



TRABAJO FIN DE GRADO

**ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA
INFORMÁTICA**

GRADO EN INGNIERÍA DEL SOFTWARE

CURSO 2013/2014

DESARROLLO DE UNA APLICACIÓN WEB PARA ESALUD

Autor: Denís Vaillo Sánchez

Tutor: César Cáceres Taladriz

Resumen

Las necesidades de atención y cuidado de los pacientes fuera de las instalaciones de los hospitales y centros médicos ha impulsado el desarrollo de tecnologías para cubrirlas. Por ello, en este Trabajo Fin de Grado se ha desarrollado una aplicación web para la gestión y telecuidado de pacientes con VIH/Sida; la cual ayuda a tener un mayor control y mejor tratamiento de los pacientes por parte del personal médico en tiempo real y sin limitaciones geográficas.

Este proyecto, forma parte del sistema de telemedicina que está implantado en el Servicio de Enfermedades Infecciosas del Hospital Clínic de Barcelona. Actualmente, este sistema da servicio a más de 70 profesionales sanitarios que realizan una media de 150 consultas al día y alrededor de unos 8000 pacientes.

Índice

1.	Introducción.....	9
1.1	Enfermedad del VIH/Sida.....	9
1.1.1	¿Qué es?	9
1.1.2	Tratamiento clínico	10
1.1.3	Epidemiología.....	11
1.2	Presentación del problema	12
1.3	Estado del arte.....	13
1.4	Objetivos.....	14
2.	Análisis	19
2.1	Requisitos funcionales	19
2.1.1	Gestión de usuarios	19
2.1.2	Control de acceso.....	22
2.1.3	Perfil del usuario	22
2.1.4	Gestión de roles	23
2.1.5	Historiales Médicos	24
2.1.6	Gestión de Citas	26
2.1.7	Comunicación	27
2.1.8	Consultas.....	28
2.2	Requisitos no funcionales	29
2.2.1	Escalabilidad.....	29
2.2.2	Facilidad de uso	29
2.2.3	Seguridad	29
2.2.4	Otros.....	29
2.3	Metodologías desarrollo software.....	30
2.3.1	Desarrollo iterativo e incremental.....	30
2.3.2	Ventajas	31
2.3.3	Inconvenientes	31
2.4	Tecnologías empleadas	33
2.4.1	Lado del servidor	33
2.4.2	Lado del cliente.....	36

2.4.3	Preparación del entorno	40
2.5	Modelado de datos conceptual.....	42
2.6	Diagramas de casos de uso	44
3.	Diseño.....	47
3.1	Modelado lógico de datos	47
3.2	Arquitectura	48
3.3	MVC	49
3.3.1	Modelo	49
3.3.2	Vista	49
3.3.3	Controlador	49
3.4	Diagrama de clases	50
3.5	Diseño de la interfaz grafica	52
3.6	Usabilidad	53
3.6.1	Facilidad de aprendizaje	53
3.6.2	Flexibilidad	54
3.6.3	Robustez.....	54
3.7	Accesibilidad	55
3.7.1	Principio 1: Perceptible.....	55
3.7.2	Principio 2: Operable	56
3.7.3	Principio 3: Comprensible	56
3.6.4	Principio 4: Robusto	56
4.	Implementación	57
4.1	Código fuente.....	57
4.1.1	Estructura	57
4.1.2	Desarrollo de las principales funcionalidades	62
4.2	Control de versiones	70
4.3	Interfaz de usuario	71
4.4	Evaluación de la usabilidad	73
4.5	Evaluación de la accesibilidad	76
5.	Conclusiones y futuros trabajos.....	79
5.1	Conclusiones	79
5.2	Trabajos futuros	80

5.2.1	Mejoras en funcionalidades existentes	80
5.2.2	Nuevas funcionalidades propuestas	80
6	Anexos	83
6.1	Manual de usuario.....	83
6.1.1	Inicio	83
6.1.2	Navegación	84
6.1.3	Página principal	84
6.1.4	Gestión de roles	87
6.1.5	Gestión de usuarios	88
6.1.6	Gestión del personal.....	90
6.1.7	Gestión de pacientes	91
6.1.8	Ajustes	93
6.1.9	Gestión de citas	93
6.1.10	Mensajería.....	95
6.1.11	Historiales médicos.....	96
6.1.12	Sala de espera.....	100
7	Bibliografía.....	105

1. Introducción

En este capítulo, se presenta una breve explicación de la enfermedad del VIH/Sida, presentación del problema, estado del arte y los objetivos.

1.1 Enfermedad del VIH/Sida

VIH es la sigla del virus de inmunodeficiencia humana. Es un virus que mata o daña las células del sistema inmunológico del organismo. SIDA es la sigla del síndrome de inmunodeficiencia adquirida, que es el estadio más avanzado de esta infección. [1]

1.1.1 ¿Qué es?

El VIH infecta a las células del sistema inmunitario, alterando o anulando su función. La infección produce un deterioro progresivo del sistema inmunitario con la consiguiente "inmunodeficiencia".

Se considera que el sistema inmunitario es deficiente cuando ya no puede cumplir su función de lucha contra las infecciones y enfermedades. El síndrome de inmunodeficiencia adquirida (SIDA) es el término que se aplica a los estadios más avanzados de la infección por VIH y se define por la presencia de alguna de las más de veinte "infecciones oportunistas" y neoplasias relacionadas con el VIH, que se desarrollan aprovechando la debilidad del sistema inmunitario.

El VIH puede transmitirse mediante relaciones sexuales vaginales, anales u orales con una persona infectada, transfusiones de sangre contaminada o el uso compartido de agujas, jeringuillas u otros instrumentos punzantes. Asimismo, puede transmitirse de madre a hijo durante el embarazo, el parto y la lactancia.

La enfermedad puede destacarse en tres fases: [2]

1. Infección aguda:

En la primera fase, o fase de infección aguda, se da la incubación del VIH. Es el tiempo de exposición al virus hasta que aparecen los primeros signos de la enfermedad. Ese período dura entre 3 y 6 semanas. Y, tras la infección, pasan entre 8 y 12 semanas hasta que el organismo produce anticuerpos contra el VIH. Los primeros síntomas son muy parecidos a los de una gripe, como fiebre y malestar. Por eso, la mayoría de los casos pasan desapercibidos.

2. Etapa asintomática:

La fase siguiente está marcada por una fuerte interacción entre las células de defensa y las constantes y rápidas mutaciones del virus. Sin embargo, el organismo no se debilita lo suficiente como para permitir nuevas enfermedades, pues los virus maduran y mueren de manera equilibrada. Ese período, que puede durar muchos años, se denomina período asintomático.

Con el ataque frecuente, las células de defensa empiezan a funcionar con menos eficiencia, hasta que son destruidas. El organismo está cada vez más debilitado y

vulnerable a infecciones comunes. La fase sintomática inicial se caracteriza por una reducción considerable de los linfocitos CD4 (glóbulos blancos del sistema inmunitario), que llegan a cifras inferiores a 200 unidades por mm³ de sangre. En los adultos sanos, este valor varía entre 800 y 1200 unidades. Los síntomas más comunes son fiebre, diarrea, sudores nocturnos y adelgazamiento.

3. Fase final o de crisis

La baja inmunidad permite la aparición de enfermedades oportunistas, así denominadas porque se aprovechan de la debilidad del organismo. De esa manera, se alcanza la fase más avanzada de la enfermedad: el sida. Quien llega a esa fase, por no conocer o no seguir el tratamiento indicado por los médicos, puede sufrir hepatitis virales, tuberculosis, neumonía, toxoplasmosis y algunos tipos de cáncer.

1.1.2 Tratamiento clínico

En la actualidad no existe ninguna vacuna contra el virus del SIDA. Los avances cada vez son mayores, por una parte tratan de controlar la infección y por otra intentan hacer más fácil la vida a los pacientes.

Por ello el tratamiento más eficaz en este momento es la Terapia antirretroviral (ARV). Se entiende por ARV la aplicación de una serie de medicamentos que tienen por objetivo impedir la replicación del virus y restaurar el sistema inmunológico del paciente. Actualmente, el tratamiento consiste en la combinación de varios fármacos. [3]

Los medicamentos antirretrovirales son de gran eficacia, retrasan la progresión de la infección por VIH y aumentan la supervivencia de los pacientes. La terapia ARV actualmente disponible no elimina el virus del cuerpo, pero prolonga la vida y la salud mediante la reducción de los efectos adversos del VIH en el sistema inmunitario.

La terapia es individualizada y debe ser prescrita, revisada y modificada por los especialistas que realizan el seguimiento de la persona infectada. Cada persona afectada debe decidir cuándo iniciar un tratamiento con fármacos tras ser informado correctamente por su médico que, habrá evaluado previamente el estado de la infección por VIH.

El tratamiento ayuda a que el virus no se multiplique, reduciendo su presencia en el organismo. Disminuye por tanto la carga viral y esto reduce la capacidad de transmitir el virus (aunque cualquier persona VIH positiva puede transmitirlo, incluso si su carga viral fuera indetectable). En relación a esta terapia es muy importante una buena adherencia al tratamiento (toma de las dosis indicadas y cumplimiento de horarios) ya que, de no ser así, disminuye la concentración de medicamento en sangre aumentando el riesgo de que el virus se haga resistente a los antirretrovirales usados y de que éstos pierdan eficacia. Es importante asimismo seguir el régimen de comidas indicado.

La terapia, por tanto, exige constancia, pero ha mejorado tanto la esperanza de vida que se dice que el VIH y el SIDA es hoy día una enfermedad crónica.

