

# ESCUELA DE INGENIERÍA DE TELECOMUNICACIÓN Y ELECTRÓNICA



## TRABAJO DE FIN DE GRADO

### DISEÑO Y DESARROLLO DE UNA APLICACIÓN WEB PARA EL CONTROL Y SEGUIMIENTO DE LA CALIDAD DE VIDA DEL PACIENTE CON VEJIGA HIPERACTIVA

**Titulación:** Grado en Ingeniería en Tecnologías de la Telecomunicación

**Mención:** Sistemas Electrónicos

**Autor:** Raúl Batista Herrera

**Tutor:** Dr. Gustavo Marrero Callicó

**Cotutor:** Dr. Aday García Del Toro

**Fecha:** 26/11/2020



# Índice

---

Resumen .....	9
Abstract .....	11
1. Introducción .....	13
1.1 Motivación .....	13
1.2. Objetivos .....	15
1.3. Estructura del documento .....	16
2. Estado del arte.....	19
2.1. Servicios actuales .....	19
2.2. Tecnologías utilizadas .....	20
3. Análisis de los requisitos.....	29
3.1. Proceso de desarrollo .....	29
3.2. Análisis de Requisitos.....	30
3.2.1. Requisitos funcionales.....	30
4. Normativa de infraestructura y seguridad para aplicaciones del gobierno de canarias.....	33
4.1 Normativa de gestión de alojamiento de aplicaciones.....	33
4.1.1 Normativa de arquitectura .....	33
4.1.2 Normativa de mantenimiento.....	34
4.2 Normativa sobre los requisitos de Seguridad en Aplicaciones Web .....	35
4.3 Requisitos y recomendaciones en el desarrollo y despliegue .....	36
5. Diseño de la aplicación .....	37
5.1. Arquitectura web .....	37
5.1.1. Herramientas de Back-End .....	37
5.1.2. Herramientas de Front-End .....	40
5.1.3. Framework usado .....	41
5.2. Estructura del proyecto .....	44
5.3. Diseño de la encuesta .....	47
5.4. Diseño de la base de datos .....	50
6. Implementación.....	59
6.1 Diagrama de conexión .....	59

6.2 Views .....	60
6.2.1. Index.Blade.php .....	60
6.2.2. Show.Blade.php .....	61
6.2.3. Addnew.Blade.php .....	62
6.3 Controller .....	63
6.4 Models.....	67
6.5 Commands .....	68
7. Pruebas y resultados .....	75
Ventana de la página principal.....	75
Ventana añadir paciente.....	76
Ventana de vista en detalle del paciente.....	77
8. Conclusiones.....	81
8.1. Conclusiones .....	81
8.2. Trabajo futuro .....	82
9. Referencias .....	85
Glosario.....	87
Presupuesto.....	89
P.1 Recursos materiales .....	89
P.2 Recursos hardware .....	89
P.3 Recursos software.....	90
P.4 Recursos humanos .....	90
P.5 Gastos en material adicional.....	92
P.6 Derechos del visado del COIT.....	92
P.7 Aplicación de impuestos .....	92

## Índice de figuras

---

Figura 1 Logotipo Fundación iSYS .....	20
Figura 2 Conexionado entre paciente y personal sanitario. ....	22
Figura 3 Comunicación HTTP .....	23
Figura 4 Modelo en 4 capas. ....	24
Figura 5 Modelo de aplicación Web tradicional .....	25
Figura 6 Modelo con tecnología AJAX.....	26
Figura 7 Modelo Vista Controlador.....	26
Figura 8 Ciclo de vida: Modelo en cascada. ....	29
Figura 9 Servidor Web Apache.....	38
Figura 10 Funcionamiento PHP.....	39
Figura 11 Base de datos MySQL.....	40
Figura 12 Herramientas Frond-End.....	40
Figura 13 Logo Laravel. ....	42
Figura 14 Recorrido de una solicitud GET .....	42
Figura 15 Logo Bootstrap .....	43
Figura 16 Logo jQuery .....	44
Figura 17 Vista principal.....	46
Figura 18 Vista detalle del paciente.....	47
Figura 19 LimeSurvey .....	48
Figura 20 Encuesta mensual primeras preguntas.....	49
Figura 21 Opción desplegable.....	49
Figura 22 Ejemplo de subpreguntas. ....	50
Figura 23 Diagrama de módulos, conexión sanitario-paciente. ....	59
Figura 24 Vista index.blade.PHP. ....	75
Figura 25 Filtro por estado.....	76
Figura 26 Búsqueda por caracteres. ....	76
Figura 27 Vista addnew.blade.PHP. ....	77
Figura 28 Vista show.blade.PHP.....	78
Figura 29 Desplegable encuesta mensual.....	79
Figura 30 Notificación impresa. ....	79

## Índice de tablas

---

Tabla 1 Análisis competitivo (One Drop: Gestión de Diabetes y ACO+)	20
Tabla 2 Stacks más usados	24
Tabla 3 Tipo de Arquitectura DGTNT.	33
Tabla 4 Navegadores que soporta bootstrap.	43
Tabla 5 Navegadores que soportan jQuery.	44
Tabla 6 Estructura de la aplicación.	45
Tabla 7 SP_Paciente	51
Tabla 8 SP_Diario	53
Tabla 9 SP_Mensual	57
Tabla 10 SP_Notification	57
Tabla 11 Costes de las herramientas hardware	90
Tabla 12 Valores de la variable Ct en función de las horas trabajadas	90
Tabla 13 Presupuesto de los costes materiales y tarifado por tiempo empleado	91
Tabla 14 Costes del material adicional	92
Tabla 15 Presupuesto total del Trabajo Fin de Grado	93

## Resumen

---

Con el objetivo de agilizar las consultas en el Complejo Hospitalario Universitario Insular-Materno Infantil (CHUIMI), se propone la realización de una aplicación web para la Unidad de Suelo Pélvico del Servicio de Ginecología, que consistirá en el seguimiento de la calidad de vida del paciente para prevenir y tratar problemas en la vejiga, en concreto la afección de la vejiga hiperactiva (VHA).

Con el desarrollo de esta nueva aplicación web se pretende mejorar la eficiencia de la Unidad de Suelo Pélvico en cuanto a mejora en la calidad de vida de las pacientes a estudio, su seguimiento y control de la adherencia al tratamiento. Mediante el uso de la app se evitarán desplazamientos innecesarios de las pacientes al CHUIMI ya que se podrá realizar el seguimiento de forma remota. Evitar desplazamientos por parte del paciente conlleva un ahorro de costes en el complejo hospitalario debido a la mejora de la eficacia y rentabilidad de las agendas. Mediante su reestructuración, se reserva la presencia física a los diagnósticos complejos o evoluciones tórpidas, potenciando la cita telefónica a los pacientes de menor complejidad y evoluciones favorables. Además, con el empleo de una base de datos se obtendrá una documentación fiable y organizada que se podrá usar en investigaciones de cara al futuro.

El procedimiento para la recopilación de datos del paciente desde el aplicativo, que es donde se centrará el proyecto, se estructura de la siguiente forma: desde la consulta, el paciente recibirá un código numérico anonimizado no vinculante con el que podrá ser partícipe del estudio de forma anónima desde cualquier zona con conexión a internet a una aplicación web. Desde el aplicativo será redireccionado a un formulario que deberá rellenar cuestiones estandarizadas con las que se pretende obtener una serie de datos referentes a la calidad de vida, centrada en el manejo de la vejiga hiperactiva. Una vez realizado el cuestionario, se enviará a la red interna del CHUIMI donde será almacenado en una base de datos. El personal sanitario podrá acceder a estos datos desde otra aplicación web únicamente accesible desde la red hospitalaria. En ella, se podrá hacer el seguimiento telemático del paciente desde cualquier terminal conectado a la red hospitalaria ahorrando así concertar citas con el paciente.

Uno de los objetivos del trabajo, además del seguimiento sanitario, ha sido el desarrollo desde cero de una aplicación web sencilla pero profesional. Se ha hecho un esfuerzo

didáctico para que este documento sirva para futuros desarrollos web similares. Por un lado, se explican las distintas actividades de desarrollo de un producto software y, por otro lado, se aplican estos conocimientos en un entorno de desarrollo en concreto, llamado Laravel, un framework para implementar código PHP de forma simple y eficiente, con una gran influencia de otros como Ruby on Rails y ASP .NET.

## Abstract

---

With the aim of streamlining consultations in the Insular-Maternal-Infant University Hospital Complex (CHUIMI), it is proposed to create a web application for the Pelvic Floor Unit of the Gynecology Service, which will consist of monitoring the quality of life of the patient to prevent and treat bladder problems, specifically overactive bladder condition (OAB).

With the development of this new web application it is intended to improve the efficiency of the Pelvic Floor Unit in terms of improvement in the quality of life of the patients under study, its monitoring and control of adherence to treatment through the use of the app will prevent Unnecessary travel of patients to the CHUIMI since monitoring can be done remotely. Avoiding travel by the patient leads to cost savings in the hospital complex due to the improvement of the effectiveness and profitability of the agendas. Through its restructuring, physical presence is destined to complex diagnoses or torpid evolutions and enhancing the telephone appointment to patients of less complexity and favorable evolutions. In addition, the use of a database will provide reliable and organized documentation that can be used in future research.

The procedure for collecting patient data from the application, which is where the project will focus, is structured as follows: from the consultation, the patient will receive a non-binding anonymized numerical code with which they can be a participant in the study of anonymously from any area with an internet connection to a web application. From the application, you will be redirected to a form that must fill in standardized questions with which it is intended to obtain a series of data referring to quality of life focused on the management of overactive bladder. Once the questionnaire is completed, it will be sent to the CHUIMI's internal network where it will be stored in a database. Healthcare personnel will be able to access this data from another web application only accessible from the hospital network. In it, the patient can be monitored electronically from any terminal connected to the hospital network, thus saving on making appointments with the patient.

One of the objectives of the work, in addition to health monitoring, has been the development from scratch of a simple but professional web application. A didactic effort has been made to make this document useful for future similar web developments. On the one hand, the different development activities of a software product are explained and, on the other hand, this knowledge is applied in a specific development environment, called

Laravel, a framework to implement PHP code in a simple and elegant way, with a heavily influenced by others like Ruby on Rails and ASP .NET.

# 1. Introducción

---

En la actualidad, las aplicaciones web aportan una serie de utilidades renovadas al usuario y aunque parezca sorprendente, debido al continuo avance y desarrollo de las tecnologías de la información y comunicación (TIC), aparecen nuevas necesidades que satisfacer. Esta capacidad de adaptación a la demanda que se produce en cada momento, es lo que hace distinguir a las TICs del resto. Este trabajo de fin de grado aúna los campos de las tecnologías de la información y la medicina, en él se van a exponer los detalles para la creación de una aplicación web con una funcionalidad de control sanitario.

## 1.1 Motivación

La vejiga hiperactiva (VHA) es un diagnóstico clínico que se define como la presencia de urgencia urinaria, acompañada generalmente de un aumento de la frecuencia miccional diurna y nicturia, con o sin incontinencia urinaria de urgencia y en ausencia de una infección del tracto urinario u otra patología previa. Es un trastorno urológico con frecuencia de micción en vigilia > 8 veces/día y frecuencia de micción durante el período de sueño > 3 veces/noche. Se trata de una patología que afecta fundamentalmente a la calidad de vida y a la normal realización de las actividades de la vida diaria, donde la urgencia urinaria (fuerte necesidad de orinar sin advertencia previa) y la incontinencia urinaria de urgencia (incapacidad de llegar al baño a tiempo a orinar y/o pérdidas de orina en forma de gota o chorro durante el día) son los síntomas distintivos [1] .

Es una patología que afecta a 1 de cada 11 adultos y al 12-17% de la población en general de los Estados Unidos. La prevalencia de la VHA aumenta en los caucásicos, blancos y asiáticos de mediana edad y el tratamiento farmacológico de primera línea son los agentes anticolinérgicos [1].

La Unidad de Suelo Pélvico del Servicio de Ginecología del CHUIMI tiene una característica propia, que es el abordaje multidisciplinar, muy raro en nuestro país. Esta atención centrada en la paciente, incluye la actuación de ginecólogos, urólogos, médicos rehabilitadores y personal de enfermería especializado en la patología de Suelo Pélvico. En concreto, el personal de enfermería tiene un papel fundamental en diferentes aspectos de atención en la unidad, básico si hablamos concretamente de incontinencia urinaria. La incontinencia

urinaria relacionada con la urgencia, combina el tratamiento farmacológico con el tratamiento conductual, que a su vez incrementa la adherencia al tratamiento médico.

La Unidad de Suelo Pélvico del Servicio de Ginecología del CHUIMI es pionero en la creación de una consulta de enfermería dirigida a este acompañamiento de la paciente en el arduo camino del tratamiento individualizado, en el que cada paciente recibe un seguimiento en función de su mejoría clínica o ausencia de ella. A su vez, se da el tiempo necesario en consulta para explicar y confirmar que se ha entendido la información. Y ese tiempo, además de que no puede estar en el contexto de una consulta médica, se vuelve más eficiente si la persona responsable pertenece al personal de enfermería.

Según los reportes dados por el equipo de enfermería, el número de veces que la paciente acude a consulta de enfermería es variable, y el contenido de la cita es básicamente una entrevista en la que la enfermera recoge los datos clínicos, revisa con la paciente un formulario llamado diario miccional e informa a la paciente de las indicaciones a seguir, mejoras en el tratamiento, así como programar la nueva cita, la posibilidad de alta o de remisión al médico de la Unidad.

Con el devenir del tiempo, se ha objetivado la pérdida de citas por parte de algunas pacientes, inasistencias y pérdida de pacientes que el personal sanitario achaca a que la valoración de las consultas por parte de ellas, en un momento dado del seguimiento, puede parecerles rutinarias, innecesarias, insuficientes o incluso puede estar influenciado por la dificultad en el traslado desde su domicilio o las limitaciones inherentes a nuestra geografía insular.

Por tanto, con el objetivo de mantener tasas elevadas de persistencia terapéutica, incrementando a su vez la calidad y la eficiencia en la asistencia a estas pacientes, se pretende incluir las nuevas tecnologías en la Unidad de Suelo Pélvico del CHUIMI. Que éstas sirvan para acercar la paciente a su seguimiento, minimizando el número de veces que tenga que trasladarse al hospital. Además, existe un beneficio secundario, redirigir los tiempos de consulta a aquellas otras tareas para las que, dada la elevada incidencia de incontinencia urinaria, no se dispone actualmente de agenda.

## 1.2. Objetivos

A tenor de lo expuesto en el capítulo anterior, el complejo hospitalario se encuentra en la necesidad de crear un entorno web, sencillo, para el control y seguimiento de la calidad de vida del paciente con vejiga hiperactiva, que acompañe al paciente el tiempo necesario para su mejora.

Este entorno implica nuevas metas a completar, donde seguir un orden metodológico es crucial para evitar el mayor número de errores. El primer objetivo será la creación de los formularios o encuestas donde el paciente podrá interactuar. Este debe estar diseñado de manera intuitiva para un fácil manejo, dado que muchas veces estará destinado a usuarios con una comprensión baja en el uso de plataformas web. El segundo objetivo es gestionar un sistema que almacene los pacientes y los formularios relacionados con ellos. Por último, se exige que haya una gran seguridad de los datos almacenados y de la funcionalidad interna de dicho sistema ya que esta aplicación está destinada a un organismo público y debe satisfacer una serie de normativas ya estandarizadas. Todo ello debe estar planteado de una forma interactiva, fácil y amigable en el uso de la interfaz de la aplicación.

La manera de afrontar estos objetivos será clave para completar este Proyecto de Final de Grado. Por ello se dividirá la aplicación en diferentes secciones o módulos para mejorar la planificación del proyecto. El objetivo final es captar la atención del paciente para que se adhiera a este tipo de seguimiento diferente al que están acostumbradas evitando lo máximo posible el desinterés por su parte.

Con la realización de esta aplicación se pretende conseguir una serie de utilidades centrándonos en el seguimiento y control detallado del paciente, de donde se podrá sacar datos interesantes sobre su estado de salud, valorando sus avances y obteniendo estadísticas relevantes, como por ejemplo, la relación entre medicamento tomado y las micciones diarias o liquido tomado y posibles escapes a lo largo del día. Aquí es donde tiene lugar otro de los objetivos más importantes de la aplicación. Se deberá tener en cuenta estas opciones para permitir obtener datos que midan la calidad de vida del paciente, sus avances o su adherencia al tratamiento.

A su vez, en este documento se explica el proceso de desarrollo de una aplicación web desde su concepción, junto con un framework de uso simple, con el objetivo ordenar de manera profesional los módulos de nuestra aplicación utilizando una sintaxis común tanto para su mantenimiento como para desarrollos posteriores.

### 1.3. Estructura del documento

La estructura de este documento está constituida por los siguientes apartados: introducción, estado del arte, normativas seguridad para el desarrollo de la aplicación, análisis y diseño de la aplicación, implementación, pruebas, conclusiones y presupuesto.

Antes de explicar el desarrollo de la aplicación en sí, se analizan las herramientas y tecnologías que existen en la actualidad, para conocer lo que se puede llegar a conseguir y así reforzar la base de la aplicación. Es de gran utilidad comparar las soluciones software ya disponibles para abordar los objetivos planteados, destacando tanto sus ventajas como sus inconvenientes, para luego dar paso a las tecnologías que se van a utilizar. Se realizará un breve resumen de los conocimientos necesarios para el desarrollo con dichas tecnologías.

En el capítulo siguiente se realiza una descripción detallada de la solución a implementar, aportando el catálogo con las especificaciones y requerimientos de la aplicación.

En el cuarto capítulo se exponen los criterios de seguridad que debe seguir una aplicación Web en la infraestructura corporativa del Gobierno de Canarias para mantener la seguridad y la privacidad tanto de los pacientes como del personal del complejo.

A continuación, una vez definidos todos los requisitos necesarios, se procede a diseñar la arquitectura de la aplicación. Debemos tener una idea clara sobre lo que tenemos que hacer y elegir correctamente las herramientas y el lenguaje de programación que emplearemos, ya que una buena elección de nuestras herramientas de trabajo nos puede facilitar mucho las tareas a realizar en las siguientes fases de desarrollo de la aplicación, siendo por tanto una fase de real importancia. En este apartado también se incluye diversos diagramas sobre el diseño del sistema.

Una vez explicada la funcionalidad y diseño web, en el capítulo 6 se explica la implementación de cada uno de los módulos en los que se divide la aplicación a través del framework elegido. La intención es que sirva de guía del proceso al programar sus algoritmos y explicando toda su lógica y las consultas a la base de datos para almacenar o recuperar los datos necesarios.

En el séptimo capítulo se realizan una serie de pruebas sobre la aplicación para comprobar la eficiencia y la calidad del software. Una vez completado, también se analizan los resultados obtenidos. En el siguiente capítulo se comentan las conclusiones obtenidas acerca del proyecto, así como las nuevas ideas para futuras versiones.

Por último, se expone los costos asociados al diseño, desarrollo e implementación de dicho proyecto, evaluando el material empleado y el tiempo invertido en él.



## 2. Estado del arte

---

Para cumplir con los objetivos establecidos, previamente hay que comparar los sistemas similares que existen en la actualidad. En la segunda parte del capítulo, se explica la tecnología que sigue una aplicación web y los conceptos básicos que son necesarios.

### 2.1. Servicios actuales

El uso de las tecnologías para apoyar las ciencias de la salud tiene el potencial de transformar la prestación de servicios sanitarios en todo el mundo. Esto incluye rápidos avances en el ámbito tecnológico y el crecimiento exponencial de las aplicaciones web o móviles, lo que conlleva nuevas oportunidades para la integración de la salud móvil en los servicios sanitarios. Sin embargo, la calidad de estas apps es muy variable, tanto en contenidos como en funcionalidades, por ello siempre es mejor una atención directa con los especialistas sanitarios y seguir sus recomendaciones acerca de que aplicaciones usar.

Hoy en día existen numerosos y variados tipos de aplicaciones destinadas al control y seguimiento sanitario de diferentes enfermedades, siendo estas apps enfocadas a problemas de salud crónicos como pueden ser la diabetes, soriasis o distintos tipos de cáncer entre muchos otros. Generalmente estas aplicaciones te hacen un seguimiento de tu dieta, citas con el médico, información útil sobre su afección o tipo de medicación recomendada por expertos [2].

Las aplicaciones más populares según una encuesta realizada por la fundación iSYS (fundación dedicada al desarrollo de proyectos sociales de salud digital), se pueden diferenciar si va dirigido a un profesional del sector o para un paciente, además han creado un baremo el iSYScore v.1 [3] en conjunto con un amplio espectro de expertos, médicos, enfermeras, especialistas en salud pública, comunicación y tecnología, sin olvidar los principales destinatarios de estas apps, como son los pacientes, lo que da la posibilidad de elaborar este baremo que permite evaluar y puntuar cualquier aplicación de salud.

Las aplicaciones más relevantes para este trabajo y que tienen las mejores críticas tanto de los consumidores como del personal médico, son: One Drop: Gestión de Diabetes y ACO+ [2]. Por ello han sido los elegidos en el análisis competitivo que se adjunta en la Tabla 1. Para su estudio, se han destacado las valoraciones según interés, confianza y utilidad.



Figura 1 Logotipo Fundación iSYS [2].

