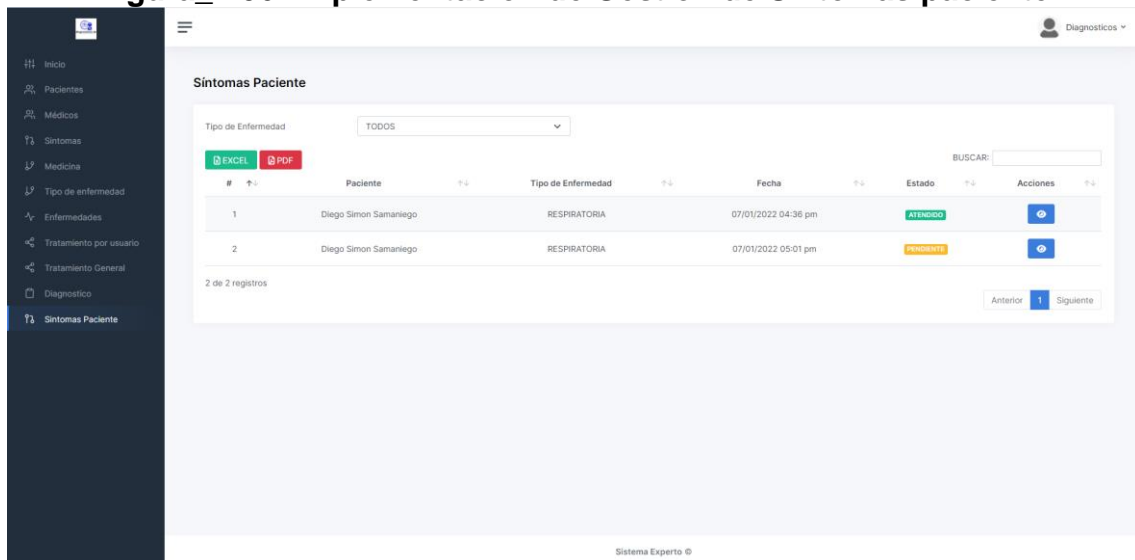
63     "ajax": {
64     },
65     "aProcessing": true,
66     "aServerSide": true,
67     "destroy": true,
68     "ajax": {
69         "method": "POST",
70         "url": "{{ route('sintomas.paciente.lista') }}",
71         "data": {tipo_enfermedad: tipo_enfermedad}
72     },
73     "iDisplayLength": 10,
74     dom: 'Bfrtip',
75     buttons: [
76     {
77         extend: 'excelHtml5',
78         title: 'Sistema experto - Sintomas Paciente',
79         className: 'btn btn-success',
80         text: '<i class="far fa-file-excel"></i> EXCEL',
81         exportOptions: { columns: ":not(.no-export)" }
82     },
83     {
84         extend: 'pdfHtml5',
85         title: 'Sistema experto - Sintomas Paciente',
86         orientation: 'landscape',
87         className: 'btn btn-danger',
88         text: '<i class="far fa-file-pdf"></i> PDF',
89         exportOptions: { columns: ":not(.no-export)" }
90     },
91     ],
92     });
93 }
94 </script>
95 @endsection

```

## IMPLEMENTACIÓN

En la siguiente figura se presenta la interfaz de este requerimiento, la cual fue definida anteriormente para proceder con el desarrollo.