1.1.3 Epidemiología

En 2012, se calculaba que había 35,3 millones de adultos y niños contagiados con el VIH en el mundo. La mayoría de estas personas está en África al sur del Sahara y en algunas partes de Asia. Como se puede ver en la línea correspondiente al “mundo”, si bien las personas continúan infectándose, la tasa de nuevas infecciones está disminuyendo. [4]

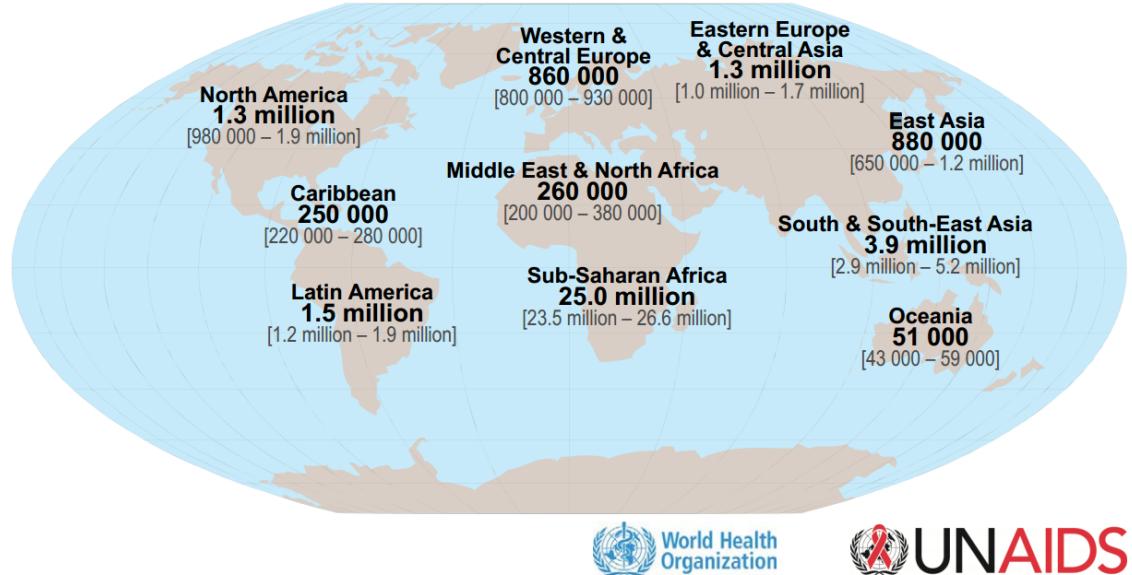


Ilustración 1. Estimación de adultos y niños viviendo con VIH

En todo el mundo, 2,3 millones de adultos y niños se infectaron por primera vez con el VIH en 2012, lo cual representa una baja de 38% en comparación con el máximo de 3,7 millones en 1997. Es un verdadero avance: una reducción continua en la tasa de nuevas infecciones conducirá a una disminución en el número de personas que viven con el VIH y mueren a raíz del virus.

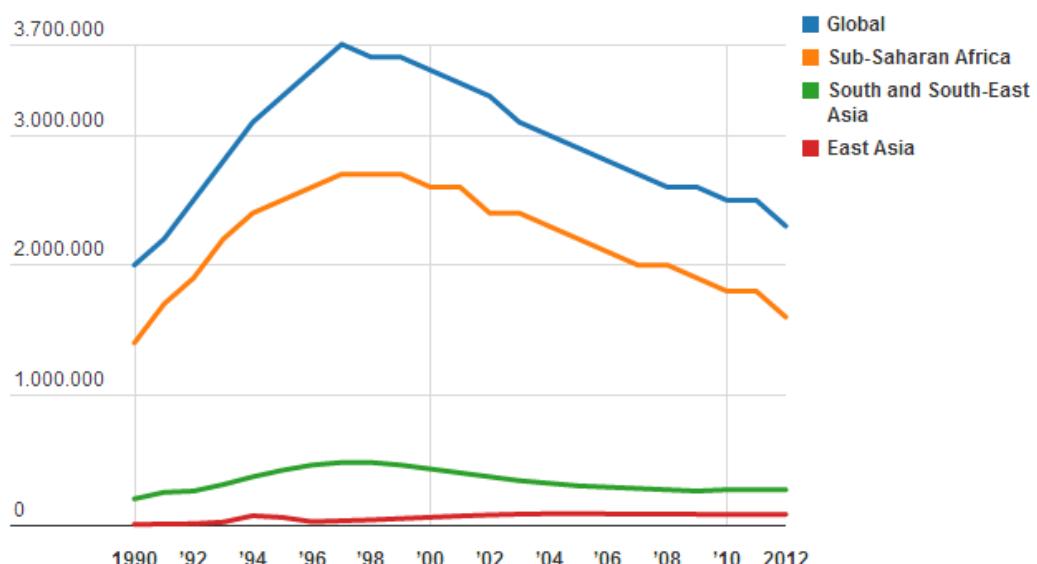


Ilustración 2. Nuevos casos detectados en adultos y niños

En 2012, 1,6 millones de personas murieron debido a causas relacionadas con el SIDA en todo el mundo, cifra que significa una reducción de 30% desde el máximo de 2,3 millones en 2005. Según la Organización Mundial de la Salud (OMS) .Hay dos razones principales para esta disminución: por un lado, la mayor disponibilidad de tratamiento ARV y la atención de los infectados por el VIH y, por otro, la menor cantidad de infectados por primera vez con el virus desde el pico en 1997.[5]

Hay una gran diferencia entre la tendencia mundial de esperanza de vida con las tendencias en los países de África al sur del Sahara más gravemente afectados por el VIH. En 1987, la esperanza de vida en Zimbabwe era de casi 62 años. En el momento culmen de la epidemia en 2003, había caído un 30% llegando a 43 años, y en la actualidad con una declinación en la prevalencia del VIH este indicador está comenzando a subir de nuevo.

1.2 Presentación del problema

Con el avance de la tecnología, la telemedicina está teniendo una mejor acogida en la sociedad. La telemedicina incluye tanto diagnóstico y tratamiento, como también la educación médica. Es un recurso tecnológico que posibilita la optimización de los servicios de atención en salud, ahorrando tiempo y dinero y facilitando el acceso a zonas distantes para tener atención de especialistas. [6]

El problema de los pacientes con VIH es la necesidad de tener un estricto control de la enfermedad y esto conlleva una gran cantidad de visitas al especialista, pero no solo esto, la red es una de las principales fuentes de información para los pacientes que anónimamente se intentan informar de su enfermedad, esto provoca que el tratamiento no sea el adecuado y en gran parte de las ocasiones esta información encontrada es errónea, ya que detrás de ella no existe ningún experto en la materia.

En la lucha contra la calidad de la información médica en la web diversas entidades han elaborado diferentes sellos de calidad, códigos de conducta, acreditaciones o recomendaciones, con el objeto de que los sitios web con contenidos en salud, puedan adoptarlos y así informar y asegurar a las personas usuarias, que en sus sitios web se respetan unos mínimos de calidad. [7]

Para reducir el impacto de los posibles riesgos de la utilización de información de escasa calidad, en los últimos años han sido desarrollados códigos de ética y de conducta que en ocasiones se acompañan de sellos o marcas de confianza, que certifican la validez de la información y la de sus responsables. Estos sellos o marcas tienen por objeto ayudar al usuario, ya sea profesional o paciente, a la hora de seleccionar la información a consultar.

Para todos estos sellos el proceso de acreditación del sitio web es similar: el cliente solicita voluntariamente el sello, una comisión evalúa que la página y los contenidos estén de acuerdo con sus códigos de conducta y, después de que la página subsane las deficiencias que se pudieran haber encontrado, se otorga el sello de calidad, que aparecerá en la página solicitante, y éste formará parte de la base de datos de quién acredita.

Entre los sellos de calidad y códigos de conducta se pueden destacar:

1. Recomendaciones de la Unión Europea de 2002.
2. HONcode de la HON (Health on the Net Foundation “Fundación Salud en la Red”) que es el sello de calidad más extendido actualmente a escala internacional.
3. Código de conducta del programa de certificación de webs médicas, Web Médica Acreditada (WMA) del Colegio Oficial de Médicos de Barcelona, que presenta el aval de un colegio profesional y es el que actualmente tiene mayor aceptación según una encuesta realizada por la HON.
4. Sello y Código ético del Proyecto Webs Médicas de Calidad en lengua española.
5. MedCIRCLE es un proyecto financiado por la Unión Europea, iniciado en el 2002 y dirigido por un consorcio de tres portales europeos sobre salud, España, Alemania y Francia.
6. Código Ético y Sello de la Agencia de Calidad Sanitaria de Andalucía incluido dentro de su proceso de Acreditación de Páginas Web Sanitarias.

Si este medio tuviese una gran acogida proporcionaría esa ayuda a los pacientes en la búsqueda de información sobre el tratamiento contra su enfermedad, apoyándoles en todo momento personal sanitario cualificado en la materia y llevando el control total de su enfermedad.

1.3 Estado del arte

En el caso del VIH/SIDA se trata de una enfermedad muy compleja y su tratamiento necesita un apoyo continuo por parte del personal médico. Por ello han surgido algunos proyectos tecnológicos enfocados exclusivamente en esta enfermedad, cabe mencionar principalmente estos tres:

- Proyecto de gestión de pacientes con VIH a domicilio utilizando la red de cable:

Este proyecto fue desarrollado por el Centro de Ciencias de la Salud de la Universidad del Estado de Louisiana, en Estados Unidos. Se basaron en un producto comercial llamado HealthiumTM, desarrollado para recordar la toma de la medicación a personas de edad avanzada a través de la televisión. El sistema se aplicó a 11 pacientes con VIH durante 4 meses. También disponía de la posibilidad de realizar llamadas de videoconferencia, pero este servicio no fue muy bien valorado por los pacientes. [8]

- Proyecto de televisita de pacientes con VIH:

Desarrollado por VA New Jersey Health Care System. El objetivo del proyecto era evitar traslados de pacientes que vivían lejos de su centro médico de referencia, montando un enlace de videoconferencia entre un centro más cercano y el centro de referencia. Las pruebas se realizaron con 10 pacientes que vivían a más de 75Km del centro médico de referencia y a menos de 30 Km del centro intermedio desde donde se realizaba la videoconferencia. [9]

- Hospital VIHrtual:

El 14 de marzo de 2011 el Hospital Clínic de Barcelona presentó el Hospital VIHrtual, un programa de telemedicina que permite atender por control remoto a pacientes infectados por VIH. Mediante una conexión en directo a través de una cámara web, esta herramienta permite pasar consulta virtual a los pacientes, que de esta manera pueden ser atendidos desde su casa o de donde dispongan de un ordenador con conexión a Internet. Este hospital virtual permite un control integral del paciente en los aspectos médico, farmacéutico, psicológico y de calidad de vida. Aunque los resultados son igual de satisfactorios que los obtenidos en una visita presencial, desde el propio Clínic apuntan que no sustituye el sistema clásico presencial, sino que lo complementa y mejora. [10]

Este último sistema es el punto de referencia para este trabajo.