	<i>One Drop: Gestión de Diabetes</i>	<i>ACO+</i>
<i>Interés</i>	Disponible en las dos plataformas más populares (iOS y Android), con buenas valoraciones.	Esta aplicación se puede obtener tanto en la App store de iOS como en Google Play la tienda de aplicaciones de Android.
<i>Confianza</i>	Obtiene la máxima puntuación por disponer del sello CE de producto sanitario.	Obtiene una buena puntuación por estar valida por un organismo sanitario (Clase 1), por disponer de web, citar fuentes de evidencia, nombrar la entidad responsable, tener una actualización reciente y por informar de las fuentes de financiación.
<i>Utilidad</i>	Puntúa por facilitar el seguimiento de la diabetes, información sobre la condición y facilitar la conexión con otros afectados.	Puntúa en las variables de "herramienta", al facilitar cálculos, apoyo en un procedimiento asistencial y en el archivo-manejo de datos de paciente. Como medio puntúa por facilitar el seguimiento de pacientes.

Tabla 1 Análisis competitivo (One Drop: Gestión de Diabetes y ACO+).

Como se puede comprobar se debe tener una alta relación entre interés comercial confianza con el usuario y utilidad en el aspecto a tratar para que una aplicación consiga abrirse hueco en el gran mercado de apps del sector.

La era de la comunicación móvil se consolida cada vez con más fuerza y es necesario adaptarse a estos cambios, provocados también por las grandes empresas, como en su día hicieron Facebook o Apple (eliminando el uso de la tecnología Flash) y hace algunos años Google con una nueva exigencia para los sitios Web, penalizando las Webs que no sean compatibles con dispositivos móviles, es decir, que no cuenten con un diseño adaptable.

## 2.2. Tecnologías utilizadas

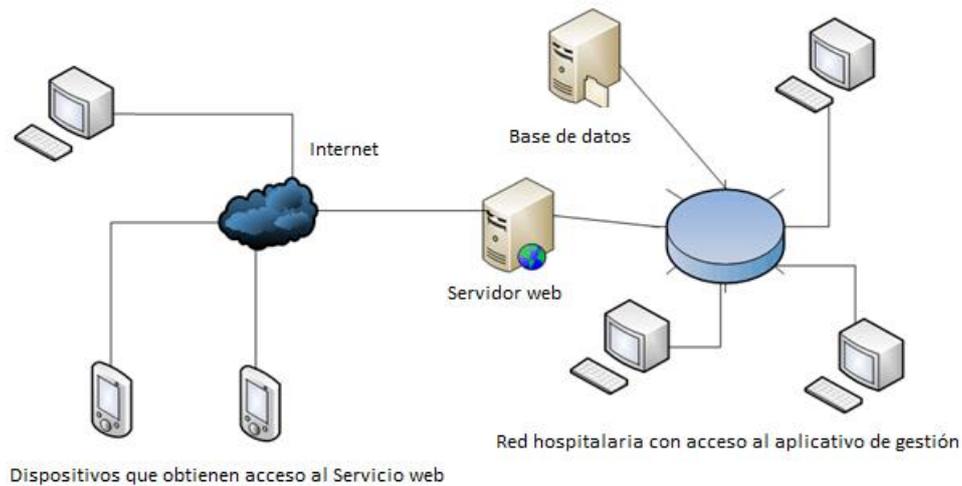
Con el paso de los años, ha surgido la necesidad de que la información sea accesible tanto en sistemas con redes internas o locales como desde el exterior, y que dicha información pueda ser transportada en todo momento. Estas necesidades han provocado un cambio en las aplicaciones tradicionales de escritorio y han transaccionado a las llamadas aplicaciones

Web. Con este cambio, ya no solo se requiere que se muestre la información en la aplicación, también se exigen una serie de cualidades imprescindibles como pueden ser seguridad, portabilidad, escalabilidad, eficiencia, disponibilidad etc. A continuación, se explican las tecnologías empleadas [4].

Una aplicación web es una aplicación o herramienta informática accesible desde cualquier navegador, bien sea a través de internet (lo habitual) o bien a través de una red local. A la hora de diseñar una estructura Web, lo primero a tener en cuenta es el lugar donde se almacenará el contenido de la aplicación. El entorno puede ser, según su funcionamiento, una extranet o intranet. En el primer caso, se podría acceder a la Aplicación Web desde cualquier punto con acceso a Internet, sin embargo, en nuestro caso concreto, la aplicación funcionaría a través de una intranet donde únicamente se podría acceder desde la red interna CHUIMI, pudiendo entrar solo el personal adecuado para su uso.

En el siguiente paso se debería de contratar un dominio Web a una compañía especializada (registrador de dominios) a cambio de tener un nombre de dominio que sea representativo del sitio Web, algo que cambia con respecto a nuestro caso será el Gobierno de Canarias quien lo proporciona. Otro tema a tener en cuenta es el DNS (Sistema de Nombres de Dominio) que se encarga de traducir el nombre de la URL en una dirección IP y enviar la petición web al servidor. Esta abstracción permite que cualquier servicio de red pueda estar disponible desde cualquier dispositivo en la red, aun cuando la dirección IP sea diferente.

Una vez que se ha obtenido un nombre de dominio, es necesario disponer de un alojamiento Web, es decir, de un servicio (servidor Web) donde poder guardar la aplicación Web y que se encargue de procesar las peticiones en el puerto correspondiente del servidor. Los usuarios, mediante un navegador, emiten solicitudes vía protocolo HTTP a un servidor Web, que aloja una serie de aplicaciones o páginas Web que ofrecen diferentes servicios a las personas conectadas. Por ello, es aquí donde se centraliza todo el trabajo, mientras que en el navegador sólo se presenta la información.



**Figura 2 Conexionado entre paciente y personal sanitario.**

Existen varios tipos de alojamiento web [5], entre los cuales cada usuario tiene la opción de elegir la que más se adecue sus necesidades. Entre los principales tipos se encuentran el alojamiento gratuito, que cuenta con recursos muy limitados; el alojamiento compartido, donde se alojan clientes de varios sitios en un mismo servidor y es un servicio económico y con buen rendimiento, ideal para pequeñas y medianas empresas; servidores virtuales, que ofrecen el control de una computadora, que se realiza mediante una máquina virtual. Así se pueden administrar varios dominios de forma fácil y económica, recomendado para empresas de diseño y programación web; servidores dedicados, donde un solo cliente o empresa dispone de todos los recursos de la máquina pero que generalmente tiene un coste mayor al del alojamiento compartido debido a la necesidad de contratar servicios adicionales para la administración y configuración del servidor; alojamiento web en la nube, basado en las tecnologías más innovadoras que permiten a un gran número de máquinas actuar como un sistema además que aportan mayor seguridad al estar conectados a varios servidores en lugar de solamente uno.

El modelo de comunicación entre cliente y servidor se establece mediante una conexión a Internet y se usa el protocolo HTTP para el intercambio de información. El navegador Web interpreta las páginas dinámicas generadas por el servidor Web.

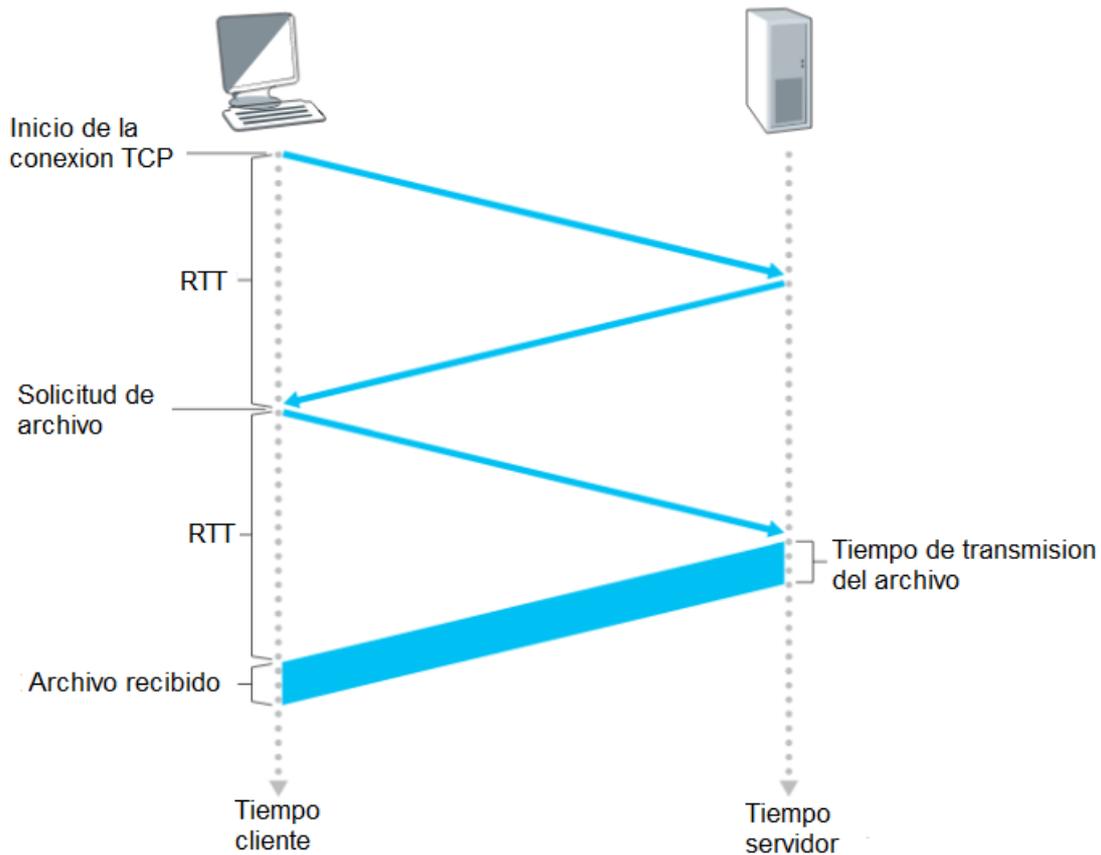


Figura 3 Comunicación HTTP [6].

La aplicación normalmente hace uso de una base de datos para almacenar y traer toda la información necesaria. En resumen, el sistema se distribuye en tres componentes principales: el navegador, la aplicación y la base de datos. Esta estrategia hace que las aplicaciones se encuentren con importantes problemas de escalabilidad, disponibilidad, seguridad, usabilidad, eficiencia o integración, entre otros. Como solución, se ha generalizado la división de las aplicaciones Web en capas [7] (Ver Figura 4):

- **Capa de presentación** o capa cliente orientada a soportar la interactividad de los usuarios con las funcionalidades brindadas por la capa de negocio. En esta capa se encuentran los controles visuales, formularios, etc.
- **Capa de servicio.** Esta capa permite desacoplar la interfaz de usuario del resto de capas, permitiendo que las funcionalidades de nuestra aplicación sean accesibles por otras aplicaciones u servicios.
- **Capa de negocio.** Es la que soporta toda la lógica de negocio. En esta capa encontramos todas aquellas funciones que hacen algún tipo de tratamiento de los datos, se aplica las reglas de negocio, etc.

- **Capa de datos**, que se encarga de mantener toda la información y de almacenar y suministrar los datos al nivel de negocio.

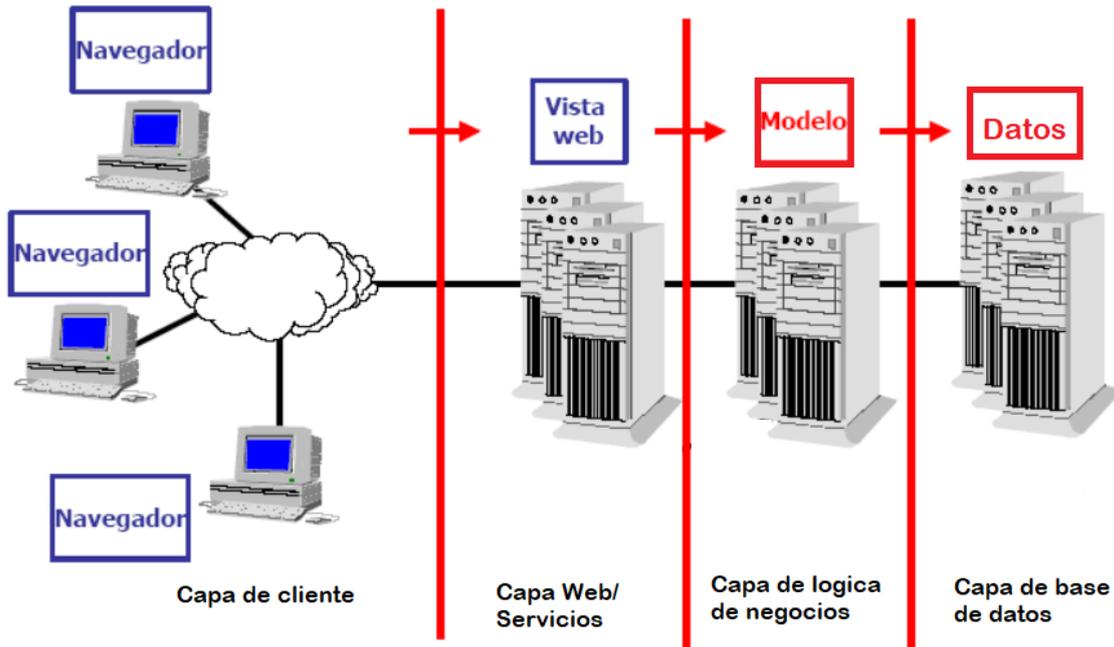


Figura 4 Modelo en 4 capas [7].

Cada capa interactúa solamente con sus capas adyacentes, por ejemplo, la capa de presentación no se da cuenta del tipo de base de datos utilizado. Esta separación en capas permite una gran flexibilidad a la hora de construir aplicaciones. Para una mayor facilidad de desarrollo, surge una serie de pilas de aplicaciones. La Tabla 2 muestra una lista de las más populares en la actualidad.

Servidor Web	Framework	Lenguaje de programación	Gestor de base de datos
Apache	Symfony, Laravel	PHP	PostgreSQL (phpPgAdmin)
NGINX	Django	Python	MySQL (phpMyAdmin)
IIS / IIS Express	ASP.net	VB, C#	SQL Server (Access)
Node.js	Express	JavaScript	MongoDB (mongoVue)

Tabla 2 Stacks más usados [8].

Como se ha podido ver en la tabla anterior, existen numerosos stacks conocidos, como por ejemplo: MERN (MongoDB/Express/React/NodeJS), LAMP (Linux/Apache/MySQL/PHP), etc., que facilitan el desarrollo de aplicaciones Web [8]. Además, utilizar este tipo de arquitecturas tienen muchas ventajas, como pueden ser mayor robustez en la aplicación, proporcionan mayor flexibilidad modificar solo los módulos a tratar de una capa sin afectar a las restantes, ofrece la posibilidad de reutilizar código en otras aplicaciones o versiones futuras y por último ofrece una alta escalabilidad, añadiendo más servidores para manejar más peticiones de usuario mejorando el rendimiento.

Por el contrario, las aplicaciones Web también traen consigo varios aspectos negativos, entre los que hay que tener en cuenta: mayor incremento del tráfico en la red; mayor balance de carga y tolerancia frente a posibles fallos; mayor exigencia de seguridad ya que las aplicaciones están expuestas a ataques maliciosos.

El último paso a tener en cuenta, es el diseño Web. Hasta ahora, el trabajo realizado en la parte del servidor, lo realiza el programador Web o Back-End Developer. Pero la parte en la que se indica cómo se debe mostrar la información al usuario, no deja de ser menos importante. Por ello existen varias tecnologías, lenguajes como HTML5, CSS3, JavaScript y frameworks como por ejemplo JQuery, Bootstrap o AngularJS, de los que se encarga el diseñador Web o Front-End Developer

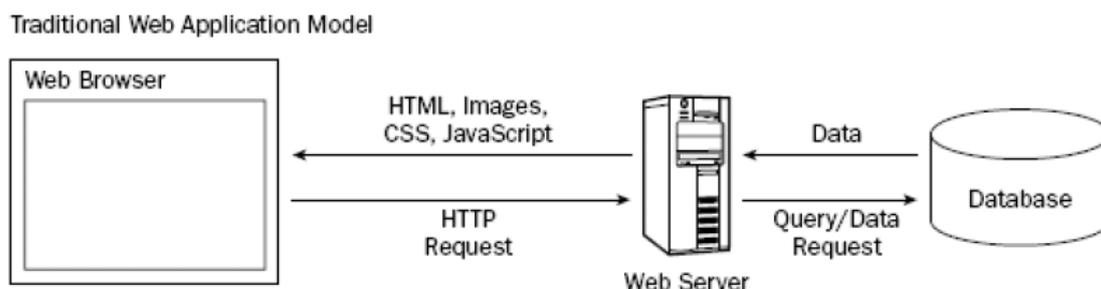


Figura 5 Modelo de aplicación Web tradicional [9].

Lo cierto es que, en los últimos años, el diseño Web está teniendo más importancia junto con el desarrollo del lado del servidor, y cada vez es más frecuente conectarse a los datos desde el navegador sin pasar por el servidor. Aparte de las tecnologías GET y PUT, existen otras como AJAX (Figura 6), que es una técnica de desarrollo web para crear aplicaciones interactivas que resultan bastante útiles para conectarse con la base de datos en un proceso de verificación de formularios, por ejemplo. Esta mantiene una comunicación asíncrona

donde se puede usar JSON o XML como forma de empaquetar la información en el intercambio de los datos, dando la posibilidad de realizar cambios sobre las páginas sin necesidad de recargarlas, mejorando la interactividad y la velocidad en las aplicaciones.

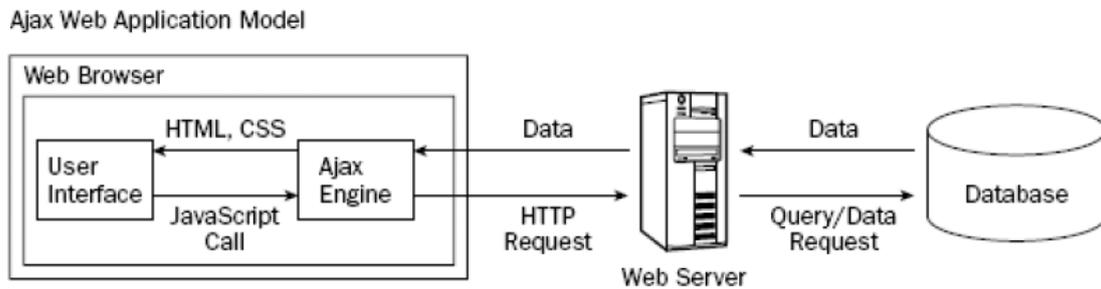


Figura 6 Modelo con tecnología AJAX [9].

Una de las soluciones a la hora de tratar algunos problemas complejos de la arquitectura web es seguir el estilo de arquitectura MVC (Modelo Vista Controlador) que se encarga de separar la lógica de negocio de la interfaz de usuario. Es uno de los más utilizados en las aplicaciones ya que facilita la funcionalidad, mantenibilidad y escalabilidad del sistema de forma sencilla.

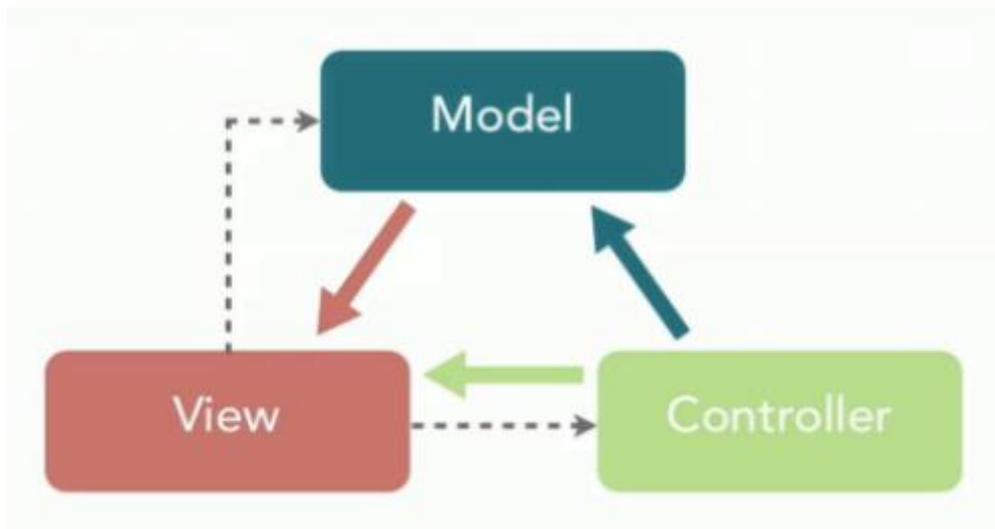


Figura 7 Modelo Vista Controlador [10].

Utilizando esta estructura, la capa intermedia de una aplicación Web se divide en tres componentes encargadas de apartados muy diferenciados: El **Modelo** contiene una representación de los datos que maneja el sistema, su lógica de negocio, y sus mecanismos de persistencia. La **Vista**, o interfaz de usuario, está compuesta por la información que se envía al cliente y los mecanismos interacción con éste. Y por último el **Controlador**, que

actúa como intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno [11].



### 3. Análisis de los requisitos

En este capítulo se describen los requisitos necesarios para llevar a cabo el proyecto. En los siguientes subcapítulos se detallan las características de la aplicación, además se enumeran las especificaciones de dicho producto.

#### 3.1. Proceso de desarrollo

El objetivo final de este Trabajo de Fin de Grado en Ingeniería en Telecomunicación y Electrónica es la construcción y desarrollo de un producto software, en específico una aplicación Web para el seguimiento y tratamiento de pacientes con VHA.

Se seguirán todas las etapas del proceso de desarrollo de software (Figura 8): análisis de requerimientos, diseño del sistema, desarrollo, integración, pruebas y mantenimiento, usando un modelo de ciclo de vida en cascada. Este es un modelo clásico, de los más usados para proyectos sencillos donde se usa un enfoque metodológico que ordena rigurosamente las etapas del proceso para el desarrollo software, de tal forma que el inicio de cada etapa debe esperar a la finalización de la etapa anterior.