**Figura\_ 106: Implementación de Gestión de Síntomas paciente**

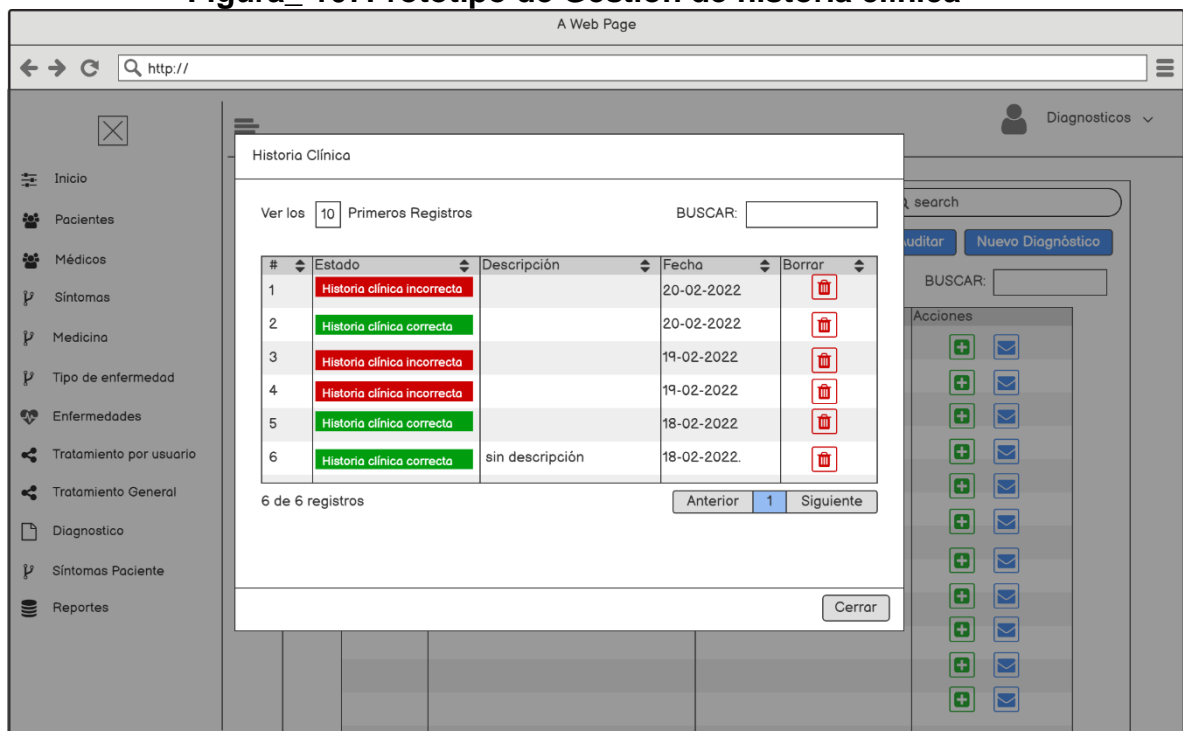


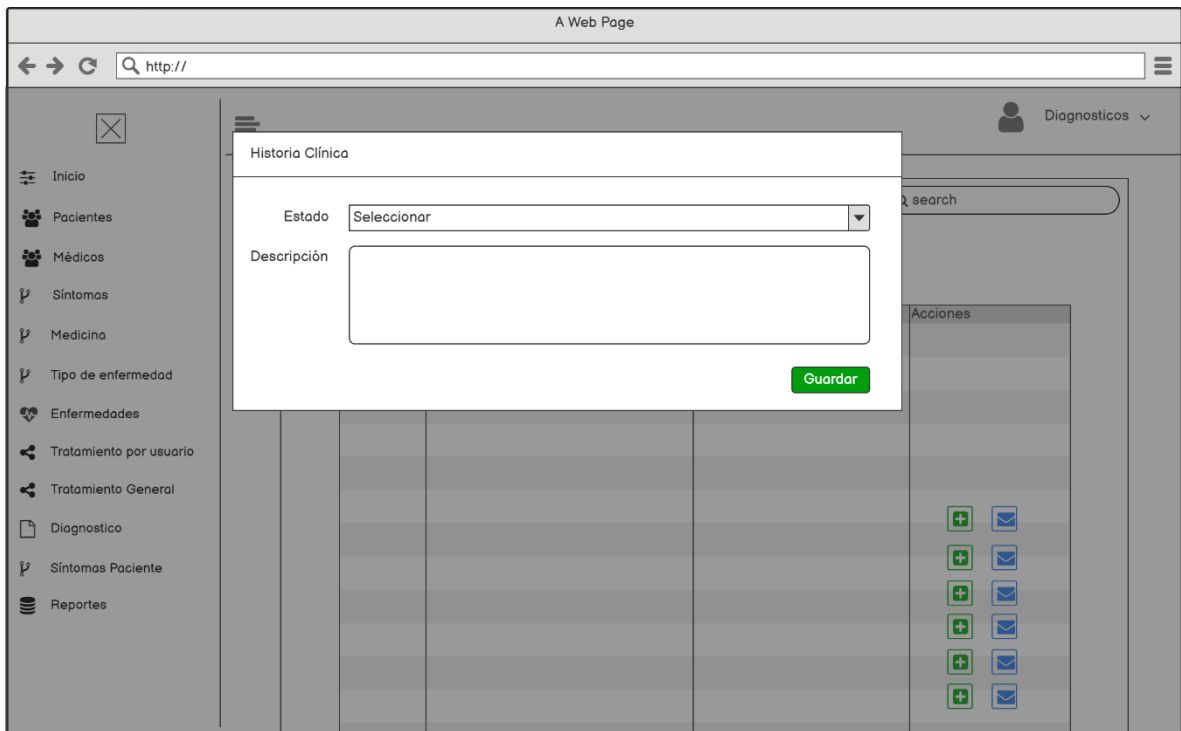
**RF18: El sistema web permitirá gestionar y mostrar la historia clínica de cada paciente.**

## DISEÑO PROTOTIPO

De la misma manera a continuación se muestra el prototipo desarrollado para este requerimiento, la cual fue presentado mediante el producto owner para poder continuar con la parte del desarrollo, el programa en el que se desarrollo fue el Balsamiq mockups.