1.4 Objetivos

La aplicación actual (Hospital VIHrtual) no se adecua a las nuevas tecnologías y limita a los pacientes al acceso a la misma únicamente desde un ordenador de sobremesa, lo cual es una gran barrera para tener un control total del tratamiento.

En la actualidad, el uso de Smartphones y Tablets para la consulta de webs está en constante crecimiento, por ello la creación de una alternativa para ellos es obligatoria.[11]



Ilustración 3. Evolución del tráfico web por dispositivo

El tráfico web está dominado por los ordenadores tradicionales, pero está basado en su uso en el ámbito laboral. Fuera de horarios laborales el dominio es total por parte de dispositivos móviles.

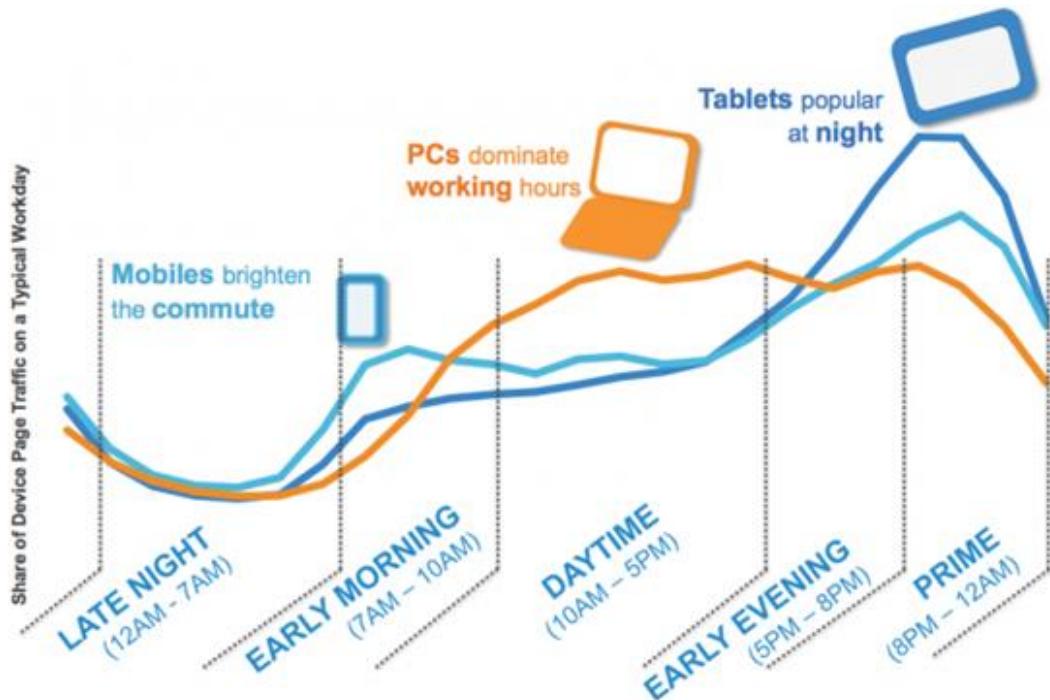


Ilustración 4. Trafico web por horarios según dispositivos en días laborables

Tras estudiar los valores anteriores, se ha llegado a la conclusión de la necesidad de crear una aplicación que sea compatible con todos los dispositivos. Las posibles opciones que se podrían desarrollar son:

- La creación de aplicaciones nativas para cada sistema operativo móvil:

Esta solución implica el desarrollo de aplicaciones independientes, lo que conlleva gran cantidad de trabajo y la necesidad de aprendizaje de los lenguajes que cada uno de ellos necesita.

- La creación de un sistema web móvil homólogo al actual:

Esta solución en parte podría ser buena ya que se mantendría la aplicación actual y sería crear su homóloga en un formato móvil, pero a la larga conllevaría al mantenimiento y evolución de dos visualizaciones independientes. Y este formato móvil comúnmente presenta una visualización mínima de todas las funcionalidades. En la actualidad con la gran variedad de tamaños de pantalla no se podría ajustar a todas ellas, teniendo únicamente la normal y la móvil, pero para tablets no tendría el ajuste necesario para una buena usabilidad.

- La creación de una plataforma web *responsive* compatible con todos los dispositivos y navegadores. Esto solucionaría el problema de mantenimiento y sería capaz de adaptarse a cualquier dispositivo sin necesidad de realizar tareas de adaptación. El mantenimiento y desarrollo de cualquier mejora sería más simple al únicamente tener que realizar una visualización común a todos. En caso de que se necesite crear aplicaciones nativas para sistemas operativos móviles es posible

crear webapp que son aplicaciones basadas en un navegador web que apuntan a la dirección deseada.

Para este proyecto la alternativa elegida es la creación de una plataforma desde cero para poder realizar un diseño *responsive* que sea accesible desde cualquier dispositivo adaptándose correctamente a su pantalla.

Tras el estudio de las posibles alternativas se obtienen los objetivos generales que deberá cumplir la aplicación:

1. Desarrollo de una web *responsive* adaptando los servicios disponibles en el proyecto Hospital VIHrtual a las últimas tecnologías web.
2. Este sistema de información clínica incluirá la gestión completa de pacientes con VIH tanto de forma online como presencial.

Dados estos objetivos generales se obtienen a grandes rasgos todas las funcionalidades necesarias para el sistema, que más adelante serán identificadas como requisitos funcionales.

- Gestión de usuarios:

El cual incluye una gestión tanto de personal del propio hospital, así como de los pacientes del mismo. Debe poder darse de alta pacientes y personal de forma sencilla y rápida. Los usuarios serán diferenciados dentro de la aplicación por roles identificativos (médicos, pacientes, personal, etc...)

- Control de acceso:

La plataforma es cerrada y su acceso debe ser con un usuario dado de alta. Dentro de este control se registrarán todas las acciones que el usuario realice sobre la misma.

- Perfil del usuario:

Cada usuario tendrá su propio perfil donde podrá configurar elementos básicos de su cuenta.

- Gestión de Roles:

La plataforma debe ser totalmente configurable adaptándose a las necesidades del hospital, pudiéndose gestionar qué acciones tiene disponible cada tipo de usuario.

- Historiales Médicos:

La gestión de historiales médicos de los pacientes es uno de los puntos clave del sistema, se debe tener acceso a la información clínica del paciente en todo momento, para tener un control del estado del paciente, tratamientos, protocolos, analíticas, etc...

- Gestión de citas:

Deben poderse gestionar las citas de los pacientes con sus médicos, pudiendo visualizar y solicitar citas tanto por parte del personal como del propio paciente (dependiendo de la configuración de los roles). En todo momento los pacientes y el personal deben tener una visualización clara de su agenda de citas.

- Comunicación:

Para favorecer el flujo de información y que los pacientes estén informados en todo momento existirá un canal de mensajería uno a uno para cualquier tipo de consulta. La posibilidad de comunicarse con otro usuario viene impuesta por el tipo de usuario.

- Consultas:

Cada profesional sanitario tendrá su propia sala de espera donde poder atender a los pacientes, estas atenciones pueden hacerse online a través de un servicio de videollamada, así como presenciales, donde el personal sanitario podrá ver y gestionar a su paciente.

Todos estos servicios abordan la solución a las situaciones que hacen que los pacientes no consigan llevar su tratamiento de la manera correcta, dando solución a los problemas identificados desde el inicio:

- Visitas al centro médico:

Con la posibilidad de realizar las consultas de forma online evitamos que los pacientes deban acudir hasta un lugar físico que conlleva una gran pérdida de tiempo y en muchos casos a la imposibilidad de poder asistir a las mismas. Todas las dudas que el paciente pueda tener de su enfermedad tendrán una rápida solución pudiéndose comunicar con especialistas en todo momento, evitando que obtengan información incorrecta de medios exteriores.

Los resultados de analíticas y la gestión de su medicación pueden ser conocidos al momento.

- Atención anónima y personalizada:

Los pacientes suelen ser reticentes a que su enfermedad sea conocida por la sociedad, por ello de esta forma se puede mantener un anonimato favoreciendo a que tengan un tratamiento correcto.

Aparte de todos los objetivos funcionales de la web existen requisitos funcionales que deberían estar presentes en la aplicación:

- Ubicuidad: Tener acceso desde cualquier lugar a la información.
- Simplicidad: Todas las funcionalidades deben ser lo más sencillas de usar posible.
- Seguridad: Acceso seguro a los datos limitando la posibilidad de acceder a datos privados, y registrando cada acceso a los mismos.

- **Accesibilidad:** Debe poder ser utilizada por el mayor número de personas, independientemente de sus capacidades.
- **Usabilidad:** La interfaz debe ser fácil de usar, sin necesidad de tener que leer documentación.
- **Fiabilidad:** El sistema debe tener un buen funcionamiento bajo cualquier condición.
- **Mantenibilidad:** Introducir nuevos cambios deber ser sencillo , así como la posibilidad de evolucionar y adaptarse a futuros cambios

2. Análisis

En este capítulo, se llevará a cabo la primera etapa del ciclo de vida del software, en la que se determinan las necesidades del cliente, así como los servicios que debe proporcionar la aplicación.