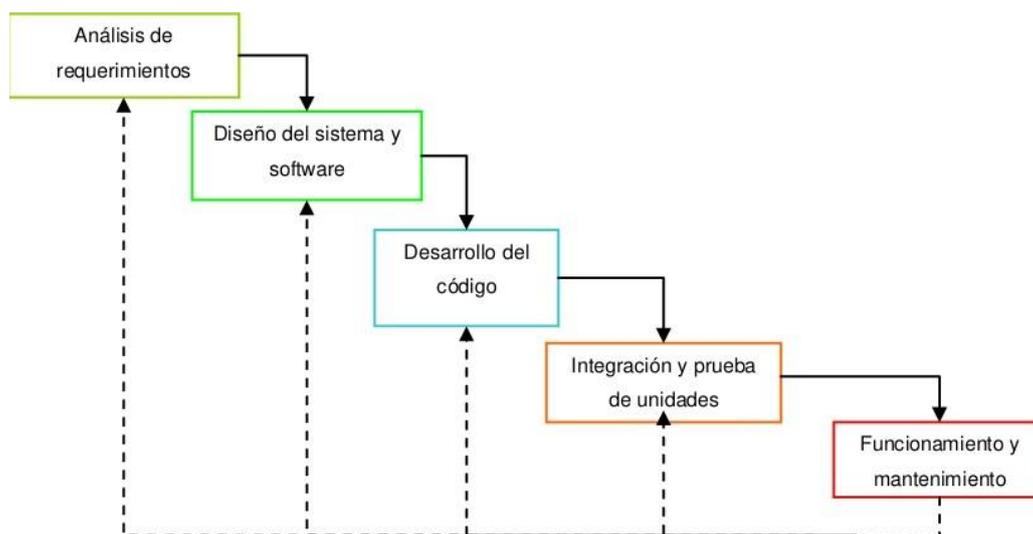


Figura 8 Ciclo de vida: Modelo en cascada.

Al final de cada etapa, el modelo está diseñado para llevar a cabo una revisión final, que se encarga de determinar si el proyecto está listo para avanzar a la siguiente fase. No se realizará la entrega del producto final hasta la validación de la última etapa. cualquier error de diseño detectado en la etapa de prueba conduce necesariamente al rediseño y nueva

programación del código afectado, aumentando los costos del desarrollo, por ello para la realización de proyectos de mayor escala se suelen utilizar modelos diferentes como pueden ser el incremental, en espiral o el de prototipos.

## 3.2. Análisis de Requisitos

En este punto se van a obtener y definir cuáles son las necesidades que se deben satisfacer, antes de comenzar a desarrollar el producto. Para ello, se realizará un estudio completo de las necesidades que deberá satisfacer la solución escogida finalmente, este estudio define una serie de funcionalidades. A éstas se les conoce como requisitos del sistema y se dividen en dos tipos: requisitos funcionales y requisitos no funcionales.

### 3.2.1. Requisitos funcionales

Con los requisitos funcionales se pretende expresar como debe ser el funcionamiento y/o comportamiento del sistema frente a las interacciones con su entorno. Estos requerimientos van a depender de los posibles usuarios de la aplicación.

#### **Aplicación**

RF (1). – Acceso restringido a los usuarios no dados de alta en el servicio informático del CHUIMI.

RF (2). - El sistema dispone de un módulo de gestión de los pacientes.

RF (3). - El sistema permitirá añadir nuevos pacientes.

RF (4). - La aplicación acepta solamente dos tipos de usuarios: administración y personal sanitario.

RF (5). - La aplicación tiene que listar a los pacientes que forman parte del estudio.

RF (6). - La aplicación tiene un servicio de contacto automático para enviar E-mails y SMS a los pacientes como recordatorio.

RF (7). - El envío de formulario debe almacenar fecha y hora en la recepción.

RF (8). - La aplicación podrá realizar cálculos matemáticos automáticos estándares para el estudio del VHA.

RF (9). - Los pacientes tendrán un tiempo determinado para responder la encuesta.

RF (10). - Los pacientes se podrán dividir dependiendo de su estado, en pacientes que han respondido la encuesta, pacientes que no la han respondido o pacientes que están fuera de plazo.

RF (11). - El sistema ha de ser capaz de poder buscar un paciente por nombre, fecha y también se podrán hacer búsquedas según su estado (si ha enviado o no la encuesta a tiempo).

### **Personal sanitario**

RF (12). - El personal sanitario podrá añadir pacientes a la aplicación desde la base de datos central del hospital.

RF (13). - El personal podrá realizar búsquedas por NHC (Número de historia clínica), nombre u otras categorías del paciente.

RF (14). - El personal podrá acceder a una vista detallada de los datos del paciente.

### **Paciente**

RF (15). - Cada paciente debe tener un código numérico anonimizado por para identificarse en el formulario-encuesta.

RF (16). - Los pacientes podrán rellenar la encuesta pudiendo responder las distintas preguntas diseñadas y expresando sus opiniones en sus apartados específicos.

RF (17). - Los pacientes deben responder todas las cuestiones para poder enviar el formulario.

## **3.2.2. Requisitos no funcionales**

Los requisitos no funcionales no describen información a guardar, ni funciones a realizar, sino características de funcionamiento.

### **Interfaz y usabilidad**

RNF (1). - La interfaz gráfica destinada al personal sanitario será sencilla, atractiva y fácil de manejar.

RNF (2). - La encuesta dirigida al paciente debe ser atractiva, breve y sencilla.

### **Mantenimiento y portabilidad**

RNF (3). - El administrador tendrá la opción de poder mantener el sistema con actualizaciones de versiones con el fin de corregir errores y mejorar los servicios.

RNF (4). - La aplicación será privada, permitiendo al administrador el acceso al servidor.

### **Recursos**

RNF (6). - La información de todos los usuarios de la aplicación se almacenará en una base de datos.

RNF (7). - La información de todos los datos de los formularios enviados por los pacientes estarán almacenado en una base de datos.

**Seguridad y privacidad**

RNF (8). - Existirá un registro de usuarios que han accedido a la aplicación y a la hora que han accedido.

RNF (9). - Existirá un registro de las acciones tomadas dentro de la aplicación por parte del personal sanitario.

**Requisitos tecnológicos**

RNF (10). - La aplicación usará diseño responsivo y será compatible con dispositivos móviles y tabletas con conexión a la intranet.

## 4. Normativa de infraestructura y seguridad para aplicaciones del gobierno de canarias

---

Dado que esta aplicación pertenece a una entidad hospitalaria esta debe seguir unos protocolos de seguridad mayores que otras aplicaciones web destinadas a otros ámbitos, en este capítulo se citan algunos aspectos a tener en cuenta dadas por el Gobierno de Canarias según la Dirección General de Telecomunicación y Nuevas Tecnologías.

### 4.1 Normativa de gestión de alojamiento de aplicaciones

En este apartado presenta la normativa que debe seguir una aplicación web entorno a su gestión de alojamiento descrita por la Dirección General de Telecomunicaciones y Nuevas Tecnologías (DGTNT) [12] con el objetivo final de:

- Minimizar los costes corporativos de mantenimiento.
- Aprovechar al máximo los recursos disponibles.
- Asegurar la calidad del software desarrollado.
- Facilitar la integración de la aplicación con otros sistemas de la organización.
- Estimar las necesidades de recursos de la aplicación.
- Generar una base de datos del conocimiento corporativo.
- Proporcionar alta disponibilidad y facilitar la creación de planes de contingencia para minimizar los efectos negativos frente a desastres.
- Facilitar una administración estándar desvinculándola del requerimiento de un administrador específico.

#### 4.1.1 Normativa de arquitectura

La DGTNT ofrece dos tipos de arquitectura tecnológica para el alojamiento web que se han diseñado teniendo en cuenta el equilibrio entre el coste de los servicios que se prestan, y la autonomía de administrarlos por parte de los responsables técnicos de los mismos [13]:

Arquitectura	Descripción	Modalidad
Core	Infraestructura como servicio	Control Centralizado
Gestión Compartida	Plataforma como servicio	Valor añadido bajo demanda
		Control Delegado / Autoprovisión

Tabla 3 Tipo de Arquitectura DGTNT [12].

La arquitectura **Core** está formada por un parque de servidores que dan soporte estructural del sistema a todos los servicios corporativos y departamentales del Gobierno de Canarias, y cumplen con los requisitos establecidos en esta normativa. A diferencia de la arquitectura de **Gestión Compartida** que se refiere al entorno de infraestructura tecnológica que la DGTNT ofrece para el alojamiento de las aplicaciones corporativas y departamentales que quedan fuera de la modalidad Core. En estas el responsable de la aplicación dispondrá de credenciales de administración que le permitirá realizar la instalación, personalizaciones que sean necesarias, así como su posterior soporte y mantenimiento.

La aplicación desarrollada está bajo la Arquitectura de Gestión Compartida que está constituida por servidores sobre los que CiberCentro (Centro Integral de Servicios Telemáticos del Gobierno de Canarias) realizará la instalación de al menos el sistema operativo, software antivirus y copias de seguridad mientras que la instalación, parametrización y despliegues de la aplicación serán trabajo del responsable/gestor de la aplicación y del equipo técnico que se designe.

Todos los servidores están auditados por una herramienta de seguridad que realiza un escaneo automático de vulnerabilidades. En caso de detectar vulnerabilidades o riesgos provocados por un servidor de Gestión Compartida que comprometan la seguridad de la red corporativa, el sistema bloqueará de forma automática todos los accesos al servidor afectado desde dentro y/o fuera de la red corporativa y se habilitará únicamente el acceso a una IP determinada para que el responsable/gestor pueda acceder y solucionar la vulnerabilidad encontrada.

Por motivos de seguridad todos los servidores se encuentran ubicados en redes específicas protegidas por Firewall, tanto las redes internas como las externas al complejo, sin embargo, para una mayor protección de datos la aplicación estará ubicada en una red exclusivamente interna, es decir, que existe una restricción en el servicio prestado por la aplicación alojada en dicho servidor, dado que será accesible únicamente desde la red corporativa del Gobierno de Canarias.

#### 4.1.2 Normativa de mantenimiento

Debe tenerse en cuenta que no basta con cumplir con una correcta instalación de la infraestructura, es obligación de los responsables cumplir con las normativas y medidas de seguridad definidas por la DGTNT. Entre otras:

- Garantizar la actualización de los S.O. y aplicaciones de forma periódica.
- Mantener un control de acceso adecuado a los servidores.
- No hacer uso de software no permitido o ilegítimo, o de aplicaciones que puedan suponer un conflicto con los servicios corporativos que comprometan la seguridad.
- Respetar la nomenclatura asignada en la entrega del servicio. En el caso que sea necesario realizar un cambio de nombre, el responsable debe informar a CyberCentro para actualizar el registro en Gestión de Configuración.
- Respetar la Normativa de Herramientas y Aplicaciones Corporativas.
- Respetar la configuración de los servidores en cuanto a las políticas de Seguridad (actualizaciones de S.O., antivirus, etc.) cualquier modificación realizada será asumida por el responsable / gestor del servidor.

Además, si una aplicación alojada en un entorno CyberGestionado quedara en obsolescencia tecnológica (tanto por versionado del software que utiliza como por tipo de software base), se dará un plazo de tres meses para proceder a su actualización. Transcurrido este plazo, se bloquearán los accesos a los servicios correspondientes.

## 4.2 Normativa sobre los requisitos de Seguridad en Aplicaciones Web

La identificación temprana de las necesidades de seguridad a lo largo de todas las etapas del ciclo de vida ahorra tiempo y dinero, pero sobre todo, facilita la incorporación de la seguridad en las aplicaciones. En este apartado se definen los requerimientos y recomendaciones que el responsable del software debe aplicar cuando se desarrolla o adquiere una aplicación web que va a ser publicada bajo la infraestructura del Gobierno de Canarias [14].

- Todas las aplicaciones deben utilizar el protocolo HTTPS.
- No deben utilizarse puertos no estándar para servir las aplicaciones.
- Toda aplicación que valide contra alguno de los repositorios corporativos deberá hacer uso obligatoriamente de un captcha.
- Usar el método POST para el intercambio de información sensible entre el cliente Web y la aplicación, y el método GET sólo para consultar información, de manera que nunca se envíen mediante los parámetros de una petición GET datos sensibles.
- No usar nunca el método GET para una petición que pueda ocasionar un cambio en el servidor, con el fin de evitar que se ejecute al hacer clic en un enlace.
- No utilizar las cabeceras HTTP como método de validación o envío de información sensible
- Realizar una validación de todos y cada uno de los parámetros de entrada.
- Todos los fallos en la validación de entradas deben dar como resultado el rechazo de la petición, o la modificación de los valores inválidos por unos válidos.

- Realizar una comprobación de los permisos del usuario que realiza la petición. Debe comprobarse que el usuario tenga permisos para realizarla, tanto en el cliente como en el servidor.
- Explícitamente inicializar todas las variables y mecanismos de almacenamiento de información, durante su declaración o antes de usarlos por primera vez.
- Siempre que se utilicen algoritmos de cifrado, deben estar acreditados por el CCN

Junto con los requerimientos anteriormente señalados existen otros muchos, como son la autenticación, para una correcta identificación de los usuarios de las aplicaciones, la autorización, la gestión de sesiones ya que es necesario en la mayoría de las aplicaciones disponer de un mecanismo para mantener el estado de la sesión de un usuario, la validación de los datos, la configuración de la infraestructura, el manejo de errores y la protección de la información.

### 4.3 Requisitos y recomendaciones en el desarrollo y despliegue

De conformidad con el Esquema Nacional de Seguridad (ENS), nuestro proyecto ha de pasar por una serie de exigencias a la hora de su despliegue, deben existir entornos diferenciados de Desarrollo y Explotación, y cualquier cambio debe ser probado en un entorno de Pre-explotación antes de pasar a Explotación. Se deben evitar los riesgos de acceso o cambios no autorizados en el entorno de Explotación por ello deben de cumplir los siguientes requisitos [12]:

**Entorno de Desarrollo:** Este entorno estará disponible en la modalidad de Gestión Delegada únicamente bajo demanda en los casos que se requiera, siendo necesaria la autorización de la DGTNT.

**Entorno de Pre-explotación:** Su configuración ha de ser idéntica a la configuración de Explotación. En este entorno no deben emplearse datos reales, y en él se llevará a cabo el proceso de certificación de seguridad de la aplicación, sin el cual no es viable la publicación en Explotación en aquellos casos en que sea exigible conforme al Proceso para certificar la seguridad de las aplicaciones Web.

**Entorno de Explotación:** Se trasladarán las aplicaciones que hayan superado la certificación de seguridad en el entorno de Pre-explotación, y tras haber seguido el procedimiento de Gestión de Cambios. Una vez en Explotación, si se quieren modificar los contenidos desplegados, deberán implementarse en Desarrollo y probarse en Pre-explotación y verificar su operatividad antes de pasar a Explotación.

## 5. Diseño de la aplicación

---

Una vez se ha realizado el análisis de requisitos y se dispone de una idea general de lo que el sistema pretende y necesita, es hora de comenzar la fase de diseño de la aplicación. En ella, se pretende tener un diseño que especifique como debe ser el sistema, para luego proceder a realizar la programación de las distintas entidades. Primero se especificará la arquitectura elegida que resuelva mejor el problema, junto con sus herramientas utilizadas y posteriormente se presentará los modelos de la aplicación y el diseño de la base de datos.

### 5.1. Arquitectura web

Para explicar la arquitectura web de la aplicación, es necesario elegir previamente las herramientas o tecnologías que se van a usar para después abordar con mayor facilidad los requisitos mencionados anteriormente. Al haber múltiples alternativas, es una fase importante ya que una buena o mala elección repercute en las siguientes fases del proyecto.

#### 5.1.1. Herramientas de Back-End

Las herramientas Back-End son aquellas que se encuentran del lado del servidor y se encargan de interactuar con bases de datos, verificar maniobras de sesiones de usuarios, montar la página en un servidor y servir todas las vistas creadas por el desarrollador Front-End.

Para la realización de este proyecto se va a utilizar un servidor Apache II con PHP 5.4 sobre una plataforma Rancher Kubernetes [15] con imagen base Debian [16]. Kubernetes es una plataforma portable y extensible de código abierto para administrar cargas de trabajo y servicios, esta, facilita la automatización y la configuración declarativa ahorrando con ello el tiempo en la codificación. Al estar implementada con Rancher, esta facilita la implementación de servicios y la visibilidad de todo lo que se ejecuta en un clúster. Se obtiene así una mayor facilidad en la definición de reglas de programación, comprobaciones de estado, controladores de ingreso, seguridad en el almacenamiento y otras opciones de configuración clave.

Resulta indispensable la utilización de un servidor Web que atienda las peticiones de los clientes y responda con los contenidos correspondientes. Pudiendo escoger entre decenas

de servidores de alta calidad como pueden ser Nginx o IIS, sin embargo, se ha decidido usar un servidor Apache [17] ya que es un servidor Web de código abierto que implementa el protocolo HTTP/1.1 y es de uso conocido por el personal que administra la red del CHUIMI. Apache es una multiplataforma y su arquitectura es muy modular y extensible, lo que permite de una manera muy sencilla ampliar sus capacidades, además esta nos permite usar HTTPS que es parte de la obligación que establece la Normativa sobre los requisitos de Seguridad en Aplicaciones Web del Gobierno de Canarias.



Figura 9 Servidor Web Apache [17].

Un caso concreto del servidor web Apache es el módulo ModRewrite [18], que se encarga de traducir, redirigir y modificar direcciones URL para hacerlas más amigables. Hoy en día, este módulo es una herramienta indispensable sobre todo de cara al posicionamiento en Google, ya que una URL es sensible a mayúsculas y minúsculas y no es un caso favorable al duplicar contenido. Aunque generalmente se utiliza para conseguir que las direcciones visibles sean sencillas de recordar y evitar las incómodas variables.

A la hora de elegir un lenguaje interpretado, existe una gran variedad: ASP.net, Java, Ruby, Python, PHP o incluso JavaScript también del lado del servidor (que puede resultar atractivo, al usar un mismo lenguaje en toda la aplicación). Se ha decidido usar PHP [19] (Hypertext Preprocessor), al ser el lenguaje más conocido y con mayor comunidad de usuarios detrás, que cuenta con una extensa biblioteca de funciones. PHP sirve para crear aplicaciones web dinámicas con acceso a información almacenada en una base de datos y que permite incorporar código en las páginas HTML de forma sencilla. Usa técnicas de programación orientada a objetos y sigue el patrón de diseño Modelo Vista Controlador. Además, es un lenguaje soportado perfectamente por el servidor web Apache.

Para que se pueda entender mejor el funcionamiento de PHP en un servidor se puede ver la figura 10 donde se hace un recorrido ejemplo de un usuario haciendo una petición de acceso a una Web en la que a continuación PHP interpreta y le da una respuesta en HTML.

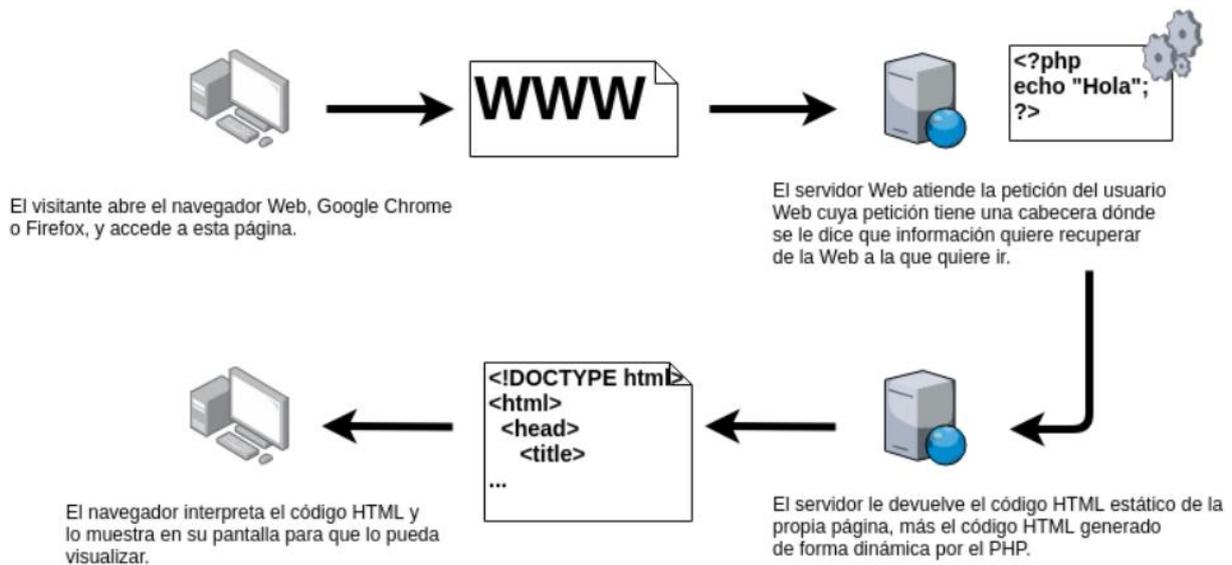


Figura 10 Funcionamiento PHP [20]

El servidor Web y en especial, este módulo PHP, puede conectarse a una base de datos para almacenar o extraer información a la hora de elaborar las páginas. Aquí también existen varias alternativas: MySQL, SQL Server, MongoDB o PostgreSQL, una base de datos NoSQL en la que se puede ejecutar JavaScript para realizar consultas. Sin embargo, en nuestro caso concreto usaremos MySQL [20] por ser la base de datos usada por el servicio informático del CHUIMI en la mayoría de las aplicaciones creadas por su parte y por tanto se tiene una mayor familiaridad. Siendo MySQL muy rápida en la lectura cuando utiliza el motor no transaccional MyISAM, muestra de ello su popularidad siendo la base de datos más usada en el mundo, pero esta virtud también puede ser un defecto ya que puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. Por suerte en aplicaciones Web en las que hay baja concurrencia en modificación de datos y en cambio el entorno es intensivo en lectura de datos, como se estima que será la aplicación en cuestión, hace a MySQL ideal para nuestro objetivo.