**Figura\_ 107** Prototipo de Gestión de historia clínica





## DESARROLLO

En las siguientes figuras se pueden observar la parte del desarrollo del sistema, la cual va comenzando por el controlador, modelo y vista.



Figura\_108: Controlador de Gestión de historia clínica

```
DiagnosticoController.php x
456
457 public function historiaClinica(Request $request)
458 {
459     $historia_clinica = HistoriaClinica::where('id_usuario',$request->id_usuario)
460     ->orderByDesc('id')
461     ->get();
462
463     $data = collect();
464
465     foreach ($historia_clinica as $key => $value) {
466         $boton = '<button type="button" class="btn btn-danger"
467             onclick="delete_historia_clinica('.$value->id.')"><i class="fas
468             fa-trash-alt"></i></button>';
469         $data_historia_clinica = array(
470             ($key*1),
471             ($value->estado == 1) ? '<div class="badge bg-success">Historia clínica
472             correcta</div>' : '<div class="badge bg-danger">Historia clínica
473             incorrecta</div>',
474             $value->descripcion,
475             date('d-m-Y',strtotime($value->created_at)) ,
476             $boton
477         );
478         $data->push($data_historia_clinica);
479     }
480
481     $results = array(
482         "draw" => 0,
483         "recordsTotal" => count($data),
484         "recordsFiltered" => count($data),
485         "data" => $data
486     );
487
488     return response()->json($results);
489 }
490
491 public function historiaClinicaStore(Request $request)
492 {
493     HistoriaClinica::create($request->all());
494     return response()->json(
495         array(
496             'result' => true,
497             'message' => "Historia Clínica registrado correctamente"
498         )
499     );
500 }
501
502 public function historiaClinicaEliminar($id)
503 {
504     HistoriaClinica::where('id',$id)->delete();
505     return response()->json(
506         array(
507             'result' => true,
508             'message' => "Historia Clínica borrado correctamente"
509         )
510     );
511 }
512 }
```

Figura\_ 109: Modelo de Gestión de historia clínica

```
HistoriaClinica.php x
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7 use Illuminate\Database\Eloquent\SoftDeletes;
8
9 class HistoriaClinica extends Model
10 {
11     use HasFactory,SoftDeletes;
12
13     public $table = 'historia_clinica';
14     protected $dates = ['deleted_at'];
15
16     public $fillable = [
17         'id_usuario',
18         'estado',
19         'descripcion'
20     ];
21
22     function paciente(){
23         return $this->belongsTo('App\Models\User','id_usuario','id');
24     }
25 }
26
```

Figura\_ 110: Vista de Gestión de historia clínica

```
index.blade.php
47
48 <div class="modal fade" id="ver-auditoria" tabindex="-1" role="dialog" aria-hidden="true">
49   <div class="modal-dialog modal-dialog-centered modal-lg" role="document">
50     <div class="modal-content">
51       <div class="modal-header">
52         <h5 class="modal-title">Historia Clínica</h5>
53         <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close">
54         </button>
55       </div>
56       <div class="modal-body m-3">
57         <div class="table-responsive">
58           <table id="auditoria" class="table table-striped text-center" style="width:100%">
59             <thead>
60               <tr>
61                 <th>#</th>
62                 <th>Estado</th>