A continuación, se describirán los requisitos funcionales y no funcionales, las metodologías y tecnologías empleadas, el modelado de datos y diagramas de casos de uso.

2.1 Requisitos funcionales

Los requisitos funcionales son los encargados de establecer los comportamientos del sistema. A continuación se describirán por cada funcionalidad todos los requisitos necesarios:

2.1.1 Gestión de usuarios

Como toda aplicación es necesario llevar a cabo la administración de los usuarios. Inicialmente esta tarea será llevada a cabo por el administrador del sistema, pero más adelante podrán realizarla usuarios con permisos.

La plataforma distingue entre 4 tipos de usuarios:

- Pacientes:

Serán las personas que sufren la enfermedad del VIH/SIDA y gestionan su tratamiento en la plataforma.

- Personal Sanitario:

Serán los encargados de llevar a cabo los tratamientos de los pacientes.

- Personal Administrativo:

Orientado a usuarios encargados del buen funcionamiento del hospital, encargándose de dar citas, gestionar el personal, etc..., dependiendo de los permisos que tenga cada usuario.

- Administradores:

Este tipo de usuarios serán los menos comunes, se encargan de la gestión de la aplicación. De inicio tendrán permisos para realizar cualquier acción sobre la aplicación. Y serán los encargados de la configuración y creación de los usuarios iniciales.

La gestión de usuarios estará dividida en tres pequeñas aplicaciones para intentar facilitar y dar la posibilidad de dividir la labor más importante del sistema. Tendremos una aplicación principal donde se administran todos los usuarios de la plataforma, y dos más específicas para controlar el personal sanitario y los pacientes.

2.1.1.1 Usuarios

La gestión de todos los usuarios de la aplicación tendrá disponible una serie de funcionalidades que permitirá administrar toda la información relativa al usuario sin importar su rol.

Los requisitos identificados para la correcta gestión de usuarios son:

Usuarios – 01	Alta
Cualquier usuario, con los permisos necesarios, podrá dar de alta a nuevos usuarios. La información a cumplimentar dependerá del tipo de usuario a crear	
Usuarios – 02	Baja
Cualquier usuario, con los permisos necesarios, podrá dar de baja a cualquier otro usuario de la aplicación. La información nunca será borrada de la base de datos.	
Usuarios – 03	Modificación
Cualquier usuario, con los permisos necesarios, podrá modificar a posteriori toda la información relativa al usuario, incluso cambiar el rol.	
Usuarios – 04	Reseteo contraseña
Cualquier usuario, con los permisos necesarios, podrá resetear la contraseña a otro usuario, el cual recibirá un aviso de dicha modificación.	
Usuarios – 05	Consulta
Cualquier usuario, con los permisos necesarios, podrá consultar el listado de usuarios y toda la información.	
Usuarios – 06	Búsqueda
El listado de usuarios tendrá la posibilidad de realizar una búsqueda por nombre o identificadores, para facilitar su uso.	

2.1.1.2 Personal Sanitario

Para facilitar la gestión del personal se ha separado esta funcionalidad donde aparecerá únicamente información relevante para gestionar el personal sanitario del hospital. Los usuarios mostrados en esta funcionalidad son el personal sanitario y administrativo.

Los requisitos identificados para ello son:

Personal Sanitario – 01	Consulta
Cualquier usuario, con los permisos necesarios, podrá consultar el listado de usuarios y toda la información relevante como empleado del hospital.	
Personal Sanitario – 02	Consultar pacientes del profesional sanitario
Cualquier usuario, con los permisos necesarios, podrá consultar los pacientes que tienen asignados el profesional sanitario.	
Personal Sanitario – 03	Asignar pacientes al profesional sanitario
Cualquier usuario, con los permisos necesarios, podrá asignar pacientes al profesional sanitario.	
Personal Sanitario – 04	Quitar pacientes al profesional sanitario
Cualquier usuario, con los permisos necesarios, podrá quitar pacientes que tenga asignado el profesional sanitario.	
Personal Sanitario – 05	Editar
Cualquier usuario, con los permisos necesarios, podrá editar la información relevante del usuario como trabajador.	
Personal Sanitario – 06	Búsqueda
El listado de usuarios tendrá la posibilidad de filtrarse o realizar búsquedas por nombre o identificadores.	

2.1.1.3 Pacientes

Para facilitar la gestión de los pacientes se ha separado esta funcionalidad donde aparecerá únicamente información relevante como pacientes del hospital. Los usuarios mostrados en esta funcionalidad son únicamente los pacientes.

Los requisitos identificados son:

Pacientes – 01	Consulta
Cualquier usuario, con los permisos necesarios, podrá consultar el listado de pacientes y toda la información relevante como paciente.	

Pacientes – 02	Consultar profesionales sanitarios del paciente
Cualquier usuario, con los permisos necesarios, podrá consultar los profesionales sanitarios que tiene asignados cada paciente	
Pacientes – 03	Asignar profesionales sanitarios al paciente
Cualquier usuario, con los permisos necesarios, podrá asignar profesionales sanitarios a los pacientes	
Pacientes – 04	Quitar profesionales sanitarios al paciente
Cualquier usuario, con los permisos necesarios, podrá quitar profesionales sanitarios que tenga asignado el paciente.	
Pacientes – 05	Editar
Cualquier usuario, con los permisos necesarios, podrá editar la información relevante del usuario como paciente.	
Pacientes – 06	Búsqueda
El listado de pacientes tendrá la posibilidad de filtrarse o realizar búsquedas por nombre o identificadores.	

2.1.2 Control de acceso

La aplicación tiene un acceso restringido únicamente para usuarios registrados, y todas sus acciones deben quedar constancia.

Los requisitos identificados son:

Acceso – 01	Identificación
El acceso a la plataforma será realizado mediante el DNI completo y la contraseña asociada.	
Acceso – 02	Control
Toda acción que realice el usuario sobre la aplicación será registrada en la base de datos	

2.1.3 Perfil del usuario

Como toda aplicación cada usuario podrá modificar ciertos valores de su perfil personal, como la contraseña o el email.

Los requisitos identificados son:

Perfil – 01	Cambio contraseña
Cualquier usuario podrá solicitar el cambio de contraseña, para ello deberá confirmar cuál es su contraseña actual y que contraseña nueva quiere introducir	
Perfil – 02	Cambio de email
Cualquier usuario podrá solicitar el cambio de email, para ello deberá introducir su nuevo email y confirmar que tiene acceso a él accediendo al enlace de confirmación que recibirá	

2.1.4 Gestión de roles

Como se ha visto anteriormente la plataforma se basa en cuatro tipos de usuarios, estos se pueden gestionar dando o quitando permisos que tienen para acceder o modificar información de cada herramienta. Esta modificación de permisos permitirá la creación de perfiles de usuarios de cada rol. Inicialmente cada rol tiene un perfil por defecto.

Los requisitos identificados son:

Gestión Roles – 01	Consulta
Cualquier usuario, con los permisos necesarios, podrá consultar los roles y perfiles de la aplicación, así como los permisos asociados a cada uno de ellos.	
Gestión Roles – 02	Crear perfil
Cualquier usuario, con los permisos necesarios, podrá crear nuevos perfiles de permisos para cualquier rol.	
Gestión Roles – 03	Editar perfil
Cualquier usuario, con los permisos necesarios, podrá modificar los perfiles de los roles, incluyendo nombre y permisos.	
Gestión Roles – 04	Eliminar perfil
Cualquier usuario, con los permisos necesarios, podrá eliminar los perfiles de roles que se hayan creado, menos los por defecto.	
Gestión Roles – 05	Asignar perfil
Cualquier usuario, con los permisos necesarios, podrá cambiar el perfil o rol de los usuarios de la aplicación.	

Gestión Roles – 06	Cambiar permisos usuario
Cualquier usuario, con los permisos necesarios, podrá cambiar los permisos de cada usuario de forma individual.	
Gestión Roles – 08	Consultar permisos usuario
Cualquier usuario, con los permisos necesarios, podrá consultar los permisos de usuarios de la aplicación	
Gestión Roles – 09	Búsqueda usuarios
Cualquier usuario, con los permisos necesarios, podrá realizar búsquedas sobre el listado de usuarios de la aplicación.	

Uno de los permisos existentes es el habilitar el acceso o no a la plataforma, esto no podrá ser aplicado a los administradores de la plataforma, que siempre podrán acceder a la misma.

Dado este gran nivel de personalización de perfiles puede darse el caso de que cualquier usuario puede realizar todas las acciones existentes en la plataforma, estará en manos de quien configure la plataforma el buen uso de esta flexibilidad tan grande.

2.1.5 Historiales Médicos

Los pacientes del hospital llevaran asociado un historial médico único en el cual se recopilará toda la información sobre su tratamiento frente al VIH/SIDA.

Los requisitos identificados son:

Historial Médico – 01	Consulta historiales
Cualquier usuario, con los permisos necesarios, podrá consultar la lista de historiales a los que tiene acceso.	
Historial Médico – 02	Búsqueda historiales
Cualquier usuario, con los permisos necesarios, podrá aplicar filtros sobre el listado completo de historiales	
Historial Médico – 03	Consulta historial
Cualquier usuario, con los permisos necesarios, podrá consultar toda la información del historial médico de un paciente, incluyendo información usuario, informes, protocolos, tratamientos y analíticas.	