Figura 11 Base de datos MySQL [20].

### 5.1.2. Herramientas de Front-End

Las herramientas Front-End sirven para agilizar y optimizar el trabajo realizado con los lenguajes de programación y de diseño del lado del cliente, principalmente Javascript y CSS, pero también con imágenes, HTML, etc. Estas herramientas son fundamentales en el desarrollo web, ya que nos ahorran mucho trabajo en el día a día a la hora de realizar desarrollos de sitios y aplicaciones web, eliminando cantidad de trabajo pesado y repetitivo [21]. Su utilidad básica es la de automatizar tareas, como la creación de archivos optimizados para sitios web, el empaquetado de código Javascript y CSS y la compilación del Javascript.



Figura 12 Herramientas Frond-End [22].

Los contenidos devueltos por el servidor son interpretados por los navegadores Web de los clientes, que los muestran por pantalla acorde a unas fuentes y formatos. Es decir, el servidor Web se limita a transferir el código de la página sin llevar a cabo ninguna interpretación de ella. Aquí es donde intervienen las herramientas del lado del cliente. Una de ellas se trata del lenguaje HTML, un lenguaje de marcado que se utiliza para el desarrollo de páginas o documentos web, que tiene como finalidad indicar como va ordenado el

contenido de una página web. Esto lo hace por medio de las marcas de hipertexto llamadas tags, es decir, para añadir un elemento externo a la página (imagen, vídeo, script, entre otros.), este no se incrusta directamente en el código de la página, sino que se hace una referencia a la ubicación de dicho elemento mediante texto [23].

Aunque la funcionalidad y estructura de una página web es sencilla, puede llevar un arduo trabajo construirla si se incluye en una misma página los contenidos, el diseño y la programación. Lo mejor es separar la estructura de un documento de su presentación y diseño, para ello se emplea CSS que es un sencillo lenguaje de programación, fácil de entender que consiste en un grupo de reglas que los navegadores Web interpretan y de acuerdo a lo especificado, representan el contenido de la página. Mediante CSS se pueden configurar muchos parámetros, entre ellos el color, el tamaño, la posición, el formato del texto, en general todo lo referente a la estructura y diseño artístico de la página Web.

### 5.1.3. Framework usado

Una gran ventaja que se dispone en la actualidad a la hora de programar es que ya no se tiene que empezar de cero puesto que podemos hacer uso de frameworks, estas herramientas evitan al usuario tener que programar el acceso a base de datos, la validación de formularios o la seguridad entre otras, de esta manera resulta más fácil centrarse en programar la aplicación.

Para la realización de nuestro proyecto haremos uso de Laravel, un framework potente para aplicaciones web con sintaxis expresiva y elegante creado por Taylor Otwell en el año 2011, que facilita a los desarrolladores la creación de aplicaciones web de forma rápida con un código limpio usando patrones profesionales para los usuarios más avanzados. Lo que nos permite crear tanto aplicaciones simples como otras más complejas [24].

Laravel es un framework para el lenguaje de programación PHP. Aunque PHP es conocido por tener una sintaxis poco deseable, es fácil de usar, fácil de desplegar y se le puede encontrar en muchos de los sitios web modernos que se usan día a día. Laravel no solo ofrece atajos útiles, herramientas y componentes, sino que también intenta arreglar alguna de las flaquezas de PHP.



Figura 13 Logo Laravel [24].

Cuando ingresamos a una *url* directamente desde el navegador lo hacemos mediante una petición HTTP de tipo GET, esta solicitud se envía al archivo donde se encuentran las rutas (routers), en caso de no existir nos dará un error, si la ruta existe, nos llevará a un controlador en el cual se encuentra la lógica, el controlador interactuará con un modelo para recuperar información de una base de datos. Esta información llega al controlador y desde el controlador invocamos una vista que finalmente será donde se carga y se muestra en el navegador. En la figura 14 se muestra de forma esquemática el ejemplo descrito anteriormente.

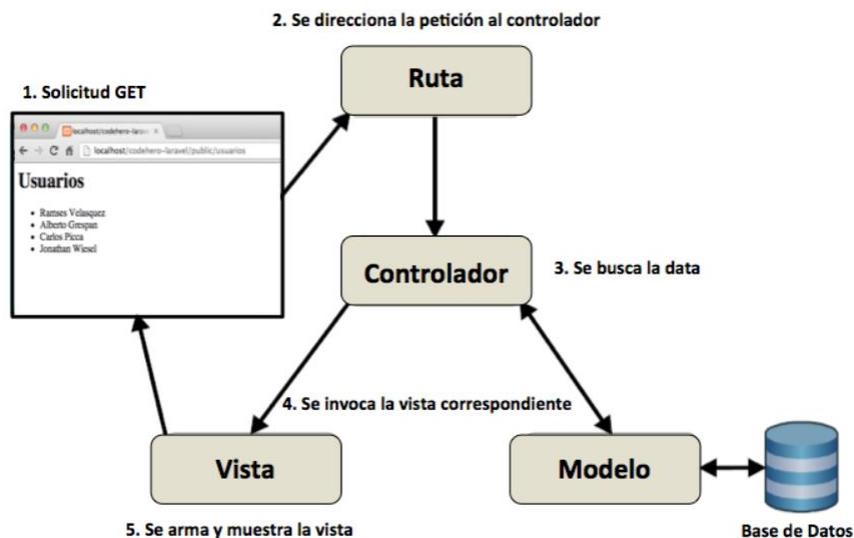


Figura 14 Recorrido de una solicitud GET [25].

Laravel además puede integrar otros FrameWorks simplificándonos el trabajo, en nuestro caso dispondremos de dos más como son Bootstrap [26] y JQuery [27]. Bootstrap nos ayudará con la parte más visual mientras que JQuery nos simplificará la manera de interactuar con los documentos HTML, manejar eventos y desarrollar animaciones.

La mejor forma de aprovechar las características que ofrecen las hojas de estilo sin tener que diseñar todos los archivos CSS es usar Bootstrap, un framework para CSS que permite crear interfaces web con CSS y Javascript adaptable en función del tamaño de la pantalla del dispositivo desde el que estamos navegando. Además, este Framework nos abstrae de tener que preocuparnos por las Media Queries empleadas para adaptar las hojas de estilo y los porcentajes en tus CSS para hacer una web compatible con todos los dispositivos, facilitando la programación.



Figura 15 Logo Bootstrap [26].

Bootstrap ha sido pensado para utilizarse en las versiones recientes de los navegadores de escritorio y navegadores móviles. Aunque también funciona con los navegadores más antiguos, en estos casos es posible que algunos componentes se vean peor de lo esperado. En la tabla 4 se muestran las versiones más recientes de los navegadores y plataformas que puede soportar Bootstrap:

	<b>Chrome</b>	<b>Firefox</b>	<b>Internet Explorer</b>	<b>Opera</b>	<b>Safari</b>
<b>Android</b>	Soportado	No soportado	-	No soportado	-
<b>iOS</b>	Soportado	-	-	No soportado	Soportado
<b>Mac OS X</b>	Soportado	Soportado	-	Soportado	Soportado
<b>Windows</b>	Soportado	Soportado	Soportado	Soportado	No soportado

Tabla 4 Navegadores que soporta Bootstrap [28].

La aparición de lenguajes de programación interpretados como JavaScript, provocó que las páginas HTML también incluyeran el código de las aplicaciones que se utilizan para crear páginas web dinámicas, por lo que es más práctico hacer una separación de estas [23]. Para

el proyecto se usará el framework jQuery de JavaScript para ayudar a incorporar efectos dinámicos, animaciones, modificaciones del comportamiento en determinados eventos o ventanas con mensajes de aviso al usuario.



Figura 16 Logo jQuery [27].

jQuery, en su núcleo, es una biblioteca de manipulación de Modelo de Objetos del Documento (DOM). El DOM es una representación en estructura de árbol de todos los elementos de una página web. JQuery simplifica la sintaxis para buscar, seleccionar y manipular estos elementos DOM. Por ejemplo, JQuery se puede usar para encontrar un elemento en el documento con una determinada propiedad cambiar uno o más de sus atributos o hacer que responda a un evento. En la tabla 4 se muestran los navegadores que puede soportar el framework jQuery en la actualidad:

Navegador	Versión
<b>Chrome</b>	(Actual - 1) y Actual
<b>Edge</b>	(Actual - 1) y Actual
<b>Firefox</b>	(Actual - 1) y Actual, ESR
<b>Internet Explorer</b>	9+
<b>Safari</b>	(Actual - 1) y Actual
<b>Opera</b>	Actual

Tabla 5 Navegadores que soportan jQuery [27].

## 5.2. Estructura del proyecto

Para la realizar este trabajo es necesario estructurar de forma ordenada nuestro proyecto y controlar en todo momento los cambios realizados en él, por ello se ha utilizado Gitlab [29] un servicio web de control de versiones y desarrollo de software colaborativo de código abierto basado en Git. Este nos ofrece alojamiento de wikis y un sistema de seguimiento de errores.

Gracias a Gitlab se ha podido colaborar de forma coordinada con el equipo de informática del CHUIMI pudiendo modificar las clases sin deshacer lo ya creado teniendo un registro de cada cambio. Pudiendo desarrollar la aplicación en un proyecto estructurado como se muestra en la tabla 6, donde se pueden observar las carpetas que lo componen y una breve descripción de su contenido.

<b>Carpeta</b>	<b>Descripción</b>
app	En esta carpeta se encuentra todo el código personal del proyecto. Eso incluye clases que puedan ofrecer funcionalidad, archivos de configuración, controladores, Blades, routes, etc.
commands	Se encuentra la clase SueloPelvicoCommand.php donde se ubican las tareas automáticas de nuestra aplicación.
config	En esta carpeta están los archivos para configuración tanto del framework como para la aplicación en sí.
controllers	En esta carpeta se encuentra la clase SueloPelvicoController.php que nos permitirá relacionar las peticiones HTTP con manipulación lógica, que estarán ligadas a las rutas.
database	Aquí se encontrarán los archivos relacionados con el manejo de la base de datos.
lang	Aquí se encuentran todos los archivos de internacionalización, es decir, los archivos para poder pasar nuestro proyecto de un idioma a otro.
models	En models se encuentran los modelos para nuestras tablas SueloPelvicoDiario.php, SueloPelvicoMensual.php, SueloPelvicoPaciente.php y SueloPelvicoNotificacion.php que nos permitirán recibir o enviar la información a la base de datos.
routes	En esta carpeta se encuentran las rutas que usará nuestra aplicación
views	Esta carpeta ubica nuestras plantillas visuales index.blade.php, addnew.blade.php, show.blade.php, mail.blade.php y printreminder.blade.php.
Bootstrap	En esta carpeta se encuentran los archivos necesarios para inicializar el framework.
public	Dentro de este directorio colocaremos todos los recursos estáticos de nuestra aplicación, es decir, archivos css, js, imágenes y fuentes.
test	Aquí escribiremos los archivos de pruebas que serán ejecutadas posteriormente por phpunit.
External libraries	Aquí están ubicadas las librerías externas.

Tabla 6 Estructura de la aplicación.

Antes del desarrollo de nuestro proyecto es aconsejable realizar una prueba visual del diseño que podría tener la aplicación para su posterior validación por parte del personal de la Unidad de Suelo Pélvico, que serán al fin y al cabo los que harán uso de la aplicación, para ello se ha creado un modelo de la interfaz donde podrá interactuar el sanitario que maneje dicha aplicación. En la figura 17 se puede observar el modelo de la vista principal que tiene la aplicación donde se podrán ver todos los pacientes de forma anonimizada, donde se podrá filtrar a los pacientes como se ha mencionado en anteriores capítulos, borrar y añadir nuevos participantes. En esta vista de la figura 17 si hacemos doble click en el paciente que se quiere consultar se abrirá la vista detallada del paciente que se muestra en la figura 18.

Seguimiento Consulta Suelo Pélvico

**Filtro**

Estado Respondida ▼    Buscar

No respondida  
Fuera de plazo

Filtrar    Borrar

**Acciones**

+ Añadir    Borrar

Encuesta	Fecha ult encuesta	Nombre (job title) ▲	Apellido1	Apellido 2	Fecha Prox Cita	Tfno
1213 Resp	15/01/2020	Juana	Gomez	Gutierrez	12/10/2020 13:50	655555555
1215 No resp		María	Perez	Perez	12/10/2020 14:50	655555555
1216 Fuera plazo		María	Perez	Perez	12/10/2020 14:50	655555555
1217 Resp	21/01/2020	María	Perez	Perez	12/10/2020 14:50	655555555
1218 Resp	1/01/2020	María	Perez	Perez	12/10/2020 14:50	655555555
1219 No resp		María	Perez	Perez	12/10/2020 14:50	655555555

Figura 17 Vista principal.

En la figura 18 se muestra el modelo de la vista en detalle de los pacientes, donde se puede ver aparte de los datos del paciente las respuestas que este ha realizado en las encuestas y cuando se han realizado:

Detalle Seguimiento

Imprimir recordatorio Seguimiento 1561

Atención paciente fuera de plazo debía haber respondido el 14/01/2020

**Paciente**

NHC  Fecha nacimiento: 24/05/1950 (70 años)

Teléfono  Teléfono 2  Correo electrónico

Ultima cita 16/11/2019 Próxima cita 20/02/2020

**Respuestas recibidas**

	Fecha realizacion	Tipo
>	01/01/2020	Diario
>	07/01/2020	Diario

**Notificaciones enviadas**

Fecha	Tipo
13/01/2020	Correo electronico a xxxx@hhhhh
13/01/2020	SMS a 2222222

Figura 18 Vista detalle del paciente.

### 5.3. Diseño de la encuesta

Para el diseño de la encuesta, que es donde el paciente interactuará, se ha decidido usar un generador de encuestas, una herramienta muy útil para el objetivo final que tiene este proyecto. Se ha decidido usar este tipo de software debido a la familiaridad en el uso por parte del complejo hospitalario, siendo esta usada para estudios en otros campos y por el ahorro en tiempo que esta nos brinda.

Existen varios generadores y editores de encuestas, donde, quizás el más popular actualmente es SurveyMonkey [30] un generador de encuestas online gratuito que permite recopilar opiniones y transformarlas en datos impulsados por personas. Sin embargo, el

software utilizado para crear las encuestas en nuestra aplicación ha sido LimeSurvey [31], una aplicación de software libre para la realización de encuestas en línea, escrita en PHP y que utiliza bases de datos MySQL, PostgreSQL o MSSQL. Esta utilidad nos brinda la posibilidad de desarrollar, publicar y recolectar las respuestas de nuestras encuestas.



Figura 19 LimeSurvey [31].

Para el diseño de nuestra encuesta se ha tomado como ejemplo el formulario que debía rellenar el paciente en consulta proporcionado por el equipo de enfermería, intentando así adaptar el ejemplo proporcionado de una forma sencilla y divertida donde se ha buscado principalmente que el paciente no pierda el interés al responder las preguntas, dando como resultado dos encuestas, una de frecuencia diaria y otra que debe responderse de forma mensual. Estas constan de varios tipos de preguntas interactivas de ámbito clínico relacionadas con la afección de la VHA.

Como se ha mencionado se han creado dos tipos de encuestas, la encuesta diaria consta de 10 preguntas que el paciente debe rellenar, obtiene información sobre el comportamiento diario del paciente, al entrar en la encuesta el paciente debe rellenar un campo donde se deberá de identificar con una secuencia numérica aleatoria que el personal de enfermería le habrá proporcionado con anterioridad. Algunas de las preguntas de esta encuesta son cuando se acuesta y se levanta el paciente, si consume productos que podrían ser irritantes para la vejiga, cuanto liquido toma el paciente y cuanto expulsa en forma de orina.

La encuesta mensual es más extensa que la diaria ya que con la diaria no se quiere desmotivar al paciente con multitud de preguntas y que abandone el estudio, sin embargo, es necesario obtener una mayor información para indagar en las posibles soluciones para dicha afección. Por ello se crea la encuesta mensual que consta de 25 preguntas a responder y que entra en mayor detalle en los temas a tratar en esta enfermedad, como si tiene dolor cuando orina o si toma algún tipo de medicamento entre otras.

A continuación, se mostrarán unos extractos de las encuestas con la que el paciente interactuará, en estas muestras podremos apreciar el tipo de preguntas a rellenar y el diseño de la encuesta en sí.

Continuar después

Seguimiento de la paciente de las últimas cuatro semanas.

\*Introduzca el código que se le a facilitado

\*1. ¿Con qué frecuencia pierde orina?

Seleccione una de las siguientes opciones

Por favor escoja...

Marque la respuesta que mas se adecua a su situacion.

\*2. ¿Cuanta cantidad de orina pierde cuando tiene esas pérdidas?

Seleccione una de las siguientes opciones

Por favor escoja...

Figura 20 Encuesta mensual primeras preguntas.

En la figura 20 podemos ver el comienzo de la encuesta Mensual que se realizara cada 4 semanas. Como primer requisito indispensable se nos pide que introduzcamos el código facilitado por el personal sanitario a través de la app. Para mayor facilidad a la hora de rellenar la encuesta se ha adaptado el formato de cada respuesta al tipo de pregunta planteada, esto se puede apreciar tanto en la figura 21 que se ha escogido como tipo de resolución una ventana desplegable como en la figura 22 que se optó por una respuesta ramificada.

\*1. ¿Con qué frecuencia pierde orina?

Seleccione una de las siguientes opciones

Por favor escoja...

Por favor escoja...

Nunca

Una vez a la semana o menos

Dos o tres veces a la semana

Una vez al día

Varias veces al día

A todas horas

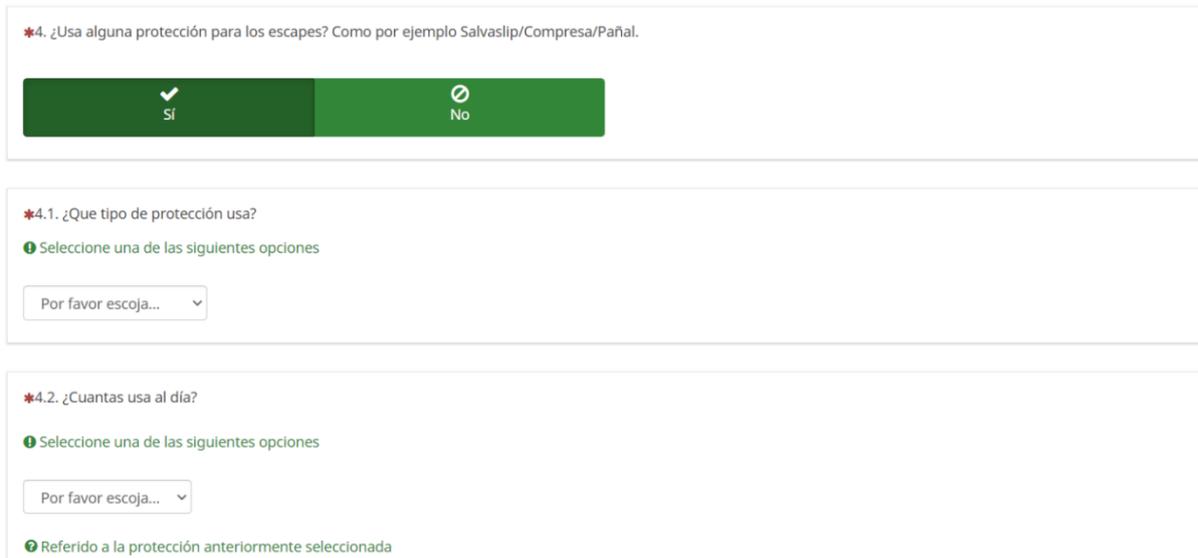
Seleccione una de las siguientes opciones

decu a su situacion.

cuando tiene esas pérdidas?

Figura 21 Opción desplegable.

A continuación, podemos ver otras preguntas, una de ellas con formato “Sí o No” cuya peculiaridad es que si seleccionamos Sí se mostrarán otras subpreguntas relacionadas con esta última.



\*4. ¿Usa alguna protección para los escapes? Como por ejemplo Salvaslip/Compresa/Pañal.

✓ Sí      ✗ No

\*4.1. ¿Que tipo de protección usa?

📌 Seleccione una de las siguientes opciones

Por favor escoja... ▾

\*4.2. ¿Cuantas usa al día?

📌 Seleccione una de las siguientes opciones

Por favor escoja... ▾

📌 Referido a la protección anteriormente seleccionada

Figura 22 Ejemplo de subpreguntas.

Con este tipo de diseños intentamos evitar una sobrecarga visual a la hora de rellenar la encuesta y así no desmotivar al paciente con una gran cantidad de preguntas a responder, ya que buscamos que se cumplimente la totalidad de la encuesta sin que sea un tedio para el encuestado.

#### 5.4. Diseño de la base de datos

Para un eficiente uso y administración de los datos hay que crear una serie de tablas que ordenen estos datos de una forma intuitiva y de fácil acceso. las tablas que conforman nuestra base de datos estarán relacionadas por la ID del paciente, con ella podremos vincular los datos obtenidos de cada formulario con el paciente correspondiente. La base de datos de nuestra aplicación constará de 4 tablas que se mostraran a continuación

## Tabla SP\_PACIENTE

Almacena la información del Paciente en seguimiento por la consulta de suelo pélvico.