63                 <th>Descripción</th>
64                 <th>Fecha</th>
65                 <th>Borrar</th>
66               </tr>
67             </thead>
68           </table>
69         </div>
70       <div class="modal-footer">
71         <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Cerrar</
72         button>
73       </div>
74     </div>
75   </div>
76
77 <div class="modal fade" id="create-auditoria" tabindex="-1" role="dialog" aria-hidden="true">
78   <div class="modal-dialog modal-dialog-centered modal-lg" role="document">
79     <div class="modal-content">
80       <div class="modal-header">
81         <h5 class="modal-title">Nuevo Historia Clínica</h5>
82         <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close">
83         </button>
84       </div>
85       <form id="historia-clinica">
86         <div class="modal-body m-3">
87           <div class="mb-3 row">
88             <label class="col-form-label col-sm-2 text-sm-end">Estado</label>
89             <div class="col-sm-10">
90               {!! Form::select('estado', ['' => 'Seleccionar', '1' => 'Historia clínica
91               correcta', '0' => 'Historia clínica incorrecta'], null, ['class' =>
92               'form-select', 'data-validation' => 'required', 'id' => 'estado']) !!}
93             </div>
94           </div>
95           <div class="mb-3 row">
96             <label class="col-form-label col-sm-2 text-sm-end">Descripción</label>
97             <div class="col-sm-10">
98               {!! Form::textarea('descripcion', null, ['class' =>
99               'form-control', 'maxlength' => '200', 'rows' => '3', 'id' =>
100              'descripcion']) !!}
101             </div>
102           </div>
103         <div class="modal-footer">
104           <button type="submit" class="btn btn-success">Guardar</button>
105         </div>
106       </form>
107     </div>
108   </div>
109 </div>
```

## IMPLEMENTACIÓN

En la siguiente figura se presenta la interfaz de este requerimiento, la cual fue definida anteriormente para proceder con el desarrollo.

Figura\_ 111: Implementación de Gestión de historia clínica

Historia Clínica

Ver los 10 Primeros Registros      BUSCAR:

#	Estado	Descripción	Fecha	Borrar
1	Historia clínica incorrecta		20-02-2022	
2	Historia clínica correcta		20-02-2022	
3	Historia clínica incorrecta		19-02-2022	
4	Historia clínica incorrecta		19-02-2022	
5	Historia clínica correcta		18-02-2022	
6	Historia clínica correcta	sin descripción	18-02-2022	

6 de 6 registros      Anterior 1 Siguiente

Cerrar

Diagnóstico

72694847      Ver auditoría    Auditar    Nuevo Diagnóstico

Diego Simon Samaniego

Nuevo Historia Clínica

Estado: Seleccionar

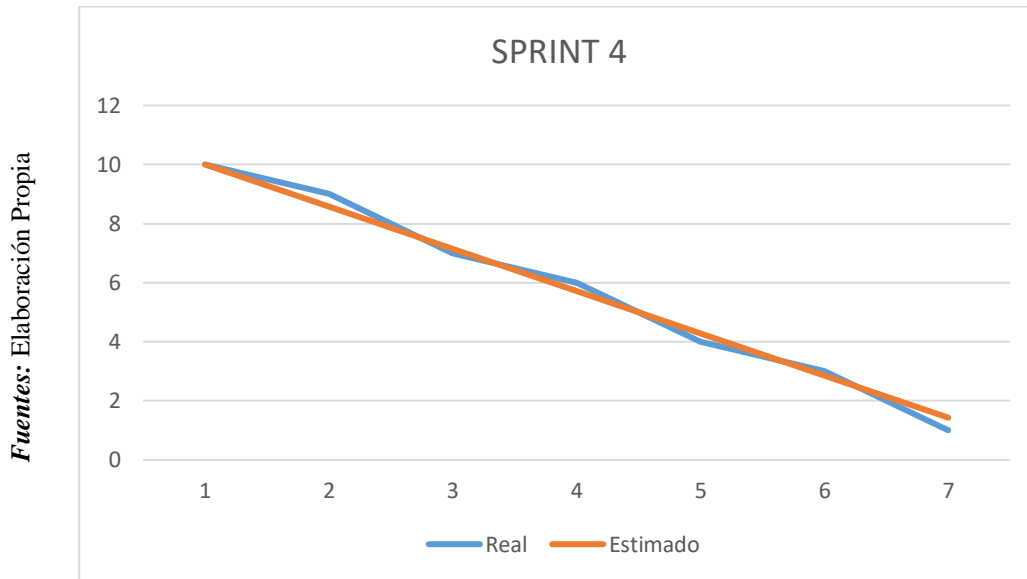
Descripción:

Guardar

5	14/02/2022	Diagnosticos Pulmonares		
6	11/01/2022	Diagnosticos Pulmonares		
7	30/12/2021	Diagnosticos Pulmonares		
8	30/12/2021	Diagnosticos Pulmonares		

## BURNDOWN DEL SPRINT N° 4

Figura\_ 112: Burndown del Sprint 4



### Burndown Sprint 4

A continuación, se presentará el significado de la figura anterior: comenzando con la línea roja la cual representa el tiempo ideal del desarrollo de la misma, así mismo la línea azul representa el tiempo real en la que se desarrolló dicho sprint. Es así como se puede decir que si la línea azul se ubica por debajo de la línea roja es porque existió un adelanto en la elaboración del sprint, y de manera contraria si la línea azul está por encima de la roja significa retraso en la elaboración del sprint.