Historial Médico – 04	Añadir tratamientos
Cualquier usuario, con los permisos necesarios, podrá añadir tratamientos nuevos a los pacientes.	
Historial Médico – 05	Editar tratamientos
Cualquier usuario, con los permisos necesarios, podrá editar los tratamientos del paciente	
Historial Médico – 06	Eliminar tratamientos
Cualquier usuario, con los permisos necesarios, podrá eliminar los tratamientos actuales de los pacientes	
Historial Médico – 07	Añadir protocolo
Cualquier usuario, con los permisos necesarios, podrá añadir protocolos nuevos a los pacientes	
Historial Médico – 08	Editar protocolos
Cualquier usuario, con los permisos necesarios, podrá editar los protocolos del paciente	
Historial Médico – 09	Eliminar protocolos
Cualquier usuario, con los permisos necesarios, podrá eliminar los protocolos del paciente	
Historial Médico – 10	Añadir analíticas
Cualquier usuario, con los permisos necesarios, podrá añadir analíticas nuevas a los pacientes	
Historial Médico – 11	Editar analíticas
Cualquier usuario, con los permisos necesarios, podrá editar las analíticas del paciente	
Historial Médico – 12	Eliminar analíticas
Cualquier usuario, con los permisos necesarios, podrá eliminar las analíticas del paciente	
Historial Médico – 13	Avisos alertas
Cualquier usuario que acceda al historial y tenga avisos nuevos sobre el paciente dirigidos a su rol recibirá una alerta con dichos avisos	

Historial Médico – 14	Consultar alertas
Cualquier usuario, con los permisos necesarios, podrá consultar el histórico de alertas del paciente	
Historial Médico – 15	Añadir alertas
Cualquier usuario, con los permisos necesarios, podrá añadir alertas nuevas a los pacientes. Que irán dirigidas hacia los roles elegidos por el usuario	
Historial Médico – 16	Eliminar alertas
Cualquier usuario, con los permisos necesarios, podrá eliminar las alertas del paciente.	
Historial Médico – 17	Dar cita
Cualquier usuario, con los permisos necesarios, podrá acceder a dar cita al paciente	
Historial Médico – 18	Enviar mensaje
Cualquier usuario, con los permisos necesarios, podrá acceder a enviar un mensaje al paciente	

2.1.6 Gestión de Citas

Todo paciente y personal sanitario deberá tener un calendario de citas. Como en todo hospital estas citas involucran por un lado a un paciente, y por el otro a un profesional sanitario. En esta gestión de citas la participación de los dos roles restantes se limitará a la administración de agendas de los roles involucrados en citas.

Los requisitos identificados son:

Citas – 01	Consultar citas
Cualquier usuario, con los permisos necesarios, podrá consultar sus citas o las de cualquier usuario (dependiendo del rol)	
Citas – 02	Crear citas
Cualquier usuario, con los permisos necesarios, podrá crear citas para el u otros usuarios (dependiendo del rol)	
Citas – 03	Editar citas
Cualquier usuario, con los permisos necesarios, podrá editar sus citas o las de otros usuarios (dependiendo del rol)	

Citas – 04	Eliminar citas
Cualquier usuario, con los permisos necesarios, podrá anular sus citas o las de otros usuarios (dependiendo del rol)	

2.1.7 Comunicación

La comunicación dentro de la plataforma se realizará mediante mensajes directos entre usuarios. Estará limitada según el rol del usuario y los permisos declarados por perfiles.

Las conversaciones podrán ser iniciadas siempre y cuando el usuario creador tenga permiso de comunicarse con el receptor. Por ello inicialmente se establecen las siguientes limitaciones:

- Los pacientes podrán iniciar conversaciones con los profesionales sanitarios que tengan asignados.
- El personal sanitario podrá iniciar conversaciones con cualquier paciente de la plataforma.
- El personal administrativo podrá iniciar conversaciones con cualquier paciente o personal sanitario.
- Los administradores podrán iniciar conversaciones con cualquier usuario de la plataforma.

Los requisitos identificados son:

Comunicación – 01	Consultar conversaciones
Cualquier usuario, con los permisos necesarios, podrá consultar sus conversaciones activas, así como los mensajes nuevos en cada una.	
Comunicación – 02	Consultar contactos
Cualquier usuario, con los permisos necesarios, podrá consultar su listado de contactos a los que pueden conversar	
Comunicación – 03	Iniciar conversación
Cualquier usuario, con los permisos necesarios, podrá iniciar una conversación con cualquiera de su lista de contactos	
Comunicación – 04	Continuar conversación
Cualquier usuario, con los permisos necesarios, podrá consultar las conversaciones que tenga abiertas, con todos los mensajes que tengan.	

Comunicación – 05	Enviar mensaje
Cualquier usuario, con los permisos necesarios, podrá enviar un mensaje a una conversación.	

2.1.8 Consultas

Las consultas médicas podrán ser de dos tipos, online y presenciales. Las dos deben poder ser gestionadas desde la plataforma. En cualquiera de las dos el profesional sanitario tendrá disponible todas las funcionalidades del historial médico.

Los requisitos identificados son:

Consulta – 01	Sala de espera
Cualquier usuario, con los permisos necesarios, podrá entrar a una sala de espera que según si es profesional sanitario o paciente tendrá, o bien el listado de pacientes para ese día o la información de su próxima cita.	
Consulta – 02	Notas
Cualquier paciente, con los permisos necesarios, podrá escribir notas que deberá elegir si son o no visibles para el profesional sanitario en la cita actual.	
Consulta – 03	Iniciar consulta
El profesional sanitario podrá iniciar la consulta con un paciente que tenga citado	
Consulta – 04	Marcar como no comparecida
El profesional sanitario podrá marca una cita como no comparecida siempre que el paciente no se presente.	
Consulta – 04	Rellenar informe
El profesional sanitario podrá llenar el informe de la cita una vez se haya iniciado.	
Consulta – 04	Finalizar videollamada
El profesional sanitario o paciente podrá finalizar la videollamada de la cita	
Consulta – 04	Finalizar consulta
El profesional sanitario podrá dar como finalizada la cita	

2.2 Requisitos no funcionales

Los requisitos no funcionales son los encargados de establecer las características o restricciones del sistema que no tengan que ver con funcionalidades. A continuación se describirán los requisitos necesarios:

2.2.1 Escalabilidad

El sistema debe permitir en el futuro el desarrollo de nuevas funcionalidades, modificarlas o eliminarlas después de su construcción y puesta en marcha inicial.

La base debe permitir incorporar nuevas funcionalidades afectando al código existente de la menor manera posible, los componentes deben ser reutilizables.

2.2.2 Facilidad de uso

El sistema deberá ser fácil de usar y el entrenamiento necesario para aprender todas las funcionalidades el menor posible.

El sistema debe presentar mensajes de error que permitan al usuario identificar el tipo de error y darle la capacidad de solucionarlo.

2.2.3 Seguridad

El acceso a la información de pacientes debe estar permitido únicamente a personal autorizado. Por ello el sistema deberá contar con mecanismos que permitan el registro de toda actividad sobre la aplicación.

El control de acceso se realizará mediante un login y password, donde el login corresponde con el DNI del usuario. Cualquier acceso a vistas no autorizadas deberá ser bloqueado y notificado al usuario.

2.2.4 Otros

La interfaz deberá adaptarse a cualquier tamaño de dispositivo, pudiendo realizarse las mismas acciones en cualquiera de ellos.

El tiempo de respuesta debe ser inmediato, evitando que operaciones de larga duración bloqueen al usuario a realizar otras acciones.

2.3 Metodologías desarrollo software

Una metodología de desarrollo de software se refiere a un marco de trabajo usado para estructurar, planear y controlar el proceso de desarrollo en sistemas de información.[12]

Se ha seguido un desarrollo iterativo e incremental, siguiendo iteraciones cortas donde se iban planificando, analizando y desarrollando cada una de las funcionalidades, así al finalizar cada iteración se tiene una demo de lo requerido al inicio de la iteración que va evolucionando con el paso de las iteraciones.

2.3.1 Desarrollo iterativo e incremental

En el desarrollo iterativo e incremental el proyecto se planifica en diversos bloques temporales llamados iteraciones. Estas se pueden entender como mini proyectos en la que se repite un proceso de trabajo similar para proporcionar un resultado completo sobre el producto final, de manera que el cliente pueda obtener los beneficios de forma incremental.

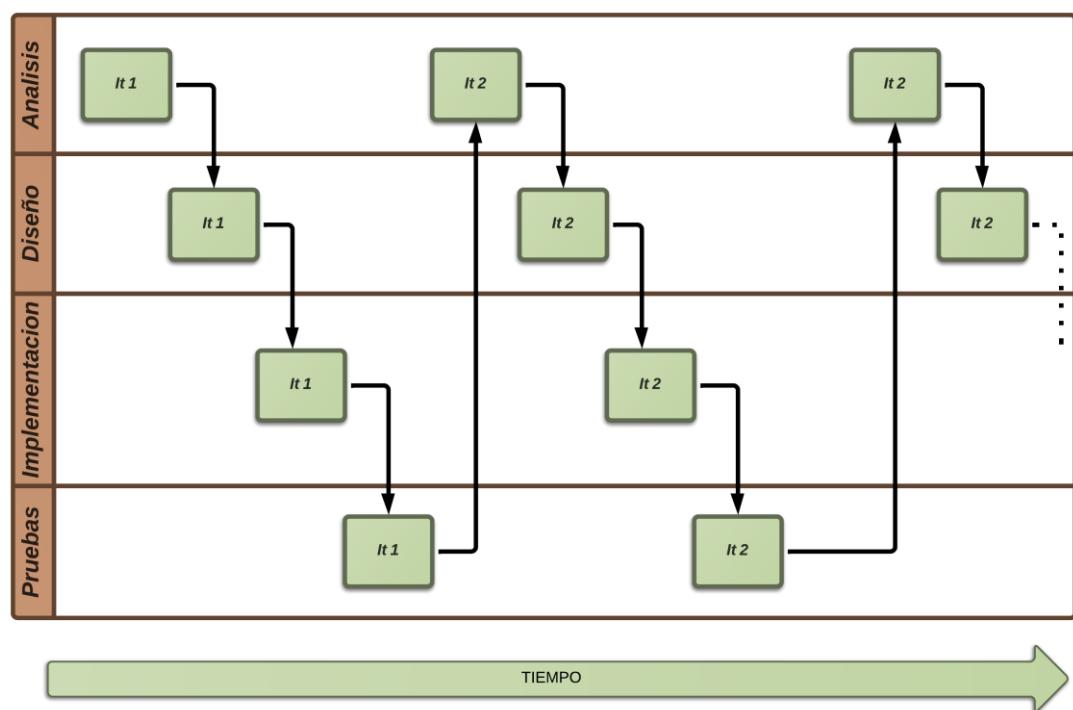


Ilustración 5. Desarrollo iterativo incremental

En cada iteración el equipo evoluciona el producto partiendo de los resultados de la iteración anterior añadiendo a esta los objetivos definidos o mejorando los que ya fueron completados.