Nombre	Tipo dato	Longitud	Descripción	Valores
Id	INT	11	Código numérico Autoincrementable que almacena la identificación del paciente.	
Nhc	VARCHAR	20	Numero de historial clínico.	
nombre	VARCHAR	50	Nombre del paciente.	
apellido1	VARCHAR	100	Primer apellido del paciente.	
apellido2	VARCHAR	100	Segundo apellido del paciente.	
telefono1	VARCHAR	20	Teléfono principal del paciente.	
telefono2	VARCHAR	20	Segundo teléfono del paciente.	
Email	VARCHAR	50	Correo electrónico asociado al paciente.	
fechanacimiento	DATE		Fecha del nacimiento del paciente.	En formato YYYY-MM-DD
fechaultcita	DATE		Fecha de la última cita presencial con el paciente.	
fechaproxcita	DATE		Fecha establecida para la próxima cita con el paciente.	
seguimiento	TINYINT	1	Apartado que señala si el paciente sigue con el tratamiento o seguimiento en consulta.	1 en caso afirmativo 0 en caso negativo

Tabla 7 SP\_Paciente.

## Tabla SP\_DIARIO

Almacena los resultados de las encuestas "Diario miccional" de los pacientes.

Nombre	Tipo dato	Longitud	Descripción	Valores
Id	INT	11	Código numérico Auto incrementable que almacena la identificación del cuestionario diario.	
idspaciente	INT	11	Relaciona la encuesta con un paciente de la tabla SP_PACIENTE.	
horalevantado	TIME		Almacena la respuesta de la pregunta ¿a qué hora se ha levantado?	En formato HH:MM
horaacostado	TIME		Almacena la respuesta de la pregunta ¿a qué hora se ha acostado?	
irritantecacao	INT	1	Almacena la cantidad del producto irritante cacao que ha ingerido el paciente.	Los valores de cantidades entrada nada, poco, moderado y
irritantealcohol			Almacena la cantidad del producto irritante alcohol que ha ingerido el paciente.	
irritantetabaco			Almacena la cantidad del producto irritante tabaco que ha ingerido el paciente.	

irritantepicante			Almacena la cantidad del producto irritante picante que ha ingerido el paciente.	mucho van del 0 al 3 respectivamente
irritantegaseosa			Almacena la cantidad del producto irritante bebida gaseosa que ha ingerido el paciente.	
irritantecafeina			Almacena la cantidad del producto irritante cafeína que ha ingerido el paciente.	
proteccionescapes	VARCHAR	50	Almacena el tipo de protección contra pérdidas de orina usados en el día de hoy.	Los valores obtenidos pueden ser salva (salvaslip), compr(compresas), panal (pañal), compn (compresa de noche) o Ropin (ropa interior)
Kegel	INT	3	Almacena el tiempo (en minutos) de empleado en ejercicios de Kegel al día.	De 0 a 999
liquidotomado6	VARCHAR	50	Almacena el dato referido a la cantidad de líquido tomado entre las 6:00 y las 8:00.	Los valores obtenidos pueden ser 0ml, 100ml, 200ml, 300ml, 400ml, 500ml, 600ml, 700ml, 800ml, 900ml y 1l
Liquidotomado8			Almacena el dato referido a la cantidad de líquido tomado entre las 8:00 y las 10:00.	
Liquidotomado10			Almacena el dato referido a la cantidad de líquido tomado entre las 10:00 y las 12:00.	
Liquidotomado12			Almacena el dato referido a la cantidad de líquido tomado entre las 12:00 y las 14:00.	
Liquidotomado14			Almacena el dato referido a la cantidad de líquido tomado entre las 14:00 y las 16:00.	
Liquidotomado16			Almacena el dato referido a la cantidad de líquido tomado entre las 16:00 y las 18:00.	
Liquidotomado18			Almacena el dato referido a la cantidad de líquido tomado entre las 18:00 y las 20:00.	
Liquidotomado20			Almacena el dato referido a la cantidad de líquido tomado entre las 20:00 y las 22:00.	
Liquidotomado22			Almacena el dato referido a la cantidad de líquido tomado entre las 22:00 y las 00:00.	
orinado6			Almacena el dato referido a la cantidad de orina expulsada entre las 6:00 y las 8:00.	Los valores obtenidos pueden ser 0ml, 100ml,

Orinado8	VARCHAR	50	Almacena el dato referido a la cantidad de orina expulsada entre las 8:00 y las 10:00.	200ml, 300ml, 400ml y 500ml
Orinado10			Almacena el dato referido a la cantidad de orina expulsada entre las 10:00 y las 12:00.	
Orinado12			Almacena el dato referido a la cantidad de orina expulsada entre las 12:00 y las 14:00.	
Orinado14			Almacena el dato referido a la cantidad de orina expulsada entre las 14:00 y las 16:00.	
Orinado16			Almacena el dato referido a la cantidad de orina expulsada entre las 16:00 y las 18:00.	
Orinado18			Almacena el dato referido a la cantidad de orina expulsada entre las 18:00 y las 20:00.	
Orinado20			Almacena el dato referido a la cantidad de orina expulsada entre las 20:00 y las 22:00.	
Orinado22			Almacena el dato referido a la cantidad de orina expulsada entre las 22:00 y las 00:00.	
escapesnocturnos	VARCHAR	50	Almacena la frecuencia con la que tiene escapes nocturnos.	Los valores obtenidos pueden ser nunca, 1o2 (1 o 2 veces a la semana), 2omas (Más de 2 veces a la semana)
observaciones	VARCHAR	100	Apartado donde el paciente aportará datos relevantes acerca de su estado u opiniones acerca del estudio.	Texto enviado por el paciente

Tabla 8 SP\_Diario.

### Tabla SP\_MENSUAL

Almacena los resultados de la encuesta “Seguimiento de la paciente con vejiga hiperactiva en la consulta de enfermería de suelo pélvico del CHUIMI”

Nombre	Tipo dato	Longitud	Descripción	Valores
Id	INT	11	Auto incrementable que almacena la identificación del cuestionario mensual.	
idspaciente	INT	11	Relaciona la encuesta con un paciente de la tabla SP_PACIENTE.	
frecuenciaperdida	INT	2	Almacena la frecuencia con la que el paciente tiene esas pérdidas.	Los valores obtenidos son del 0 al 5 correspondiendo a nunca, una vez a la semana, 2-3 veces semana, una

				vez al día, varias veces al día y a todas horas respectivamente.
cantidadperdida	INT	2	Almacena el dato de la cantidad de pérdidas con respuesta nada, poco, moderada o mucha cantidad.	Los valores obtenidos son 0, 2, 4 y 6 correspondiendo a no se me escapa nada, muy poca cantidad, una cantidad moderada y mucha cantidad respectivamente
tipoperdida	VARCHAR	50	Almacena el tipo de pérdida, como chorro o gota.	Los valores obtenidos pueden ser chorr o gota.
afectandiario	INT	2	Almacena el dato del 1 al 10 de cuanto afectan las pérdidas en la vida diaria del paciente.	Los valores obtenidos van del 1 al 10.
proteccionescapes	TINYINT	1	Dato que indica si el paciente usa algún tipo de protección contra escapes de orina.	0 para el caso negativo y 1 para caso afirmativo.
tipoproteccion	VARCHAR	50	Almacena el tipo de protección que el paciente usa contra esas pérdidas.	Los valores obtenidos pueden ser ropin (ropa interior), compr (compresas), compN (compresa de noche), salva(salvaslip), panal (pañal).
cantidadproteccion	INT	2	Almacena la cantidad de protección usada en el día.	Los valores van del 1 al 10.
dolormiccion	VARCHAR	50	Almacena la respuesta a la pregunta si tiene dolor antes o durante la micción.	Los valores pueden ser: no, antes, duran o ambos para las respuestas, No, no siento dolor de ningún tipo, Sí, antes de orinar, Sí, durante la micción y Sí, ambos.
ejerciciofisico	TINYINT	1	Almacena la respuesta de si hace ejercicio físico habitualmente.	0 para caso negativo 1 para caso positivo.
tipoejercicio	VARCHAR	11	Tipo de ejercicio físico que hace habitualmente, siendo el campo Otro, otro tipo de ejercicio o varios.	Los valores pueden ser camin, corre, yoga, pilat, Kegel o -oth- siendo caminar, correr, yoga, pilates, ejercicios de Kegel u otro tipo de ejercicio respectivamente.
liquidoingerido	VARCHAR	11	Indica la cantidad de líquido ingerido al día de 0.5L a más de 3L.	05y1L, 1u15L, 2y25L o 3Lmas.
irritantestabac	TINYINT	1	Dato que nos dice si el paciente toma tabaco de forma habitual.	
irritantecafe			Dato que nos dice si el paciente toma café de forma habitual.	
irritantenesca			Dato que nos dice si el paciente toma nescafé de forma habitual.	

irritantete			Dato que nos dice si el paciente toma te de forma habitual.	0 en caso negativo 1 en caso afirmativo.
irritantechoco			Dato que nos dice si el paciente toma chocolate de forma habitual.	
irritantegaseo			Dato que nos dice si el paciente toma bebidas gaseosas de forma habitual.	
irritantealcoh			Dato que nos dice si el paciente toma alcohol de forma habitual.	
irritantesacar			Dato que nos dice si el paciente toma sacarina de forma habitual.	
irritantepican			Dato que nos dice si el paciente toma comidas picantes de forma habitual.	
irritantenada			Dato que nos dice que el paciente no toma las sustancias irritantes mencionadas.	
orinanocturna	VARCHAR	50	Dato de las veces que el paciente se levanta a orinar por la noche.	Los datos recibidos pueden ser nunca, 1sem, 2o3se, noche o nocM para las respuestas nunca, una vez a la semana, varias a la semana, cada noche respectivamente.
cuandoperdidasnunca	TINYINT	1	Dato que indica que el paciente no pierde orina.	0 en caso negativo 1 en caso afirmativo.
cuandoperdicasantwc			Pierde orina antes de llegar al WC.	
cuandoperdidastos			Pierde orina cuando tosa o estornuda.	
cuandoperdidasduerm			Pierde cuando duerme.	
cuandoperdidasfisic			Pierde orina cuando hace esfuerzos físicos.	
cuandoperdidasdespu			Pierde orina al acabar de orinar y ya se ha vestido.	
cuandoperdidasaleat			Pierde orina sin un motivo evidente.	
cuandoperdidasconti			Pierde orina de forma continua.	
relacionessexuales	TINYINT	1	Almacena si el paciente mantiene relaciones sexuales de forma habitual.	0 en caso negativo 1 en caso afirmativo.
dolorrelaciones	TINYINT	1	Almacena si el paciente siente dolor con las	0 en caso negativo 1 en caso afirmativo.

			relaciones sexuales de forma habitual.	
perdidasenelsexo	TINYINT	1	Almacena si el paciente tiene pérdidas de orina con las relaciones sexuales de forma habitual.	0 en caso negativo 1 en caso afirmativo.
miccionesdiarias	VARCHAR	50	Dato que indica la frecuencia con la que el paciente va al baño en función temporal.	30min, 45min, 60min, 75min, 90min, 105mi, 120mi, 3hora.
Diabetico	TINYINT	1	dato que indica si el paciente es diabético o no.	0 en caso negativo 1 en caso afirmativo.
medicamiento	TINYINT	1	Indica si el paciente tiene recetado tratamiento farmacológico contra los escapes o no.	0 en caso negativo 1 en caso afirmativo.
tipomedicabetmimimiento betmi	TINYINT	1	Nos dice si el paciente toma Betmiga.	0 en caso negativo 1 en caso afirmativo.
tipomedicamientotovia			Nos dice si el paciente toma Toviaz.	
tipomedicamientovesic			Nos dice si el paciente toma Vesicare.	
tipomedicamientokente			Nos dice si el paciente toma Kentera parches.	
tipomedicamientourotr			Nos dice si el paciente toma Urotrol.	
tipomedicamientomicto			Nos dice si el paciente toma Mictonorm.	
tipomedicamientonada			Nos dice si el paciente toma Ninguno.	
tratamientomedico	TINYINT	1	Indica si el paciente se toma el medicamento de forma regular.	0 en caso negativo 1 en caso afirmativo.
orinado6	VARCHAR	50	Almacena el dato referido a la cantidad de orina expulsada entre las 6:00 y las 8:00.	Los valores obtenidos pueden ser 0ml, 100ml, 200ml, 300ml, 400ml y 500ml.
Orinado8			Almacena el dato referido a la cantidad de orina expulsada entre las 8:00 y las 10:00.	
Orinado10			Almacena el dato referido a la cantidad de orina expulsada entre las 10:00 y las 12:00.	
Orinado12			Almacena el dato referido a la cantidad de orina expulsada entre las 12:00 y las 14:00.	

Orinado14			Almacena el dato referido a la cantidad de orina expulsada entre las 14:00 y las 16:00.	
Orinado16			Almacena el dato referido a la cantidad de orina expulsada entre las 16:00 y las 18:00.	
Orinado18			Almacena el dato referido a la cantidad de orina expulsada entre las 18:00 y las 20:00.	
Orinado20			Almacena el dato referido a la cantidad de orina expulsada entre las 20:00 y las 22:00.	
Orinado22			Almacena el dato referido a la cantidad de orina expulsada entre las 22:00 y las 00:00.	
Sumaicq	INT	2	Sumatorio de las puntuaciones frecuencia perdida, cantidad perdida y afectandario.	

Tabla 9 SP\_Mensual.

### Tabla SP\_NOTIFICATION

Almacena Las notificaciones de nuestra aplicación

Nombre	Tipo dato	Longitud	Descripción	Valores
Id	INT	11	Código numérico Autoincrementable que almacena la identificacin de la notificación.	
idsppaciente	INT	11	Relaciona la notificación con un paciente de la tabla SP_PACIENTE.	
Fecha	DATETIME		Fecha de envío de la notificación.	
descripción	VARCHAR	500	Contenido de la notificación.	

Tabla 10 SP\_Notification.



## 6. Implementación

A modo de resumen es conveniente recordar las herramientas tecnológicas usadas hasta el momento. El proyecto utiliza Apache, PHP, MySQL y Laravel, integrado con jQuery y Bootstrap, además el entorno de desarrollo interactivo empleado es PhpStorm. En este capítulo profundizaremos en el apartado de codificación en el que pretende explicar las operaciones más importantes de nuestro proyecto.

### 6.1 Diagrama de conexión

A continuación, se muestra el diagrama de conexión de todos los módulos por los que está compuesta la aplicación.

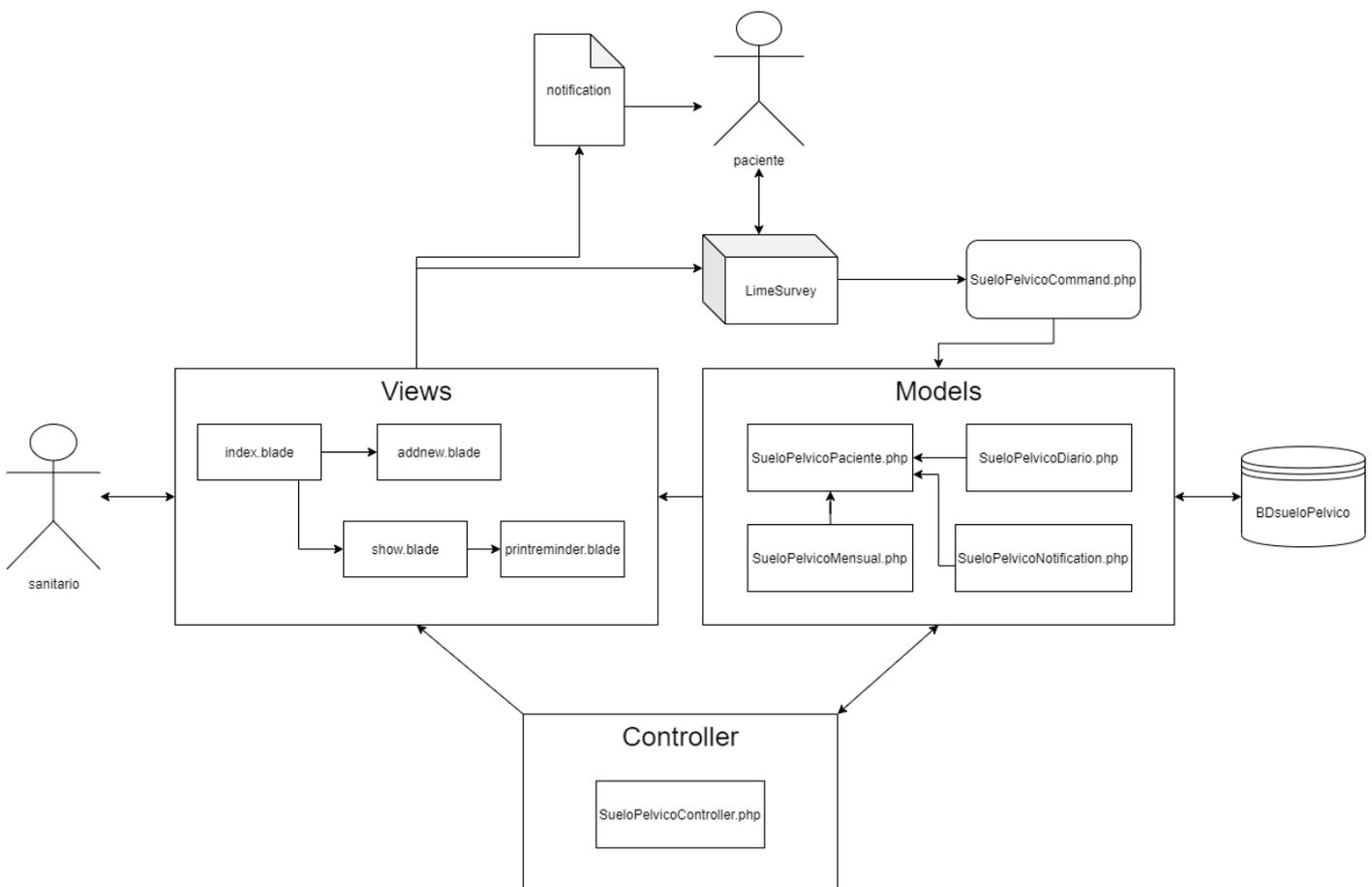


Figura 23 Diagrama de módulos, conexión sanitario-paciente.

El recorrido del flujo de información en la imagen mostrada se realiza desde que el sanitario envía la notificación con el código identificativo al paciente para entrar en la encuesta, tras realizar el formulario la información enviada por el paciente es almacenada en la base de

datos creándose un modelo de paciente, que podrá verse representado en las vistas gracias al controlador donde el personal sanitario podrá interactuar con las funciones de la app. A continuación, se explica cada módulo en mayor profundidad.

## 6.2 Views

Una parte fundamental en la tecnología que utiliza Laravel es el uso de Blade [25], que es un motor de diseño de plantillas que a diferencia de otros motores de plantillas PHP populares, Blade no le impide usar código PHP simple en sus **vistas**. De hecho, todas las vistas de Blade se compilan en código PHP simple y se almacenan en caché hasta que se modifican, lo que aporta una menor carga en la aplicación.

### 6.2.1. Index.Blade.php

Ahora que se sabe lo que es Blade en Laravel nos disponemos a crear la primera vista de nuestra aplicación. Para ordenar de manera sencilla los pacientes que formaran parte del estudio se han creado una serie de tablas en las vistas que componen nuestra app, a continuación, se muestra un ejemplo de ella perteneciente a *index.blade.php*:

```
var resultados=$( '#resultados' ).DataTable({
  "columns": [
    {"data": "encuesta"},
    {"data": "fecultencuesta", "render": $.fn.dataTable.render.moment(
'YYYY-MM-DD HH:mm:ss', 'DD/MM/YYYY' )},
    {"data": "nhc"},
    {"data": "nombre"},
    {"data": "apellido1"},
    {"data": "apellido2"},
    {"data": "fecultcita", "render": $.fn.dataTable.render.moment( 'YYYY-
MM-DD HH:mm:ss', 'DD/MM/YYYY' )},
    {"data": "fecproxcita", "render": $.fn.dataTable.render.moment( 'YYYY-
MM-DD HH:mm:ss', 'DD/MM/YYYY' )},
    {"data": "telefono"}],
  "paginate": true,
  "filter": true,
  "sortClasses": false,
  "sort": true,
  "lengthChange": true,
  "autoWidth": false,
  "paginationType": "full_numbers",
  "processing": true,
  "order": [[1, 'desc']]
});
```

DataTables es una extensión de jQuery que nos permite pintar tablas con paginado, búsqueda, ordenar por columnas, etc. Como se puede observar los datos que contiene nuestra tabla siguen la nomenclatura “data”:”dato del paciente”. Siguiéndole después los parámetros específicos de la tabla como *paginate*, que serviría para numerar las páginas de nuestra tabla o *filter* que nos añadiría funciones de búsqueda pudiendo filtrar conjuntos de letras en la tabla. Existen multitud de variaciones en los parámetros en nuestras tablas como se pueden ver con más detalle en [32].

Ya que esta será la pantalla de inicio de nuestra aplicación deberán crearse una serie de botones con las rutas que nos lleven a las siguientes pantallas. Desde *index.blade.php* se podrá direccionar a *show.blade.php* y a *add.blade.php* a través del siguiente código:

```
//Al hacer click sobre una fila de la tabla abrimos la ventana de detalle
$('#resultados tbody').on('click', 'tr[role="row"]', function () {
    modalLoading();
    window.document.location =
    '{{URL::route('suelopelvico_mostrar')}}/'+resultados.row(
    $(this).closest('tr')).data().id;
});
//boton de añadir
$('#badd').on('click', '', function () {
    modalLoading();
    window.document.location = '{{URL::route('suelopelvico_anadir')}}';
});
```

Para acceder a la pantalla de detalle del paciente no se hace desde un botón estándar, se ha dispuesto para que clicando sobre la fila del paciente que se quiera consultar nos lleve al *show.blade.php* con los datos del paciente seleccionados. Cabe destacar que tanto para el acceso a la ventana de detalle del paciente como para acceder a la pantalla donde podremos añadir nuevos participantes se ha añadido la función *modalLoading()* que nos servirá para cambiar entre pantallas, siendo esta una ventana de carga que bloquea durante un instante la posibilidad de interactuar con la aplicación para evitar que se pulsen varias veces los botones mientras se está direccionando.