## ACTA DE REUNION DEL SPRINT N° 4

Siendo las 3 pm del día 21 de noviembre del 2021, se reúne en el Policlínico Monserrat.

Presentes:

ROL	NOMBRE
Scrum Master	Luz Gomez Taype
Team Member	Gianmarco Roger Nizama Antunez
Product Owner	Miguel Rios

El encargado del Policlínico Monserrat, Luz Gomez Taype culmina la reunión con el Sr. Gianmarco Nizama señalando los últimos puntos respecto al sprint.

De esta manera se logra definir la forma de trabajo y todos los requerimientos respecto a este sprint, así mismo firmar el acta que contiene la fecha de entrega, esto para el cierre de la reunión.

Los asistentes presentes aprobaron los puntos que fueron acordados en esta reunión, así mismo beneficio la planificación de este Sprint, realizando el compromiso de entregar los resultados referentes al tiempo que fue pactado.


## ACTA DE ENTREGA DEL SPRINT N° 4

Siendo la 1 pm del día 7 de diciembre del 2021 se reúne en el Policlínico Monserrat.

ROL	NOMBRE
Scrum Master	Luz Gomez Taype
Team Member	Gianmarco Roger Nizama Antunez
Product Owner	Miguel Rios

El Sr. Gianmarco Nizama, señalan y detallan cada requerimiento desarrollado, así mismo presenta las interfaces obtenidas del product owner.

Luego de realizar las verificaciones a la sustentación del Sr. Gianmarco Nizama para continuar con la aprobación del Sprint, de manera unánime se decide la aprobación de este sprint, referente al proyecto “Sistema experto para diagnóstico de enfermedades pulmonares en el policlínico Monserrat”.

Es así como los presentes imparten su aprobación al informe del Sr. Gianmarco Nizama, referente al Sprint N° 4 la cual fue concluida, del siguiente proyecto “Sistema experto para diagnóstico de enfermedades pulmonares en el policlínico Monserrat”.


## RESUMEN DE LA REUNIÓN RETROSPECTIVA DE SPRINT N° 4

Información de la empresa y proyecto:

<b>Empresa / Organización</b>	Policlínico Monserrat
<b>Proyecto</b>	“Sistema experto para diagnóstico de enfermedades pulmonares en el policlínico Monserrat”

Información de la reunión:

<b>Lugar</b>	Policlínico Monserrat
<b>Fecha</b>	7/12/2021
<b>Número de iteración / Sprint</b>	Sprint 4
<b>Personas Convocadas a la reunión</b>	
<b>Persona que asistieron a la reunión</b>	

Formulario de reunión retrospectiva

<b>¿Qué salió bien en la Iteración? (Aciertos)</b>	<b>¿Qué no salió bien en la Iteración? (Errores)</b>
<p>Se logró ejecutar la gestión de tratamientos y el registro.</p> <p>Se logró ejecutar los registros de tratamientos generales.</p> <p>Se desarrolló la gestión de medicinas y los síntomas de cada paciente.</p>	<p>Se generaron fallas en el registro de tratamientos, sin embargo, se pudo resolver.</p>