2.3.2 Ventajas

Entre las ventajas destacan [13]

- Se puede gestionar las expectativas del cliente de manera regular debido a la toma de decisiones de cada iteración. Se pueden controlar si el cliente no sabe exactamente qué es lo que se necesita o necesita cambios a corto plazo.
- No es necesario realizar una recolección completa de todos los requisitos al inicio del proyecto
- El cliente puede obtener resultados importantes y usables ya desde las primeras iteraciones.
- Se pueden gestionar de manera natural los cambios que vayan surgiendo a lo largo del proyecto, el cliente puede proporcionar su feedback tras examinar los resultados que va obteniendo, y permite planificar los cambios necesarios para próximas iteraciones.
- El cliente como máximo puede perder los recursos dedicados a una iteración, no los de todo el proyecto.
- Al finalizar cada iteración el equipo puede decidir cómo mejorar su trabajo de cara a próximas iteraciones
- Permite mitigar desde el inicio los riesgos del proyecto, desde la primera iteración el equipo puede gestionar los problemas que pueden aparecer en una entrega del proyecto.
- Permite gestionar la complejidad del proyecto. En una iteración solo se trabaja en los requisitos que aportan más valor en ese momento y dividir la complejidad para que cada parte sea resuelta en diferentes iteraciones.
- Dado que cada iteración debe dar como resultado los requisitos terminados, se minimiza el número de errores que se producen en el desarrollo y se aumenta la calidad.

2.3.3 Inconvenientes

Entre los inconvenientes destacan: [13]

- La disponibilidad del cliente debe ser alta durante todo el proyecto dado que participa de manera continua. Al inicio de las iteraciones el cliente ha de detallar (o haber detallado previamente) los requisitos que se van a desarrollar, y al finalizar debe revisar el cumplimiento de los mismos.
- La relación con el cliente ha de estar basada en los principios de la colaboración.
- Cada iteración debe dar como resultado unos requisitos terminados y no dejar tareas pendientes para futuras iteraciones.
- Cada iteración ha de aportar un valor al cliente, entregar unos resultados cerrados.

- Es necesario disponer de técnicas y herramientas que permitan hacer cambios fácilmente en el producto, de manera que se pueda crecer en cada iteración sin hacer un gran esfuerzo adicional, manteniendo su complejidad y calidad.
- Las iteraciones deben tener un tiempo similar, por ello es necesario conocer la velocidad de desarrollo del equipo y tomar las decisiones adecuadas.

2.4 Tecnologías empleadas

En este apartado se describen las tecnologías utilizadas para el desarrollo de la aplicación y la preparación del entorno. Existen multitud de tecnologías aplicables a cualquier desarrollo software. En el caso del hospital VIHrtual al ser un desarrollo web es necesario definir las tecnologías que se usarán en el lado del cliente, para crear la interfaz de usuario, y en el lado del servidor donde se desarrolla la lógica de la aplicación.

2.4.1 Lado del servidor

En el lado del servidor existen multitud de tecnologías de construcción de aplicaciones, como Java EE, PHP, ASP.NET, Perl, etc...

Para el desarrollo de la aplicación se ha optado por utilizar el lenguaje de programación PHP. En cuanto al almacenamiento de la información se puede optar por varias tecnologías para la gestión de dicha información, como son PostgreSQL, MySql, Oracle, etc... En este caso se ha decidido optar por MySql.

2.4.1.1 PHP

Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación. PHP es el acrónimo de Hipertext Preprocesor.

Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor web, justo antes de que se envíe la página a través de Internet al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página con el código HTML resultante de la ejecución de la PHP. Como la página resultante contiene únicamente código HTML, es compatible con todos los navegadores. [14]



Ilustración 6. Funcionamiento PHP

2.4.1.1.1 Ventajas

Las principales características y ventajas destacables de PHP son: [15]

- Orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.

- Es considerado un lenguaje fácil de aprender, ya que en su desarrollo se simplificaron distintas especificaciones, como es el caso de la definición de las variables primitivas, ejemplo que se hace evidente en el uso de php arrays.
- El código fuente escrito en PHP es invisible al navegador web y al cliente, ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- Capacidad de expandir su potencial utilizando módulos (llamados ext's o extensiones).
- Posee una amplia documentación en su sitio web oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite aplicar técnicas de programación orientada a objetos. Incluso aplicaciones como Zend framework, empresa que desarrolla PHP, están totalmente desarrolladas mediante esta metodología.
- No requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.
- Tiene manejo de excepciones (desde PHP5).
- Si bien PHP no obliga a quien lo usa a seguir una determinada metodología a la hora de programar, aun haciéndolo, el programador puede aplicar en su trabajo cualquier técnica de programación o de desarrollo que le permita escribir código ordenado, estructurado y manejable. Un ejemplo de esto son los desarrollos que en PHP se han hecho del patrón de diseño Modelo Vista Controlador (MVC), que permiten separar el tratamiento y acceso a los datos, la lógica de control y la interfaz de usuario en tres componentes independientes.
- Debido a su flexibilidad ha tenido una gran acogida como lenguaje base para las aplicaciones WEB de manejo de contenido, y es su uso principal.

2.4.1.1.2 *Inconvenientes*

Los inconvenientes más destacables son [15]

- Como es un lenguaje que se interpreta en ejecución, para ciertos usos puede resultar un inconveniente que el código fuente no pueda ser ocultado. La ofuscación es una técnica que puede dificultar la lectura del código pero no necesariamente impide que el código sea examinado.

- Debido a que es un lenguaje interpretado, un script en PHP suele funcionar considerablemente más lento que su equivalente en un lenguaje de bajo nivel, sin embargo este inconveniente se puede minimizar con técnicas de caché tanto en archivos como en memoria.
- Las variables al no ser tipificadas dificulta a los diferentes IDEs para ofrecer asistencias para el tipificado del código, aunque esto no es realmente un inconveniente del lenguaje en sí. Esto es solventado por Zend Studio añadiendo un comentario con el tipo a la declaración de la variable.

2.4.1.2 MySql

Es un sistema administrativo relacional de bases de datos (RDBMS por sus siglas en inglés Relational Database Management System). Este tipo de bases de datos puede ejecutar desde acciones tan básicas, como insertar y borrar registros, actualizar información o hacer consultas simples, hasta realizar tareas tan complejas como la aplicación lo requiera. [16]

MySQL es un servidor multi-usuarios muy rápido y robusto de ejecución de instrucciones en paralelo, es decir, que múltiples usuarios distribuidos a lo largo de una red local o Internet podrá ejecutar distintas tareas sobre las bases de datos localizadas en un mismo servidor.

Utiliza el lenguaje SQL (Structured Query Language) que es el estándar de consulta a bases de datos a nivel mundial. MySQL ha estado disponible desde 1996, pero su desarrollo data desde 1979 y ha ganado 3 años consecutivos el premio Linux Journal Reader's Choice Award.

2.4.1.2.1 Ventajas

Las principales características y ventajas destacables de MySQL son: [17]

- El principal objetivo de MySQL es velocidad y robustez.
- Soporta gran cantidad de tipos de datos para las columnas.
- Gran portabilidad entre sistemas, puede trabajar en distintas plataformas y sistemas operativos.
- Cada base de datos cuenta con 3 archivos: Uno de estructura, uno de datos y uno de índice y soporta hasta 32 índices por tabla.
- Aprovecha la potencia de sistemas multiproceso, gracias a su implementación multi-hilo.
- Velocidad al realizar las operaciones, lo que le hace uno de los gestores con mejor rendimiento.
- Bajo costo en requerimientos para la elaboración de bases de datos, ya que debido a su bajo consumo puede ser ejecutado en una máquina con escasos recursos sin ningún problema.
- Facilidad de configuración e instalación.

- Soporta gran variedad de Sistemas Operativos
- Baja probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor, sino en el sistema en el que está.
- Conectividad y seguridad

2.4.1.2.2 Inconvenientes

Las desventajas encontradas para MySql son mínimas:

- Un gran porcentaje de las utilidades de MySQL no están documentadas.
- No es tan intuitivo, como otros programas (Microsoft ACCESS).

2.4.1.3 Protocolo SIP

SIP es el protocolo de comunicación utilizado para las videollamadas entre pacientes y el profesional sanitario.

SIP, o Session Initiation Protocol es un protocolo de control y señalización usado mayoritariamente en los sistemas de Telefonía IP, que fue desarrollado por el IETF (RFC 3261). Dicho protocolo permite crear, modificar y finalizar sesiones multimedia con uno o más participantes y sus mayores ventajas recaen en su simplicidad y consistencia.[18]

Hasta la fecha, existían múltiples protocolos de señalización tales como el H.323 de la ITU, el SCCP de Cisco, o el MGCP, pero parece que poco a poco SIP está ganando la batalla del estándar: Cisco está progresivamente adoptando SIP como protocolo en sus sistemas de telefonía IP en detrimento de H.323 y SCCP, Microsoft ha elegido SIP como protocolo para su nuevo OCS (Office Communication Server), y los operadores (de móvil y fijo) también están implantando SIP dentro de su estrategia de convergencia, aprovechando de este modo la escalabilidad e interoperabilidad que nos proporciona el protocolo SIP.