### 6.2.2. Show.Blade.php

La pantalla de detalle *show.blade.php* seguirá una estructura parecida a su predecesora *index.blade.php* donde en lugar de mostrar un listado de pacientes, se mostrará a través de una tabla el listado de las encuestas enviadas por el paciente especificado. Además, se pueden ver en detalle los datos personales del paciente, incluso pudiendo modificar algunos de estos parámetros como el email o el teléfono.

Esta pantalla contendrá tres apartados diferenciados donde podrá consultar los datos del paciente, el listado de las encuestas enviadas por el paciente y un listado de notificaciones enviadas al paciente. Además de los apartados de consulta, desde esta pantalla se podrán realizar una serie de acciones que interactuarán con nuestra aplicación, estas son la acción de imprimir recordatorio y la de eliminar paciente del seguimiento. Imprimir recordatorio nos llevará a una nueva pantalla donde gracias al comando `window.print()` se podrá realizar una copia impresa en papel que el paciente podrá llevarse a casa a modo de recordatorio donde estará la `url` del enlace a la encuesta, los plazos en el que debe enviar la encuesta y los números de teléfono de consulta del hospital. También se habilita un botón con la funcionalidad de eliminar al paciente del seguimiento, dado que habrá situaciones en las que la paciente deje de ir a consulta o deje de enviar encuestas y deberemos darle de baja en la aplicación.

La peculiaridad de la tabla de datos que encontramos en `show.blade.php` con respecto a la que se puede ver en la página principal es que se los datos de las encuestas se muestran a través de un desplegable insertado en la tabla:

```
//Cuando se hace click en el + de la tabla de encuestas se despliega y se
formatea la información
$('#surveys tbody').on('click', 'td.details-control', function () {
    var tr = $(this).closest('tr');
    var row = surveysDatatable.row( tr );
    //Si está abierta se cierra
    if(row.child.isShown()){
        row.child.hide();
        tr.removeClass('shown');
    } //Si no, se abre la fila
    }else{
        child='';
        if(tr.data('survey-tipo')==='mensual'){
            child=formatMensual(tr.data('survey'))
        }else{
            child=formatDiario(tr.data('survey'))
        }
        row.child( child ).show();
        tr.addClass('shown');
    }
});
```

### 6.2.3. Addnew.Blade.php

Para que se pueda ampliar el registro de pacientes se debe incluir una forma de añadir nuevos participantes dado que con el paso del tiempo estos registros pueden incrementar

mucho su tamaño. Esta ventana se usará para añadir nuevos participantes al seguimiento desde la base de datos centralizada del servicio de suelo pélvico.

El primer paso para la elaboración de esta vista será la de construir una tabla donde se encuentren todas las pacientes registradas en las agendas de UROENF1 y UROENF2, registro perteneciente a todas las pacientes que han pasado por la consulta de suelo pélvico a lo largo del tiempo.

```
//Click en la fila de la tabla
$('#resultados tbody').on( 'click', 'tr', function () {
    //desmarcar todos los marcados
    $('#resultados tr.selected').toggleClass('selected');
    //marcar el seleccionado
    $(this).toggleClass('selected');
    $('#badd').prop('disabled',false);
} );
```

Se podrá seleccionar la fila perteneciente al paciente que se quiere incluir a nuestra base de datos haciendo click sobre esta, para que después se pueda habilitar el botón de añadir al seguimiento y que aparezca en la tabla de *index.blade.php*.

```
//Botón añadir
$('#badd').on('click', '', function () {
    modalLoading();
    //Se envía los datos de la fila para que se cree el nuevo
    SueloPelvicoPaciente
    $.ajax("{{URL::route('suelopelvico_crear')}}", {
        data: JSON.stringify($('#resultados tr.selected').data('patient')),
        method: "POST",
        success: function() { window.document.location =
    "{{URL::route('suelopelvico')}}"; },
        contentType: "application/json"
    });
});
```

## 6.3 Controller

En lugar de definir toda la lógica de las peticiones en el archivo rutas es posible organizar este comportamiento usando clases tipo Controller. Los **Controladores** pueden agrupar las peticiones HTTP relacionada con la manipulación lógica en una clase.

Nuestro controlador utilizará principalmente los métodos GET y POST. Los métodos GET los usaremos para realizar solicitudes de un recurso a través de una ruta al servidor web. Y los métodos POST los usaremos para enviar datos, por medio de formularios, al servidor web.

A continuación, se explicarán las funciones más importantes que emplea nuestra aplicación desde el controlador *sueloPelvicoController.php*:

### SueloPelvicoController@index:

Es el método inicial de las rutas, lo usamos para mostrar una vista como página principal que contiene nuestra tabla con la agrupación de los pacientes.

```
public function index()
{
    return View::make('suelopelvico.index');
}
```

### SueloPelvicoController@show:

Controlador de la vista de en detalle de los pacientes seleccionados por id, esta muestra el paciente en seguimiento con sus encuestas recibidas y notificaciones.

```
public function show($id)
{
    //Si no existe id se cancela la ejecución
    if (empty($id)) {
        App::abort(400);
    }
    //Registro de la acción para LOPD
    LogChuimi::registerAction("SUELOPELVICO-SHOW|id:"+$id);
    //Generación de la vista
    $paciente=SueloPelvicoPaciente::with('mensuales', 'diarias',
'notificaciones')->find($id);
    return View::make('suelopelvico.show')->with('id', $id)->with('patient',
$paciente);
}
```

Por motivos de seguridad la LOPD en su [Artículo 31 del Reglamento \(UE\) 2016/679 \[33\]](#) exige un registro de las actividades que el usuario realiza en la aplicación, de ahí la función registerAction() que guardará en una base de datos la id del usuario, la acción realizada junto con la fecha y hora en la que se realizó.

### SueloPelvicoController@save

Método utilizado para guardar teléfonos y correos electrónicos de los pacientes indicado por id.

```
public function save($id)
{
    //Prerrequisito, si no existe id se cancela la ejecución
    if (empty($id)) {
        App::abort(400);
    }

    //Buscar el paciente para modificar los campos
    $spPaciente=SueloPelvicoPaciente::findOrFail($id);
    if (Input::get('telefono1')) {
        $spPaciente->telefono1=Input::get('telefono1');
    }
}
```

```

}
if (Input::get('telefono2')) {
    $spPaciente->telefono2=Input::get('telefono2');
}
if (Input::get('email')) {
    $spPaciente->email=Input::get('email');
}
$spPaciente->save();
//Registro de acción para LOPD
LogChuimi::registerAction("SUELOPELVICO-
GUARDAR|id:". $id.",tel1:".Input::get('telefono1').",tel2:".Input::get('telefono2').",email:".Input::get('email'));
//Mensaje de que fué bien usando La Librería de notificaciones
Notification::successInstant(Lang::get('messages.saving.success'));
//Volver a mostrar el paciente
return $this->show($id);
}

```

Con esta función podremos modificar los campos de *telefono1*, *telefono2* y *email* desde *show.blade.php*. Al final de la acción se muestra un mensaje si la función se ha completado correctamente gracias a *successInstant()*.

### SueloPelvicoController@create

Método que se usa para añadir al seguimiento de Suelo Pélvico un nuevo paciente pasado por parámetros.

```

public function create()
{
    //Obtenemos todas las entradas para la solicitud en forma de json
    $input=Input::json()->all();
    //Si ya existe un paciente con el mismo historial lo reemplazamos
    if (SueloPelvicoPaciente::where('nhc', '=', $input['nhc'])->count() > 0)
    {
        $nuevoSueloPelvicoPaciente = SueloPelvicoPaciente::where('nhc', '=',
$input['nhc'])->first();
    } else {
        //Si no, declaramos al nuevo paciente habilitando el seguimiento
        $nuevoSueloPelvicoPaciente = new SueloPelvicoPaciente();
    }
}

```

```

$nuevoSueloPelvicoPaciente->nhc=$input['nhc'];
$nuevoSueloPelvicoPaciente->nombre=$input['nombre'];
$nuevoSueloPelvicoPaciente->apellido1=$input['apellido1'];
$nuevoSueloPelvicoPaciente->apellido2=$input['apellido2'];
$nuevoSueloPelvicoPaciente->telefono1=$input['telefono1'];
$nuevoSueloPelvicoPaciente->telefono2=$input['telefono2'];
$nuevoSueloPelvicoPaciente->fechaultcita=$input['ultimacita'];
$nuevoSueloPelvicoPaciente->fechaproxcita=$input['proximacita'];
$nuevoSueloPelvicoPaciente->fechanacimiento=$input['fecnacimiento'];
$nuevoSueloPelvicoPaciente->seguimiento=1;
$nuevoSueloPelvicoPaciente->save();
//Registro de acción para LOPD
LogChuimi::registerAction("SUELOPELVICO-CREAR|nhc:". $input['nhc']);
Notification::successInstant(Lang::get('messages.saving.success'));

```

```

    return Redirect::route('suelopelvico');
}

```

Con esta función podremos crear nuevos *SueloPelvicoPacientes* con todos sus parámetros y habilitando su seguimiento para después guardarlos en nuestra base de datos.

### SueloPelvicoController@add

Vista para seleccionar un paciente de los que se encuentra en las agendas definidas por SQL\_PATIENTS.

```

public function add()
{
    //Almacenar la consulta 60 minutos en Cache
    $resultados=Cache::remember('users', 60, function () {
        return DB::connection('hce')-
>select(DB::raw(Setting::get('suelopelvico.sql')));
    });

    return View::make('suelopelvico.addnew')->with('resultados',
$resultados);
}

```

Esta función nos habilitará la vista para agregar nuevos pacientes donde cabe destacar que se tendrá un tiempo límite de 60 minutos para el proceso, sino será redireccionado a la página principal.

### SueloPelvicoController@delete

Función para quitar del seguimiento al paciente indicado por id.

```

public function delete($id)
{
    $spPaciente=SueloPelvicoPaciente::findOrFail($id);
    $spPaciente->seguimiento=0;
    $spPaciente->save();
    //Registro de acción para LOPD
    LogChuimi::registerAction("SUELOPELVICO-BORRAR|id:". $id);
    //Mensaje de que fué bien usando La Libreria de notificaciones
    Notification::successInstant(Lang::get('messages.deleting.success'));
    //Volver al listado de pacientes
    return $this->index();
}

```

Esta acción cambiará el valor de seguimiento de nuestro paciente de 1 a 0, lo que supondrá que no podamos verlo en la pantalla de *index.blade.php*, pero manteniendo los datos del paciente en nuestra base de datos para futuros estudios.

### SueloPelvicoController@printReminder

Creación de la vista para crear un recordatorio impreso.

```
public function printReminder($id)
{
    return View::make('suelopelvico.printreminder')->with('id', $id);
}
```

### SueloPelvicoController@search

Buscador de pacientes según su estado.

```
public function search()
{
    //Registro de la acción para LOPD
    LogChuimi::registerAction("SUELOPELVICO-SEARCH");
    //Solo consultamos Los que están en seguimiento
    $results=SueloPelvicoPaciente::where('seguimiento', '=', 1);
    if (Input::get('estado')) {
        $results->where('estado', '=', Input::get('estado'));
    }

    return Response::json($results->get());
}
```

Implementamos la acción de buscar un *sueloPelvicoPaciente* dado un estado a través de un input. Esta acción solo se realiza para pacientes en con seguimiento igual a 1.

## 6.4 Models

Gracias a Laravel podemos hacer uso de un ORM llamado Eloquent [34], un ORM es un Mapeo Objeto-Relacional por sus siglas (Object-Relational mapping), que es una forma de mapear los datos que se encuentran en la base de datos almacenados en un lenguaje de script SQL a objetos de PHP y viceversa, esto surge con la idea de tener un código portable con el que no tengamos la necesidad de usar lenguaje SQL dentro de nuestras clases de PHP.

Eloquent hace uso de los Modelos para recibir o enviar la información a la base de datos, un modelo nos ayuda a definir que tabla, atributos se pueden llenar y que otros se deben mantener ocultos.

A continuación, se explica las relaciones One To Many en Eloquent, que son necesarias para trabajar con nuestros formularios. Por ejemplo, un paciente puede tener varias encuestas mensuales y cada encuesta pertenece a un paciente. Las encuestas deben estar relacionadas por medio de una clave foránea con el paciente.

```

class SueloPelvicoPaciente extends BaseZSeguraModel
{
    protected $table='SP_Paciente';
    public function mensuales()
    {
        return $this->hasMany('SueloPelvicoMensual', 'idsppaciente')->orderBy('fechaencuesta');
    }
}

class SueloPelvicoMensual extends BaseZSeguraModel
{
    protected $table='SP_Mensual';
    protected $guarded = [];

    public function paciente()
    {
        return $this->belongsTo('SueloPelvicoPaciente', 'idsppaciente');
    }
}

```

Esto ocurre de igual forma con *SueloPelvicoDiario* y con *SueloPelvicoNotification*. Ya que en cada modelo se debe añadir estas funciones para que Laravel pueda relacionarlo con el ORM.

## 6.5 Commands

Para evitar la carga de tareas diarias se ha optado por implementar una serie de comandos que realicen varias tareas de forma automática. Por ello, se ha creado el archivo *commands*, cuyo objetivo será realizar la tarea de importación de datos procedentes de la plataforma LimeSurvey y su actualización, junto con el envío de recordatorios a los pacientes de forma periódica y automática.

### Obtención de respuestas desde el servidor LimeSurvey

Antes que nada, se deben obtener las respuestas de las encuestas de los pacientes procedentes del servidor del LimeSurvey, estas respuestas serán almacenadas en una tabla (*\$tableid*) cuyo id sea mayor que el del parámetro *\$lastanswerid*.

```

private function obtenerRespuestasLime($tableid, $lastanswerid)
{
    $resultados=array();
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    curl_setopt($ch, CURLOPT_URL,
Setting::get('suelospelvico.url.download').'?idtabla='.$tableid.'&idrespuesta=
'.$lastanswerid);
    $result = curl_exec($ch);
    curl_close($ch);
}

```

```

$obj = json_decode($result, true);
if (!is_null($obj) && $obj['sucess']) {
    return $obj['data'];
}

```

Debemos establecer conexión con el servidor LimeSurvey por ello haremos uso de las funciones *cURL*, estas permiten conectarse y comunicarse con diferentes tipos de servidores y diferentes tipos de protocolos. Con `curl_init()` iniciamos una nueva sesión *cURL*, con la que se devuelve el manipulador `curl` para el uso de las funciones `curl_setopt()`, `curl_exec()`, y `curl_close()`. Con `curl_setopt()` configuramos una opción para una transferencia *cURL*, estableciendo una opción en la sesión del recurso. Con `curl_exec()` ejecutamos la sesión *cURL* que se le pasa como parámetro. Esta función debe llamarse después de inicializar una sesión *cURL* y todas las opciones para la sesión están establecidas. Y para finalizar `curl_close()` esta función cierra una sesión y libera todos sus recursos.

Una vez establecida la conexión con LimeSurvey guardamos los valores de sus tablas en un archivo json, que decodificaremos y guardaremos en el array `$obj` para tenerlo en formato php.

### Insertar datos transformados en nuestra base de datos

Una vez obtenidos los datos de las encuestas del servidor LimeSurvey hay que hallar el modo de insértalos en nuestra base de datos. Para ello se ha creado la función `insertar`, que depositará los datos contenidos en `$obj` que pertenecen a una encuesta en nuestra base de datos. Dicha función tendrá como parametros `$obj` que serán los datos provenientes del LimeSurvey, `$idtabla` que es la ID de la tabla del LimeSurvey y `$tipo` que nos dirá el tipo de encuesta al que pertenece (Mensual o Diario).

```

private function insertar($obj, $idtabla, $tipo='diario')
{
    //Introducimos el id que tiene en limeSurvey y la fecha de la encuesta
    $params=['idlime'=>$obj['id'], 'fechaencuesta'=>$obj['submitdate']];

    //Obtengo la transformación que voy a usar
    $transformacion=$this->transform[$tipo];

    //Recorro toda la información
    foreach ($obj as $key =>$value) {
        if ($transformacion[$idtabla.$key]['tipo']=='bool') {
            if ($value=='Y') {
                $params[$transformacion[$idtabla.$key]['campo']] = 1;
            } else {
                $params[$transformacion[$idtabla.$key]['campo']] = 0;
            }
        }
    }
}

```

```

    }
    } elseif ($transformacion[$idtabla.$key]['tipo']=='int') {
        if (is_int($value)) {
            $params[$transformacion[$idtabla.$key]['campo']] =
intval($value);
        } else {
            throw new Exception('SueloPelvico: Error en valor
k:'.$idtabla.$key.'-v:'.$value);
        }
    } else {
        if (!is_null($value)) {
            $params[$transformacion[$idtabla.$key]['campo']] = $value;
        }
    }
}
//Se crea el objeto con los parámetros recopilados
if ($tipo=='mensual') {
    $params['sumaicq']=$params['frecuenciaperdida']+$params['cantidadperdida']+$p
arams['afectandiarior'];

    SueloPelvicoMensual::create($params);
} else {
    SueloPelvicoDiario::create($params);
}

//Devuelvo el id de la encuesta de Lime
return $obj['id'];
}

```

En esta función se debe hacer una transformación de los datos obtenidos para adaptarlos a nuestra base de datos ya que nuestra aplicación trabaja con una nomenclatura diferente a la que se usa en el LimeSurvey, por ello recorreremos toda la información realizando los cambios. Por ejemplo, las preguntas de las encuestas cuyas respuestas solo pueden ser “Sí” o “No” que provienen del LimeSurvey sus resultados vienen representados por “Y” para el caso afirmativo y “N” para el negativo, sin embargo, para nuestra aplicación se debe hacer el cambio a booleano (0 o 1 lógico) para adaptarlo a nuestro entorno de trabajo. De igual manera sucede con algunas respuestas en las que se espera un entero, sin embargo, se envían como conjunto de caracteres desde el LimeSurvey.

Una vez realizada la transformación de los valores, creamos los objetos pertenecientes a las tablas Diario y Mensual con los parámetros recopilados. Cabe añadir que para la encuesta mensual se incorpora un cálculo adicional, *sumaicq* que se hace con la suma numérica de los valores que el paciente pone en *frecuenciaperdida* (0 al 5), *cantidadperdida* (0, 2, 4 y 6) y *afectandiarior* (1 al 10).

## Cálculo del estado del paciente

Para el cálculo del estado del paciente con respecto al envío de encuestas se ha creado una función que devuelve el estado según las fechas pasadas, teniendo en cuenta que el estado viene definido por el tiempo transcurrido entre la última cita o encuesta mensual/diario respondido. El Diario se debe responder a los 7, 14, y 21 días desde la última cita. Mientras que el Mensual se debe responder a los 28 días desde la última cita. Si se excede en 1 día el estado es "Fuera plazo".

```
private function calcularEstado($fehaultcita, $fecultencuesta)
{
    $fehault=$fehaultcita;
    if (!is_null($fecultencuesta)) {
        $fehault=$fecultencuesta;
    }

    $dias=Carbon::now()->diffInDays($fehault);
    if ($dias > 7) {
        return "fueraaplazo";
    } elseif ($dias <= 7) {
        if (!is_null($fecultencuesta)) {
            return "resp";
        }
        return "noresp";
    }
}
```

Se utiliza la clase *Carbon* que es una extensión de la clase *dateTime* para contar los días transcurridos tras las respuestas, si a lo largo de la semana se ha respondido a la encuesta se devuelve resp (respondido), si no se respondió en el plazo establecido se le considera como fueraaplazo (fuera de plazo) y para el caso en el que todavía no ha respondido pero está en plazo para hacerlo se le considera noresp (no respondido).

## Creación de notificaciones

Para recordar al paciente sobre su estado y para avisar al sanitario de determinadas acciones se ha de construir algunas notificaciones, por ello se ha creado la función `crearNotificacion()`:

```
private function crearNotificacion($id, $descripcion)
{
    $spnotificacion=new SueloPelvicoNotificacion();
    $spnotificacion->idsppaciente=$id;
    $spnotificacion->descripcion=$descripcion;
    $spnotificacion->save();
}
```

## Método de notificación SMS

Como medida recordatoria para el paciente se ha diseñado un método de envío de notificaciones en el que el paciente podrá recibir las notificaciones por medio del correo electrónico y a través del móvil por SMS:

```
protected function notificarPorTelefono($telefono, $idpaciente)
{
    //Registrar envio de SMS
    if (!is_null($telefono)) {
        //Eliminamos el +34
        $telefono=trim(str_replace('+34', '', $telefono));
        //eliminamos el 0034
        $telefono=trim(str_replace('0034', '', $telefono));
        //Comprobamos que empieza por 6 o 7 para saber que es un teléfono
        móvil
        $digito1=substr($telefono, 0, 1);
        if ($digito1==='6' || $digito1==='7') {
            SMSManager::sendSMS($telefono,
                'Consulta de Suelo Pélvico recordatorio debe enviar
                encuesta para entrar:
                '.Setting::get('suelopelvico.url.survey').' Su núm de
                seguimiento es: '.$idpaciente);
        }
        $this->crearNotificacion($idpaciente, 'Enviado SMS recordatorio
        a: '.$telefono);
        $notificado=true;
    }
    return $notificado;
}
```

Dado que el envío de SMS solo puede ser a través de un teléfono móvil se han de cumplir ciertos requisitos. Para empezar, debemos asegurarnos que el número de teléfono móvil empiece por 6 o 7 correspondiendo a un número de teléfono procedente de España, por tanto, debemos eliminar el +34 o el 0034 en el comienzo de cada número para seguir una única nomenclatura. Una vez confirmado el comienzo del número podemos enviar al paciente el SMS con el contenido por medio de la función sendSMS() que es una función interna del proyecto del hospital que emula a un teléfono móvil pudiendo enviar mensajes de texto a números de teléfonos metidos por parámetros. El SMS a enviar tendrá este formato:

Consulta de Suelo Pélvico recordatorio de formulario Diario|Mensual pulse para entrar: <https://bit.ly/3i8caEi> Su núm de seguimiento es: "id".