2.4.2 Lado del cliente

Las tecnologías del lado del cliente son las que se ejecutan en el navegador del usuario, estas tecnologías son muy dependientes del sistema donde se están ejecutando, al existir multitud de sistemas operativos y gran cantidad de navegadores web que dependiendo de su versión pueden ser totalmente diferentes o incompatibles con las últimas tecnologías.

Uno de los puntos importantes a la hora de que la plataforma sea totalmente compatible con todos los dispositivos es la utilización de las últimas tecnologías web del lado del cliente que esté disponibles.

2.4.2.1 HTML5

HTML, siglas de HyperText Markup Language (lenguaje de marcas de hipertexto), hace referencia al lenguaje de marcado para la elaboración de páginas web. [19]

Es un estándar que sirve de referencia para la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, etc. Es un estándar a cargo de la W3C, organización dedicada a la estandarización de casi todas las tecnologías ligadas a la web, sobre todo en lo referente a su escritura e interpretación. Es el lenguaje con el que se definen las páginas web.

Para el desarrollo de la plataforma se ha seguido la quinta versión de este lenguaje. HTML5 establece una serie de nuevos elementos y atributos que reflejan el uso típico de los sitios web modernos. Algunos de ellos son técnicamente similares a las etiquetas `<div>` y ``, pero tienen un significado semántico, como por ejemplo `<nav>` (bloque de navegación del sitio web) y `<footer>`. Otros elementos proporcionan nuevas funcionalidades a través de una interfaz estandarizada, como los elementos `<audio>` y `<video>`. Mejora el elemento `<canvas>`, capaz de renderizar elementos 3D en los navegadores más importantes (Mozilla, Chrome, Opera, Safari e IE).[20]

2.4.2.2 CSS3

CSS es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. [21]

Es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas. Separar la definición de los contenidos y la definición de su aspecto presenta numerosas ventajas, ya que obliga a crear documentos HTML/XHTML bien definidos y con significado completo (también llamados "documentos semánticos"). Además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes.

La última versión disponible actualmente es CSS3 que incorpora una gran cantidad de propiedades nuevas e introduce la posibilidad de crear efectos 3D.

2.4.2.3 JavaScript

JavaScript es un lenguaje de programación que surgió con el objetivo inicial de programar ciertos comportamientos sobre las páginas web, respondiendo a la interacción del usuario y la realización de automatismos sencillos. En ese contexto podríamos decir que nació como un "lenguaje de scripting" del lado del cliente, sin embargo, hoy JavaScript es mucho más. Las necesidades de las aplicaciones web modernas y el HTML5 han provocado que el uso de JavaScript que encontramos hoy haya llegado a unos niveles de complejidad y prestaciones tan grandes como otros lenguajes de primer nivel.[22]

2.4.2.4 Frameworks, librerías y plugins

Para facilitar la labor de desarrollo se utilizan módulos externos que se integran con la aplicación y agilizan tareas que en caso contrario conllevarían a desarrollos complejos y más largos.

2.4.2.4.1 Ajax

Ajax es un framework para aplicaciones web que ayuda a desarrollar aplicaciones basadas en esta tecnología que permite construir páginas web dinámicas del lado del cliente. [23]

La información es leída desde el servidor o enviada a éste a través de peticiones JavaScript. Sin embargo, es requerido algún procesamiento del lado del servidor para manejar peticiones, por ejemplo, para buscar o guardar información.

Este motor AJAX pretende reducir la espera para el usuario cuando una página trata de acceder al servidor. La meta del framework es proveer este motor AJAX y funciones asociadas al servidor y del lado del cliente.

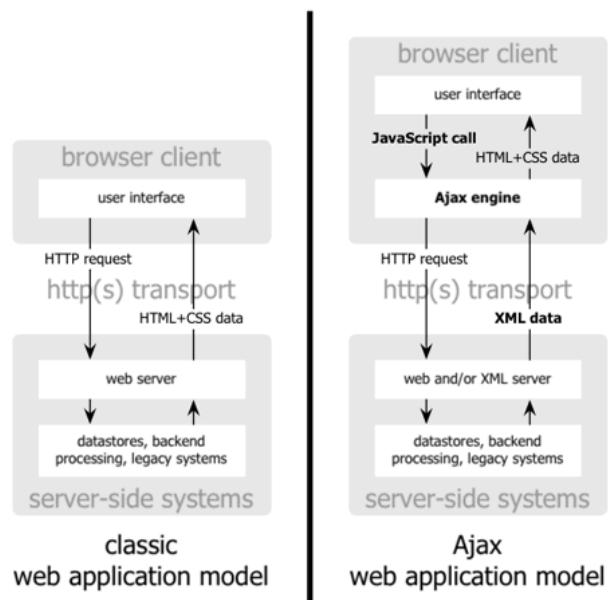


Ilustración 7. Peticiones al servidor clásicas y con Ajax

Para la navegación e interacción en la página se ha optado por este framework, así el usuario a la hora de interactuar con la pagina puede ver una mayor fluidez en la carga de cada vista al no realizarse recargarse la página por cada interacción que realice sobre ella.

2.4.2.4.2 SIPml5

Es el primer cliente HMTL5 SIP open source del mundo, con licencia BSD y escrito completamente en JavaScript. No necesita de la instalación de ningún plugin ya que la comunicación se basa en WebRTC.

Se puede utilizar para conectarse a cualquier servidor SIP para realizar y recibir llamadas de audio y/o video.

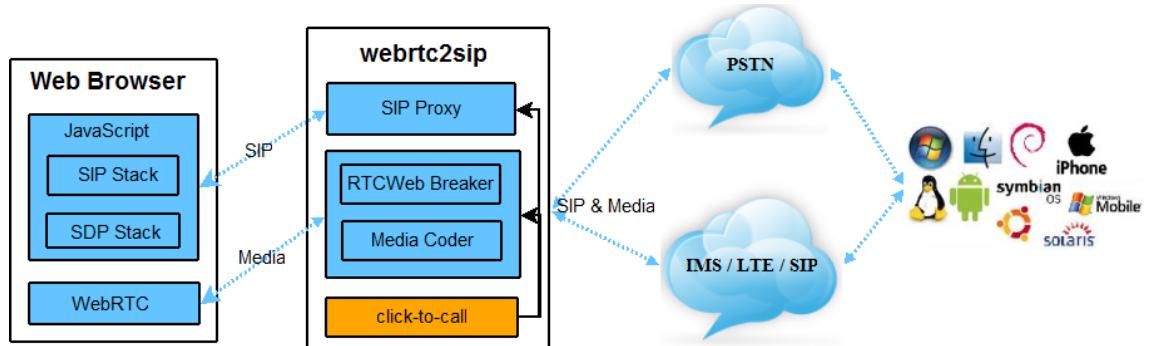


Ilustración 8. Sipml5

WebRTC, también conocido como Web Real-Time Communication, es un proyecto de código abierto – promovido por Google, Mozilla y otros – que permite comunicaciones en tiempo real sin plugins a través de una API JavaScript. Facilita las aplicaciones de llamadas de voz, chat de video y compartimiento de archivos entre navegadores. El codec soportado actualmente para WebRTC es VP8. WebRTC utiliza un servidor denominado Servidor de Conferencias Web que en conjunto con un Servidor STUN es requerido para proveer la página inicial y sincronizar las conexiones entre dos nodos WebRTC. [24]

2.4.2.4.3 Bootstrap

Bootstrap, es un framework originalmente creado por Twitter, que permite crear interfaces web con CSS y JavaScript, cuya particularidad es la de adaptar la interfaz del sitio web al tamaño del dispositivo en que se visualice. Es decir, el sitio web se adapta automáticamente al tamaño de una PC, una Tablet u otro dispositivo. Esta técnica de diseño y desarrollo se conoce como “responsive design” o diseño adaptativo.[25]

El beneficio de usar responsive design en un sitio web, es principalmente que el sitio web se adapta automáticamente al dispositivo desde donde se acceda.

2.4.2.4.4 jQuery

jQuery es una biblioteca de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. Fue presentada el 14 de enero de 2006 en el BarCamp NYC. JQuery es la biblioteca de JavaScript más utilizada.

jQuery es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privativos.

Al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.[26]

2.4.2.4.5 OWL Carousel

Owl carousel es un plugin de jQuery que permite construir carruseles de elementos. Las principales propiedades por las cuales se utiliza en la plataforma es su compatibilidad con dispositivos táctiles, es completamente *responsive*, está basado en CSS3 y tiene compatibilidad con todos los navegadores.

2.4.3 Preparación del entorno

Antes de iniciar el desarrollo de la aplicación se necesita de una serie de herramientas que faciliten la labor del desarrollador.

Al ser una plataforma web es necesario tener primero un entorno de programación, un lugar de despliegue de la web, un lugar de almacenamiento y un repositorio remoto donde mantener el código para futuras desarrollos o despliegues.

2.4.3.1 Entorno de programación

El IDE (Integrated Development Environment) que se ha utilizado para el desarrollo de la aplicación ha sido PHPStorm, el cual es un entorno únicamente para el desarrollo web, que proporciona todo lo necesario para gestionar un proyecto web en PHP. Incluyendo soporte a los lenguajes mencionados en el punto 2.4.2 necesarios para el desarrollo de esta web.

Existen multitud de entornos de desarrollo, pero suelen ser para multitud de lenguajes y no se centran exclusivamente en PHP como es el caso de PHPStorm. Cualquier desarrollo web puede hacerse con un simple editor de texto, pero gracias a los IDEs tenemos agrupadas la gestión de todos los archivos, un editor de texto con ayudas para cada lenguaje, analizadores de código, posibilidad de hacer testeos y debuggear de forma más sencilla.