## Método Fire

Esta función ejecuta la consola de comandos, lo que la convierte en la función más importante de la clase, esta utiliza todas las funciones anteriormente explicadas ajustándolas a los requisitos que se piden para nuestra aplicación.

```
public function fire()
{
    $this->init();
    //Importar datos del Lime
    //Diarios
    $respuestas=$this-
>obtenerRespuestasLime(Setting::get('suelopelvico.diario.id'),
Setting::get('suelopelvico.diario.ultimoid'));
    if (count($respuestas)>0) {
        foreach ($respuestas as $encuesta) {
            Setting::set('suelopelvico.diario.ultimoid', $this-
>insertar($encuesta, Setting::get('suelopelvico.diario.id'), 'diario'));
        }
    }
    //Mensuales
    $respuestas=$this-
>obtenerRespuestasLime(Setting::get('suelopelvico.mensual.id'),
Setting::get('suelopelvico.mensual.ultimoid'));
    if (count($respuestas)>0) {
        foreach ($respuestas as $encuesta) {
            Setting::set('suelopelvico.mensual.ultimoid', $this-
>insertar($encuesta, Setting::get('suelopelvico.mensual.id'), 'mensual'));
        }
    }
}
```

En esta parte de la función hacemos uso de obtenerRespuestasLime() para conectarnos al servidor del LimeSurvey donde enviamos desde la primera hasta última id de las tablas procedentes del LimeSurvey, luego recorreremos las id y con insertar() vamos adjuntando tabla por tabla desde la primera hasta la última.

```
$pacientes=SueloPelvicoPaciente::where('seguimiento', '=', 1)->get();
foreach ($pacientes as $paciente) {
    //Obtener fechas de citas para el paciente
    $ultfechas=DB::connection('hce')-
>select(DB::raw(Setting::get('suelopelvico.sql')." AND PROG.NHC='". $paciente-
>nhc. "'"));
    $paciente->fehaultcita=$ultfechas->ultimacita;
    $paciente->fechaproxcita=$ultfechas->proximacita;
    //Obtener fecha de la última encuesta
    $diarios=DB::table('SP_Diario')->where('idsppaciente', '=', $paciente-
>id);
    $fechaencuesta=DB::table('SP_Mensual')->where('idsppaciente', '=',
$paciente->id)->union($diarios)->max('fechaencuesta');
    $paciente->fecultencuesta=$fechaencuesta;
    $paciente->estado=$this->calcularEstado($paciente->fehaultcita,
$paciente->fechaproxcita, $paciente->fecultencuesta);
    $paciente->save();
}
```

En el anterior fragmento de código iremos actualizando las fechas de las últimas citas y últimas encuestas recibidas, así como los estados de los pacientes. Para ello nos conectaremos a la base de datos donde se almacenan las citas e importaremos las últimas a nuestra aplicación, para la obtención del estado haremos uso de `calcularEstado()` visto anteriormente.

```
//Enviar SMS y Mails
$pacientes=SueloPelvicoPaciente::where('seguimiento', '=', 1)-
>where('estado', '=', 'fueraPlazo')->get();
foreach ($pacientes as $paciente) {
    $notificado=false;
    //Enviar mail
    if (!is_null($paciente->email)) {
        $mail=$paciente->email;
        Mail::send('suelopelvico.mail', array('id'=>$paciente-
>id,'url'=>Setting::get('suelopelvico.url.survey')), function ($message) use
($mail) {
            $message->to($mail)->subject('CHUIMI Consulta de Seguimiento de
Suelo Pélvico');
        });
        $this->crearNotificacion($paciente->id, 'Enviado email recordatorio
a:'.$paciente->email);
        $notificado=true;
    }
}
```

Para finalizar definimos el envío de SMS y emails a los pacientes en seguimiento, que están fuera de plazo y que no hayan sido notificados. Para el envío de correos electrónicos recorreremos dichos pacientes y utilizamos `send()` que es un método propio de Laravel para enviar de emails y `crearNotificacion()` para la insertar la información del email enviado.

```

        $notificado=$notificado||$this->notificarPorTelefono($paciente-
>telefono1, $paciente->id);
        $notificado=$notificado||$this->notificarPorTelefono($paciente-
>telefono2, $paciente->id);

    //Si no se ha enviado por no existir método notificación lo
registramos
    if (!$notificado) {
        $this->crearNotificacion($paciente->id, 'No se pudo enviar
recordatorio por no existir email o teléfono válido.');
```

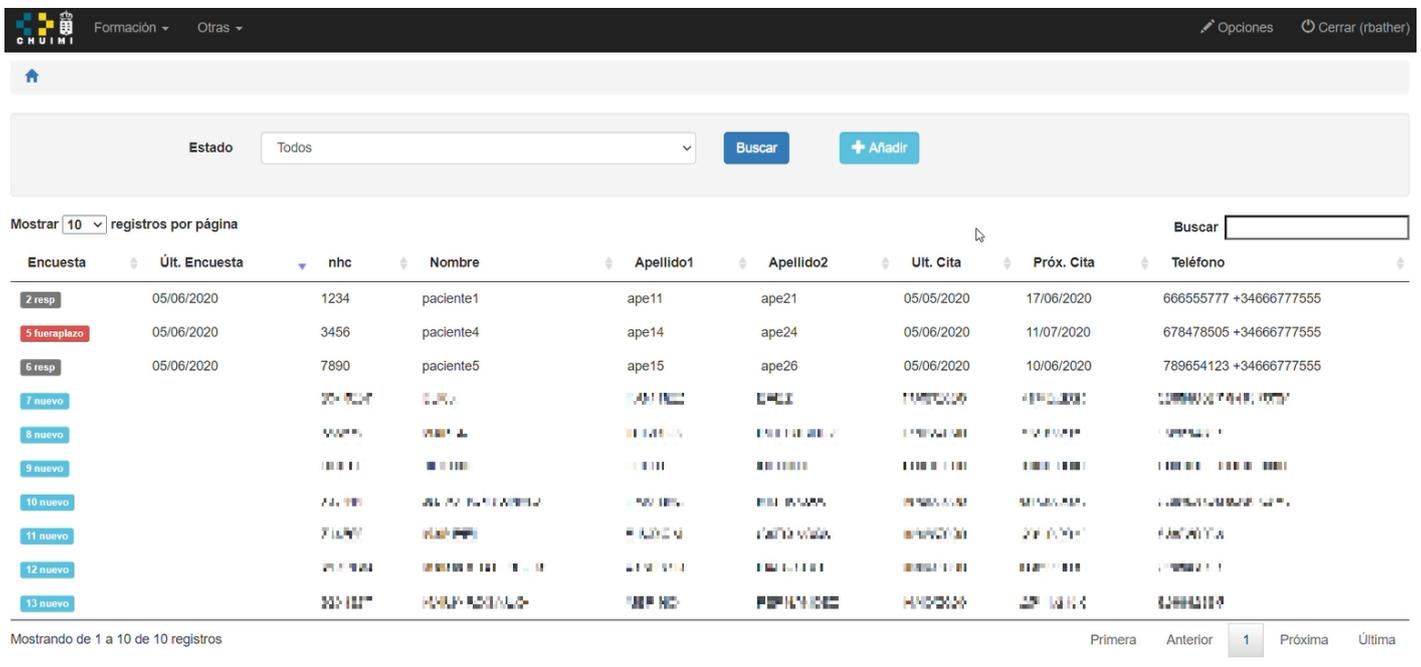
Después nos aseguramos si ha sido notificado por algún medio (email, teléfono1 o teléfono2) si no creamos una notificación para el sanitario diciendo que no se ha podido enviar el recordatorio.

## 7. Pruebas y resultados

En este capítulo se va a realizar una guía de funcionamiento de la aplicación, donde se pondrá a prueba el comportamiento del sistema en todas sus vistas y operaciones con el fin de evaluar el correcto funcionamiento de la aplicación diseñada.

### Ventana de la página principal

Se prueba la página principal de la aplicación, que es la parte de la aplicación donde se quiere dar una visión general de todos los pacientes que están en seguimiento.



Mostrar 10 registros por página

Encuesta	Últ. Encuesta	nhc	Nombre	Apellido1	Apellido2	Ult. Cita	Próx. Cita	Teléfono
2 resp	05/06/2020	1234	paciente1	ape11	ape21	05/05/2020	17/06/2020	666555777 +34666777555
5 fueraplazo	05/06/2020	3456	paciente4	ape14	ape24	05/06/2020	11/07/2020	678478505 +34666777555
6 resp	05/06/2020	7890	paciente5	ape15	ape26	05/06/2020	10/06/2020	789654123 +34666777555
7 nuevo								
8 nuevo								
9 nuevo								
10 nuevo								
11 nuevo								
12 nuevo								
13 nuevo								

Mostrando de 1 a 10 de 10 registros

Primera Anterior 1 Próxima Última

Figura 24 Vista index.blade.PHP.

Como se puede observar en la figura anterior, se realiza una tabla que almacena a los pacientes en filas, dadas en páginas regulables pudiendo mostrar varios registros (10, 25, 50 y 100), en la imagen anterior se puede observar que hay una única página con 10 pacientes donde se muestran los datos del nombre del paciente, el estado de la encuesta, la fecha de la última encuesta respondida, el número de historia clínica, apellidos, última vez que fue a consulta, próxima cita en consulta y teléfonos. Además, para una mayor facilidad en la clasificación de los pacientes, estos se pueden ordenar según el estado en el que se encuentre (resp, noresp, fueradeplazo y nuevo).

Estado: Respondido [Buscar] [Añadir]

Mostrar 10 registros por página

Encuesta	Últ. Encuesta	nhc	Nombre	Apellido1	Apellido2	Ult. Cita	Próx. Cita	Teléfono
2 resp	05/06/2020	1234	paciente1	ape11	ape21	05/05/2020	17/06/2020	666555777 +34666777555
6 resp	05/06/2020	7890	paciente5	ape15	ape26	05/06/2020	10/06/2020	789654123 +34666777555

Mostrando de 1 a 2 de 2 registros

Primera Anterior 1 Próxima Última

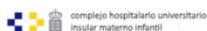


Figura 25 Filtro por estado.

Se ha agregado también la funcionalidad de búsqueda por conjunto de caracteres en el interior de la tabla, pudiendo así buscar de manera sencilla a los pacientes tanto por su nombre o apellidos como por su número de historia clínica, además de poder filtrar cualquier dato en el interior de la tabla como una fecha.

Estado: Todos [Buscar] [Añadir]

Mostrar 10 registros por página

Buscar: paciente1

Encuesta	Últ. Encuesta	nhc	Nombre	Apellido1	Apellido2	Ult. Cita	Próx. Cita	Teléfono
2 resp	05/06/2020	1234	paciente1	ape11	ape21	05/05/2020	17/06/2020	666555777 +34666777555

Mostrando de 1 a 1 de 1 registros (filtrados de 10 total)

Primera Anterior 1 Próxima Última



Figura 26 Búsqueda por caracteres.

Por último, desde esta ventana se podrán acceder a otras dos ventanas distintas, la ventana donde podremos añadir pacientes nuevos, que se podrá acceder a ella pulsando el botón azul añadir y la correspondiente a la vista en detalle de los pacientes, que llegaremos a ella haciendo click con el ratón en la fila del paciente que se quiera consultar.

### Ventana añadir paciente

En esta ventana podremos añadir nuevos pacientes junto con sus datos clínicos a nuestra página principal para tenerlos en seguimiento, estos pacientes provienen de las agendas definidas por SQL\_PATIENTS, donde hay un registro de todos los pacientes que han pasado por el hospital.

Formación Otras Opciones Cerrar (rbater)

Los pacientes que se muestran provienen de las agendas UROENF1 y UROENF2 de Drago. Escoja una paciente para que el siguiente botón esté disponible. [+ Añadir](#)

Mostrar 10 registros por página

Buscar

NHC	Nombre	Apellido1	Apellido2	Ult. Cita	Próx. Cita	Teléfono
3024442				30/06/2020	03/11/2020	
655074				30/06/2020	17/11/2020	
3024530				30/06/2020	04/11/2020	
3024557				23/06/2020	28/10/2020	
3024564				23/06/2020	16/12/2020	
3024565				17/06/2020	15/12/2020	
655065				17/06/2020	01/09/2020	
655072				17/06/2020	16/09/2020	
3024566				17/06/2020	05/08/2020	
3024568				17/06/2020	18/08/2020	

Mostrando de 1 a 10 de 110 registros

Primera Anterior 1 2 3 4 5 ... 11 Próxima Última

Figura 27 Vista addnew.blade.PHP.

El funcionamiento de esta ventana es sencillo, una vez hemos accedido a ella localizamos al paciente que se quiere incorporar al seguimiento, luego seleccionamos la fila donde se encuentre haciendo click con el ratón sobre ella donde aparecerá con una franja azul indicando que ha sido seleccionada. Una vez seleccionado el paciente pulsaremos el botón añadir ubicado encima de la tabla, donde para concluir acabará mostrándonos un mensaje de guardado correctamente y nos llevará a la ventana principal de nuestra aplicación mostrando el nuevo paciente en su tabla.

**NOTA\*:** En cumplimiento de la LOPD [33] se ha tenido que pixelar el contenido de parte de las tablas debido a que se mostraba datos personales de pacientes reales.

### Ventana de vista en detalle del paciente

En esta ventana se puede ver en detalle los datos de los pacientes de forma individual, añadiendo las notificaciones enviadas y los datos recibidos de las encuestas Mensual y Diaria.

Formación - Otras - Opciones - Cerrar (rbalther)

Imprimir recordatorio - Quitar de seguimiento

Seguimiento 2. Atención paciente fuera de plazo. Última encuesta recibida el 05/06/2020 (26 días desde ult. encuesta).

Datos del Paciente

<b>NHC</b> 1234	<b>NOMBRE APELLIDOS</b> paciente1 ape11 ape21	<b>NACIMIENTO</b> 05/06/2000
<b>ULT. CITA</b> 05/05/2020	<b>PRÓX. CITA</b> 17/06/2020	
<b>TELÉFONO</b> 666555777	<b>TELÉFONO</b> +34666777555	<b>EMAIL</b> paciente1@gmail.com

Guardar

Encuestas recibidas Mensual: 1 Diario: 1

Fecha	Tipo
05/06/2020	Mensual <b>ICQ 1</b>
15/02/2020	Diario

Mostrando de 1 a 2 de 2 registros

Notificaciones enviadas

Fecha	Tipo
18/05/2020	Se ha enviado SMS
17/05/2020	Se ha enviado mail

Mostrando de 1 a 2 de 2 registros

Figura 28 Vista show.blade.PHP.

Esta ventana tiene distintas funciones, para empezar, se mostrará un mensaje en la esquina superior derecha dando a saber el estado en el que se encuentra el paciente referenciándolo por el ID. Luego se muestran las 3 secciones diferentes que componen la ventana:

- Los datos de contacto del paciente que se podrán modificar, como son los teléfonos y el correo electrónico.
- Los datos de las encuestas que el paciente ha enviado señalando la fecha en la que se envió, mostrado en una tabla por filas indicando el tipo de encuesta al que hacen referencia. Estas filas tienen una opción desplegable para ver las respuestas recibidas en cada encuesta.
- Tabla de notificaciones enviadas al paciente indicando la fecha y por el medio en el que se ha enviado.

En la figura 27 se muestra como se ve las respuestas del ejemplo de encuesta mensual una vez desplegada.

Encuestas recibidas Mensual: 1 Diario: 1

Fecha: 05/06/2020 Tipo: Mensual ICQ 1

Frecuencia perdidas: nunca	Cantidad: ninguna	Cuanto afectan: 1	Dolor: duran	Ejercicio físico: Si	Tipo de ejercicio: null	Líquido ingerido: 1u15L			
Toma tabaco: Si	Toma cafe: Si	Toma nescafe: Si	Toma te: Si	Toma chocolate: Si	Toma bebidas gaseosas: Si	Toma alcohol: Si	Toma sacarina: Si	Toma comidas picantes: Si	No toma las irritantes: Si
Veces orinar por la noche: 1sem	No perdidas orina: Si	Pierde orina cuando tosa o estornuda: Si	Pierde cuando duerme: Si	Pierde orina cuando hace esfuerzos físicos: Si	Pierde orina al acabar de orinar y ya se ha vestido: Si	Pierde orina sin un motivo evidente: Si			
Relaciones sexuales: Si	Dolor en relaciones sexuales: Si	Perdidas de orina con las relaciones sexuales: Si	Cada cuanto va al baño: 60min	Paciente es diabetico: Si					
Recetado tratamiento farmacologico contra los escapes: Si	Toma Betmiga: Si	Toma Toviaz: Si	Toma Vesicare: Si	Usa Kentera parches: Si	Toma Urotrol: Si	Toma Mictonorm: Si	No toma ninguno de los medicamentos: Si	Toma el medicamento de forma regular: Si	
Orina 6:00 a 8:00: 200ml	Orina 8:00 a 10:00: 200ml	Orina 10:00 a 12:00: 200ml	Orina 12:00 a 14:00: 200ml	Orina 14:00 a 16:00: 200ml	Orina 16:00 a 18:00: 200ml	Orina 18:00 a 20:00: 200ml	Orina 20:00 a 22:00: 200ml	Orina 22:00 a 00:00: 200ml	

Figura 29 Desplegable encuesta mensual.

En esta misma ventana podemos eliminar del seguimiento al paciente pulsando el botón rojo ubicado en la parte superior de la página, esta acción nos enviara un aviso en forma de ventana emergente donde deberemos confirmar que queremos quitar del seguimiento al paciente. Una vez confirmado seremos redireccionados a la página principal donde ya no se verá el paciente eliminado.

Por último, podemos ver en la parte superior izquierda de la ventana el botón de imprimir recordatorio que nos redireccionará a la ventana *printreminder.blade.php*. Donde podremos hacer una copia impresa del mensaje de la figura 28 indicándonos la identificación numérica con la que debemos rellenar la encuesta.



**Seguimiento de Consulta de Suelo Pélvico**

Se le recuerda que deberá responder al Diario de Suelo Pélvico a los 7, 14 y 21 días. Y al Mensual a los 28 días desde su última cita a la consulta de Suelo Pélvico.

En cualquier caso se le mandará por correo electrónico o SMS un mensaje recordatorio.

La página web donde encontrará las encuestas y a la que podrá acceder desde su tablet, teléfono o ordenador es <https://bit.ly/3l8caEI>. En la encuesta como primer dato se le pedirá que introduzca su número de seguimiento que es: 2

Avda Marítima del Sur, Sn  
35016 – Las Palmas de Gran Canaria  
Telf.: 928 44 40 00  
Telf.: 928 44 45 00

Figura 30 Notificación impresa.



## 8. Conclusiones

---

En este capítulo, se analizan las conclusiones finales del proyecto y se reflexiona sobre las posibles tareas de mantenimiento y extensión de la aplicación. Actualmente, anticiparse a lo que va a suceder o estar preparados para el cambio, forman las bases a tener en cuenta en cualquier aplicación web verdaderamente importante.

### 8.1. Conclusiones

Tras analizar los capítulos de implementación y resultados podemos afirmar que se ha cumplido con todos los objetivos establecidos en un comienzo, se ha diseñado desde su concepción la estructura de una aplicación, se han creado los distintos formularios que componen la aplicación y se ha podido comprobar que la aplicación es totalmente funcional y disponible para el uso del personal del CHUIMI cumpliendo con todos los criterios de seguridad requeridos para su incorporación en la red hospitalaria.

Este trabajo ha consistido en el desarrollo de una aplicación para el seguimiento de pacientes afectados por VHA. Las principales funcionalidades son las siguientes.

**F (1).** - Listar los pacientes inscritos.

**F (2).** - Gestión y búsqueda de pacientes inscritos.

**F (3).** - Consulta de datos de pacientes y de los formularios enviados por estos.

**F (4).** - Añadir nuevos pacientes y eliminar pacientes en seguimiento.

**F (5).** - Generador de notificaciones en físico y por medios telemáticos para el paciente.

La metodología empleada en las reuniones con el paciente antes a la implantación de la aplicación desde el punto de vista del personal sanitario, consistía en citarse con un paciente con posibles problemas de VHA para presentarle un cuestionario que se hacía en papel tratando las cuestiones pertinentes al estado del paciente, para después analizar los resultados y establecer otra cita donde hacer la misma rutina. Si comparamos esta rutina con la que se debe hacer tras la implementación de nuestra aplicación se puede notar grandes beneficios, siendo el principal la eliminación de esas citas donde se invertía tanto tiempo en una tarea tan simple.

Las funcionalidades que ofrecen la implantación de esta tecnología suponen un gran ahorro en tiempo y por tanto en costes para la Unidad de Suelo Pélvico del Servicio Canario de Salud, teniendo la posibilidad de invertir el tiempo que solían emplear en atender a pacientes con esta afección en otro tipo de tareas. Añadiendo también un beneficio directo al paciente ofreciéndole la posibilidad realizar la encuesta en la comodidad de su casa y por tanto evitar desplazamientos al hospital para rellenar un simple formulario.

Además, la aplicación ha sido desarrollada con vistas a poder utilizarse tras su implementación y posterior verificación, lo que tiene un atractivo añadido: crear una aplicación web que va a ser llevada a una fase de producción real y un mantenimiento posterior.

## 8.2. Trabajo futuro

Este diseño plantea expectativas bastante interesantes con respecto al futuro, dado que ha sido creada con la intención de ser una herramienta de trabajo que el personal sanitario de la Unidad de Suelo Pélvico usará de forma diaria. Sin embargo, estamos en una fase temprana tras su desarrollo donde no se puede saber a ciencia cierta todas las posibles modificaciones al uso, puesto que al tiempo que se ha realizado este documento no se ha puesto al servicio del demandante.

A pesar de lo dicho anteriormente, se tiene grandes esperanzas en torno al futuro de esta aplicación, por ello, se plantea la posibilidad de integrar al proyecto funciones para obtener estadísticas sobre aspectos de los pacientes, por ejemplo, rangos de edad de los pacientes asociado al estudio, respuestas más frecuentes etc. Añadiéndolos en gráficas interactivas donde se puedan ver de una forma más sencilla los avances en los pacientes.

Para un futuro también se baraja implantar otro tipo de acceso para los pacientes, cambiando el actual por otro más fiable que ya que este puede producir equivocaciones por parte del paciente al introducir una ID equivocada al responder una encuesta, sin embargo, es una ardua labor dada las restricciones de seguridad a las que se ve sometida la red hospitalaria.

El uso de esta nueva Aplicación Web en la actualidad tiene un valor añadido y es que debido a la pandemia mundial que la sociedad está viviendo desde el 11 de marzo del 2020 por causa del Covid-19 [35] donde cada individuo debe mantener una distancia social y evitar lo

máximo posible el contacto físico, se incrementa la importancia de los medios telemáticos más que nunca, lo que realza la relevancia de la aplicación diseñada que al reducir las visitas al hospital previene de posibles contagios siendo este hecho de mayor importancia para entornos hospitalarios donde es más común que transiten personas de mayor riesgo. Además, la implantación de esta herramienta web da la posibilidad de reubicar a parte del personal sanitario a tareas más importantes relacionadas con el virus. Tener en cuenta que evitar el colapso sanitario y frenar el número de contagios es responsabilidad de todos y usar los medios telemáticos en este tipo de consultas es una alternativa que nos beneficia en gran medida.

Para concluir esta aplicación quiere mejorar en el futuro la facilidad en el seguimiento de pacientes con problemas de Vejiga Hiperactiva aplicando un proceso automatizando ahorrado en personal y tiempo. Cambiando la forma de tratar a dichos afectados con el principal objetivo de mejorar su calidad de vida.



## 9. Referencias

---

- [1] Parasuraman, Muralidharan, y Jaya Raja Kuma, «Mirabegron- a beta-3 adrenergic agonist for overactive bladder syndrome: A review.», pp. 593-596., 2013.
- [2] «Catálogo de aplicaciones sobre la salud», *fundacionisys*, ene. 20, 2020.  
<https://www.fundacionisys.org/es/apps-de-salud/catalogo-de-apps#profesional>.
- [3] J. A. Gallego, I. Grau, y B. Kostov, «Método de valoración de aplicaciones móviles de salud en español: el índice iSYScore», dic. 2016, [En línea]. Disponible en:  
<https://www.elsevier.es/es-revista-semergen-medicina-familia-40-resumen-metodo-valoracion-aplicaciones-moviles-salud-S1138359315004281>.
- [4] R. V. Lerma-Blasco, «Aplicaciones Web». 2013, [En línea]. Disponible en:  
<https://polmirosmix2.files.wordpress.com/2015/06/aplicaciones-web-2013-grado-medio-mcgraw-hill.pdf>.
- [5] «Alojamiento web», 2013. <http://www.tualojamientoweb.com/intro/clases-alojamiento-web.htm>.
- [6] R. Charmane, «Application Layer», oct. 29, 2020.  
<http://mscancer22.tripod.com/applicationlayer/id17.html>.
- [7] C. Sánchez González, «Aplicaciones en capas», sep. 28, 2004.  
<http://oness.sourceforge.net/proyecto/html/ch03s02.html>.
- [8] «Cómo elegir el stack de software adecuado para tu producto», mar. 03, 2020.  
<https://www.codigofuente.org/elegir-stack-software-adecuado-producto/>.
- [9] J. U. Miranda Charles, «Ajax», jul. 09, 2020. <https://jonmircha.com/ajax>.
- [10] D. P. G., «Creación de una aplicación MVC», feb. 24, 2019.  
<https://www.eltallerdesharepoint.com/net/creacion-una-aplicacion-mvc-en-azure/>.
- [11] S. Burbeck, «How to use Model-View-Controller (MVC)», 1992.  
<https://web.archive.org/web/20120429161935/http://st-www.cs.illinois.edu/users/smarch/st-docs/mvc.html>.
- [12] J. D. Ferrer Quintana y V. M. Melián Santana, «Normativa de Gestión de Alojamiento de Aplicaciones». nov. 11, 2019.
- [13] J. D. Ferrer Quintana y V. M. Melián Santana, «Normativa de uso de la Infraestructura corporativa para Aplicaciones del Gobierno de Canarias». abr. 21, 2020.
- [14] J. D. Ferrer Quintana y M. A. Castellano Trujillo, «Normativa sobre requisitos Seguridad en Aplicaciones Web». nov. 07, 2018.
- [15] L. SHENG, W. SHANNON, y S. DARREN, «Rancher Kubernetes», *Rancher*, 2020.  
<https://rancher.com/>.
- [16] I. Murdock, «Debian, The universal operating system», sep. 26, 2020.  
<https://www.debian.org/index.es.html>.
- [17] B. Behlendorf, «Apache server», nov. 20, 2020. <https://www.apache.org/>.
- [18] B. Behlendorf, «ModRewrite, Apache server». 2020, [En línea]. Disponible en:  
[https://httpd.apache.org/docs/2.4/mod/mod\\_rewrite.html](https://httpd.apache.org/docs/2.4/mod/mod_rewrite.html).
- [19] A. Mehdi, B. Friedhelm, y D. Antony, «PHP manual», *Manual PHP*, jul. 27, 2020.  
<https://www.php.net/manual/es/>.
- [20] M. Widenius, «MySQL», 2020. <https://www.mysql.com/>.
- [21] A. Ferry Mestres, «Introducción al frontend y backend», feb. 2018.  
[http://cv.uoc.edu/annotation/a9c35c372dcee6e6b92afad6993cd048/620334/PID\\_00250214/PID\\_00250214.html#:~:text=En%20cuanto%20a%20la%20tecnolog%C3%ADa,datos%20y%20algunas%20otras%20cosas](http://cv.uoc.edu/annotation/a9c35c372dcee6e6b92afad6993cd048/620334/PID_00250214/PID_00250214.html#:~:text=En%20cuanto%20a%20la%20tecnolog%C3%ADa,datos%20y%20algunas%20otras%20cosas).

- [22] J. Manuel Alarcón, «¿Qué necesitas saber en 2020 para ser un buen desarrollador web Front-End?», sep. 03, 2019. <https://www.campusmvp.es/recursos/post/que-necesitas-saber-para-ser-un-buen-desarrollador-web-front-end.aspx>.
- [23] J. D. Gauchat, «El gran libro de HTML5, CSS3 y Javascript». 2012, [En línea]. Disponible en: <https://gutl.jovenclub.cu/wp-content/uploads/2013/10/El+gran+libro+de+HTML5+CSS3+y+Javascrip.pdf>.
- [24] T. Otwell, «Laravel Documentation», 2020. <https://laravel.com/docs/7.x>.
- [25] O. Cordova Aburto y R. Cuevas, «Laravel-5», *gitbooks*, dic. 29, 2017. <https://richos.gitbooks.io/laravel-5/content/>.
- [26] M. Otto, «Bootstrap Documentation», 2020. <https://getbootstrap.com/docs/4.5/getting-started/introduction/>.
- [27] D. Methvin, «jQuery», may 2020. <https://jquery.com/>.
- [28] O. Mark y J. Thornton, *Bootstrap 3, el manual oficial*. 2018.
- [29] S. Sijbrandij, «Gitlab», 2020. <https://about.gitlab.com/>.
- [30] Z. Lurie, «SurveyMonkey». <https://es.surveymonkey.com/>.
- [31] C. Schmitz y J. Cleeland, «LimeSurvey», 2020. <https://www.limesurvey.org/es/>.
- [32] S. Galloway, «DataTables», 2020. <https://datatables.net/manual/>.
- [33] Jefatura de Estado, «Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales.» dic. 06, 2018, [En línea]. Disponible en: <https://www.boe.es/buscar/pdf/2018/BOE-A-2018-16673-consolidado.pdf>.
- [34] T. Otwell, «Eloquent: API Resources». 2020, [En línea]. Disponible en: <https://laravel.com/docs/7.x/eloquent-resources#introduction>.
- [35] Centro de Coordinación de Alertas y y Emergencias Sanitarias, «Información Científica-Técnica. Enfermedad por coronavirus, COVID-19». nov. 12, 2020, [En línea]. Disponible en: <https://www.mscbs.gob.es/profesionales/saludPublica/ccayes/alertasActual/nCov/documentos/ITCoronavirus.pdf#:~:text=El%20d%C3%ADa%2011%20de%20marzo,declar%C3%B3%20la%20pandemia%20mundial>.
- [36] CSIRC, «Administración y uso de la plataforma LimeSurvey». jun. 15, 2020, [En línea]. Disponible en: [https://csirc.ugr.es/informatica/Galerias/OtrosFicheros/manualLimeSurvey\\_CSIRC\\_v5.pdf](https://csirc.ugr.es/informatica/Galerias/OtrosFicheros/manualLimeSurvey_CSIRC_v5.pdf).
- [37] «Manual LimeSurvey», *Manual de LimeSurvey*, sep. 14, 2020. [https://manual.limesurvey.org/LimeSurvey\\_Manual/es](https://manual.limesurvey.org/LimeSurvey_Manual/es).
- [38] «Modelo-vista-controlador», *Wikipedia, la enciclopedia libre*. nov. 21, 2020, [En línea]. Disponible en: <https://es.wikipedia.org/wiki/Modelo%E2%80%93vista%E2%80%93controlador>.
- [39] H. Refsnes, S. Refsnes, y J. E. Refsnes, «HTML, CSS, JAVASCRIPT, SQL Tutorial», *W3school*, 2020. <https://www.w3schools.com/tags/default.asp>.
- [40] «Mapeo Objeto-Relacional», *Wikipedia, la enciclopedia libre*. sep. 19, 2020, [En línea]. Disponible en: [https://es.wikipedia.org/wiki/Mapeo\\_objeto-relacional](https://es.wikipedia.org/wiki/Mapeo_objeto-relacional).
- [41] «Servidor HTTP Apache», *Wikipedia, la enciclopedia libre*. nov. 21, 2020, [En línea]. Disponible en: [https://es.wikipedia.org/wiki/Servidor\\_HTTP\\_Apache](https://es.wikipedia.org/wiki/Servidor_HTTP_Apache).
- [42] J. Eguiluz, «Introducción a AJAX». jul. 18, 2018, [En línea]. Disponible en: <https://uniwebdad.com/libros/ajax>.

## Glosario

---

- AJAX** Asynchronous JavaScript and XML. Tecnología que permite la comunicación entre el servidor y el cliente de forma asíncrona.
- Blade** Sistema de plantillas que proporciona Laravel cuya principal característica es que sus vistas son compiladas en código PHP plano y almacenadas en caché hasta que sean modificadas, lo que significa que Blade no añade sobrecarga a tu aplicación.
- CCN** Centro Criptológico Nacional, referencia a nivel nacional e internacional en materia de ciberseguridad.
- Clúster** Son sistemas distribuidos de granjas de computadoras unidos entre sí normalmente por una red de alta velocidad y que se comportan como si fuesen un único servidor. A diferencia de la computación en malla, los clústeres de computadoras tienen a cada nodo realizando la misma tarea, controlada y planificada por software.
- CSS** Cascading Style Sheets. Lenguaje de formato que permite definir los estilos de una página web.
- DOM** Document Object Model o Modelo de Objetos del Documento, es una interfaz de programación de aplicaciones para documentos válidos HTML. Define la estructura lógica de los documentos y el modo en que se accede y manipula.
- DNS** Domain Name System, es un sistema de nomenclatura jerárquico descentralizado para dispositivos conectados a redes IP como Internet o una red privada. Este sistema asocia información variada con nombres de dominio asignados a cada uno de los participantes.
- Framework** Estructura de soporte definida, en la cual un proyecto de software puede ser organizado y desarrollado en una plataforma o marco de trabajo.
- GIT** Software controlador de versiones. Llevar registro de los cambios en archivos de computadora y coordinar el trabajo que varias personas realizan sobre archivos compartidos.
- HTML** Hyper Text Markup Language. Lenguaje de marcado de contenidos en páginas web basado en etiquetas.
- HTTP** HyperText Transfer Protocol. Protocolo de intercambio de información vía web.
- ICIQ** International Consultation on Incontinence Questionnaire, cuestionario estándar para pacientes con incontinencia urinaria.

- JavaScript** Lenguaje de programación interpretado que se utiliza para crear páginas web dinámicas, es decir, que incorporan efectos y/o acciones.
- JQuery** Librería JavaScript sencilla, eficiente y con multitud de funcionalidades que permite, entre otras cosas, manipular el código HTML de forma dinámica (en tiempo de ejecución).
- JSON** JavaScript Object Notation. Formato ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML.
- LOPD** Ley Organica de Proteccion de Datos. Ley que obliga personas, empresas y organismos, tanto privados como públicos que dispongan de datos de carácter personal a cumplir una serie de requisitos y aplicar determinadas medidas de seguridad en función del tipo de datos que posean.
- Media query** Consultas de medios. Se utilizan para modificar tu página web o aplicación en función del tipo de dispositivo.
- MVC** Modelo Vista Controlador. Patrón de arquitectura de software, utilizado en aplicaciones Web, encargado de separar la lógica de negocio de la interfaz del usuario.
- NHC** Número de Historia Clínica.
- ORM** Object-Relational Mapping, técnica de programación que convierte datos entre sistemas de tipos de un lenguaje de programación orientado a objetos y una base de datos relacional.
- SQL** Structured Query Language. Lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones sobre los datos.
- URL** Uniform Resource Locator. Localizador de recursos uniforme. Secuencia de caracteres, de acuerdo a un formato estándar, que se usa para nombrar recursos en Internet para su localización o identificación.
- VHA** Vejiga Hiperactiva, diagnóstico clínico que se define como la presencia de urgencia urinaria, acompañada generalmente de un aumento de la frecuencia miccional.
- XML** eXtensible Markup Language, traducido como "Lenguaje de Mercado Extensible" es un metalenguaje que permite definir lenguajes de marcas utilizado para almacenar datos en forma legible.

## Presupuesto

---

En este capítulo se recoge el presupuesto total del presente TFG. Para ello se han seguido las recomendaciones del Colegio oficial de Ingenieros de Telecomunicación (a partir de ahora, COIT) sobre los baremos orientativos para trabajos profesionales desde 2009. Este presupuesto se ha organizado en varios apartados para poder diferenciar los distintos costes derivados del desarrollo de este TFG:

1. Recursos materiales.
2. Recursos humanos.
3. Gastos en material adicional.
4. Derechos de visado del COIT.
5. Aplicación de impuestos.

### P.1 Recursos materiales

Debido a que en este proyecto se ha intentado siempre trabajar con un software open-source, sólo se tendrán que tener en cuenta el equipo hardware utilizado, su sistema operativo y los paquetes software que se han usado para la redacción de esta memoria.

El coste de amortización viene estipulado en un plazo de 5 años. Para realizar este cálculo, se hará uso de un sistema de amortización lineal o constante, donde el bien material que se tenga, va perdiendo valor a lo largo de vida útil de una forma constante, quedando como valor residual el valor que se le da a un material una vez que ha sobrepasado su vida útil. Según la ecuación P.1:

$$Cuota\ anual = \frac{Valor\ de\ adquisición - Valor\ residual}{Años\ vida\ útil} \quad (Ec.\ P.1)$$

### P.2 Recursos hardware

Se han utilizado en el proceso de este proyecto las siguientes herramientas hardware (Tabla P.1):

Ordenador portátil MSI de 15.6", con procesador Intel Core i7-8750H CPU @ 2.20GHz, memoria RAM de 16 GB y con 1 TB de capacidad de disco duro.

COSTES DE LAS HERRAMIENTAS HARDWARE				
Descripción	Valor de adquisición	Valor residual	Vida útil	Valor amortización (4 meses)
Ordenador portátil MSI de 15.6"	1100 €	350€	3 años	62.5€

Tabla 11 Costes de las herramientas hardware

### P.3 Recursos software

Las herramientas software que se hayan usado en este proyecto, aparte de las nombradas en el propio proyecto, son:

Microsoft Office 2010 para Windows. Como este software esta suministrado por la ULPGC, no hay gastos en recursos software.

### P.4 Recursos humanos

Para realizar este proyecto se han invertido alrededor de 300 horas en diseño, desarrollo y elaboración de documentación. Según el COIT, para calcular el valor del trabajo empleado se ha de calcular según la ecuación P.2:

$$H = C_t * 74.88 * H_n + C_t * 96.72 * H_e \quad (\text{Ec. P.2})$$

Teniendo:

- $H$ : Honorarios totales recibidos por el proyecto.
- $C_t$ : factor de corrección dependiente del número de horas trabajadas.
- $H_n$ : Horas trabajas en horario laboral.
- $H_e$ : Horas trabajas fuera de horario laboral (en este proyecto no han existido así que su valor es 0)

Teniendo también en cuenta los criterios del COIT, el factor de corrección es el siguiente según las horas trabajadas (Tabla 12):

HORAS EMPLEADAS	FACTOR DE CORRECCION $C_t$
$X < 36$	1
$36 < x < 72$	0.90
$72 < x < 108$	0.80
$108 < x < 144$	0.70
$144 < x < 180$	0.65
$180 < x < 360$	0.60
$360 < x < 540$	0.55

Tabla 12 Valores de la variable  $C_t$  en función de las horas trabajadas

Según esta tabla, al estar este proyecto finalizado en 300 horas, se deberá de escoger el valor del factor de corrección dado para el intervalo de las 180-360 horas:  $C_t = 0.60$ . Dado esto, la fórmula anterior quedaría:

$$H = 0.6 * 74.88 * 300 + 0.6 * 96.72 * 0 = 13478.40 \text{ €}$$

Los honorarios derivados del tiempo dedicado al proyecto libres de impuestos ascienden a unos **TRECE MIL CUATROCIENTOS SETENTA Y OCHO EUROS CON CUARENTA CÉNTIMOS (13478.40 €)**.

También se debe de tener en cuenta los honorarios derivados de la redacción del TFG. Su cálculo se rige según la ecuación P.3:

$$R = 0.07 * P * C_n \quad (\text{Ec. P.3})$$

Teniendo:

- $P$ : presupuesto del trabajo.
- $C_n$ : Coeficiente de ponderación dependiente del presupuesto del trabajo.

El presupuesto del trabajo es la suma de lo anteriormente calculado por ahora (Tabla 13):

Descripción	Costes
Recursos hardware	250€
Trabajo tarifado por tiempo empleado	13478.40 €
Total	13728.40 €

Tabla 13 Presupuesto de los costes materiales y tarifado por tiempo empleado

Se obtiene que el presupuesto total hasta el momento es de 13728.40 €. De aquí a terminar el presupuesto no van a existir grandes desembolsos, y dado que el coeficiente de ponderación definido por el COIT para presupuestos menores de 30050 € es de 1, tenemos que:

$$R = 0.07 * 13728.40 * 1 = 960.99\text{€}$$

Por tanto, el coste libre de impuestos derivado de la redacción del TFG es de **NOVECIENTOS SESENTA EUROS CON NOVENTA Y NUEVE CÉNTIMOS (960.99€)**

## P.5 Gastos en material adicional

Aparte de los gastos derivados del material hardware (y software si hubiera), todos los trabajos también han de usar otros materiales o recursos como son folios, tinta para impresiones o gastos en el transporte:

Descripción	Costes
Folios	10€
Tóner de impresora	50€
Transporte	90€
Total	150€

Tabla 14 Costes del material adicional

## P.6 Derechos del visado del COIT

Los gastos del visado del COIT según la ecuación P.4 son:

$$V = 0.006 * P * C_v \quad (\text{Ec. P.4})$$

Teniendo

- $P$ : Presupuesto del trabajo
- $C_v$ : Coeficiente reductor en función del presupuesto del trabajo.

$P$  es el presupuesto hasta este momento, dado por la suma de los costes de ejecución material, redacción y material fungible:

$$P = 13728.40 + 960.99€ + 150 = 14839.39 €$$

Como el coeficiente de ponderación definido por el COIT para presupuestos menores de 30050 € es de 1, tenemos que:

$$V = 0.006 * 14839.39 * 1 = 89.04 €$$

Por tanto, el coste de los derechos de visado del COIT para este Trabajo ascienden a **OCHENTA Y NUEVE EUROS CON CUATRO CÉNTIMOS (89.04 €)**.

Como último gasto fijo, tenemos fijados los gastos de tramitación y envío en 6.01 €

## P.7 Aplicación de impuestos

El coste total de este TFG asciende a 16295,43 € libres de impuestos. Se le ha de sumar el 7% de IGIC, con lo que el coste definitivo de este TFG asciende a (Tabla 15):

<b>COSTES TOTALES DEL TRABAJO FIN DE GRADO</b>	
<b>Descripción</b>	<b>Coste total (€)</b>
Recursos materiales	250
Coste de ingeniería	14839.39
Coste de redacción	960.99
Material adicional	150
Derechos de visado	89.04
Tramitación y envío	6.01
Aplicación de impuestos (7% IGIC)	1140.68
<b>Total</b>	<b>17436.11</b>

Tabla 15 Presupuesto total del Trabajo Fin de Grado