2.4.3.2 Servidor web y base de datos

Toda aplicación PHP debe ser ejecutada en un servidor web capaz de interpretar scripts de este lenguaje. Para ello el elegido es Apache, el cual es un servidor web HTTP de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.12 y la noción de sitio virtual. [27]

Como se ha mencionado anteriormente la elección para la gestión de la base de datos es Mysql.

Para el desarrollo en local se ha optado por usar XAMPP, el cual proporciona lo mencionado anteriormente. XAMPP es un servidor independiente de plataforma, software libre, que consiste principalmente en la base de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script: PHP y Perl. El nombre proviene del acrónimo de X (para cualquiera de los diferentes sistemas operativos), Apache, MySQL, PHP, Perl. [28]

El programa está liberado bajo la licencia GNU y actúa como un servidor web libre, fácil de usar y capaz de interpretar páginas dinámicas.

2.4.3.3 Repositorio de código

Todo desarrollo software debe tener asociado un control de versiones que gestione los cambios que se realicen sobre el código fuente del producto. Este control puede hacerse de forma manual o con herramientas que faciliten.

Para la aplicación se ha utilizado Git, el cual es un software de control de versiones distribuido pensado para la eficiencia y confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente.[29]

El diseño de Git resulta de la experiencia del diseñador de Linux, Linus Torvalds, manteniendo una enorme cantidad de código distribuido y gestionado por mucha gente, que incide en numerosos detalles de rendimiento.

Dos de las características que favorecen al uso de Git frente a otros son: [30]

- La primera gran diferencia de Git con respecto a otros sistemas de control de versiones es la forma que tiene de manejar los cambios en los ficheros. Mientras que otros scv's (sistemas de control de versiones) almacenan los archivos originales, conservando una lista de los cambios realizados a dichos archivos en cada versión, Git guarda una “foto” (snapshot) del estado de cada archivo en un momento concreto. Si uno de los archivos no ha cambiado no crea una nueva copia del mismo, simplemente crea una referencia al archivo original.
- La segunda es la eficiencia. Git se basa en que cada programador almacena una copia completa del repositorio en su máquina de forma local, incluido el historial de cambios. Esto implica que muchas de las operaciones realizadas sobre el código fuente no tienen lugar en la red, permitiendo que la velocidad de proceso dependa únicamente en los recursos locales.

2.5 Modelado de datos conceptual

Como primer paso para una correcta definición de la estructura de datos hay que identificar todos los elementos que intervienen y sus relaciones.

Dentro de la etapa de análisis lo importante es recoger la máxima cantidad de información posible, sin importar los detalles y sí representando únicamente la información relevante.

Independientemente del sistema gestor de base de datos hay que realizar una visión global de todos los datos de la aplicación, para ello se ha creado el siguiente modelo de Entidad-Relación (E/R)

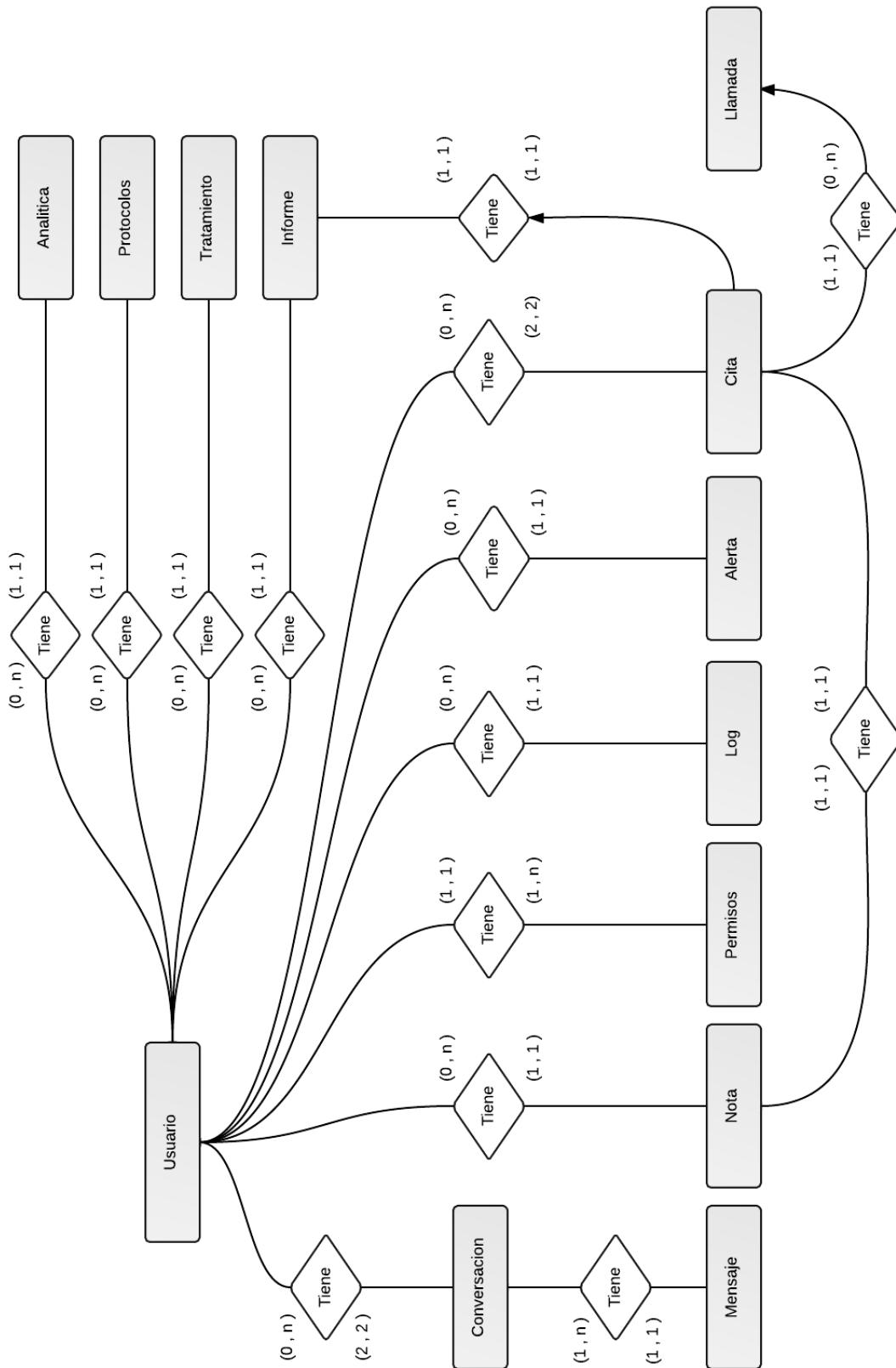


Ilustración 9. Modelo entidad/relación

2.6 Diagramas de casos de uso

Los diagramas de casos de uso nos muestran de forma gráfica las acciones posibles del usuario sobre el sistema. Estas acciones han sido definidas en los requisitos funcionales.

El diagrama básico muestra cuál es la integración con el entorno, donde se tienen los diferentes usuarios o dispositivos externos y el sistema:

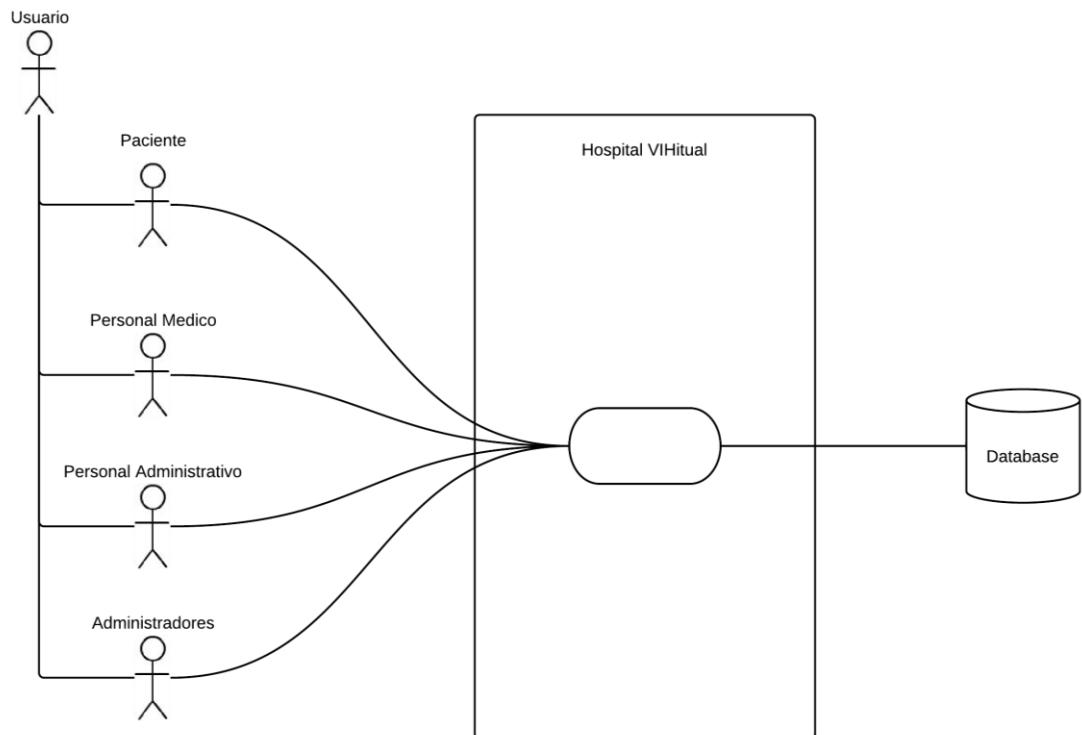


Ilustración 10 Diagrama casos de uso entorno

Dado que la aplicación tiene múltiples funciones vamos a separar los casos de uso para cada una de ellas, pero en primer lugar veremos todas las posibles funcionalidades después de acceder a la plataforma:

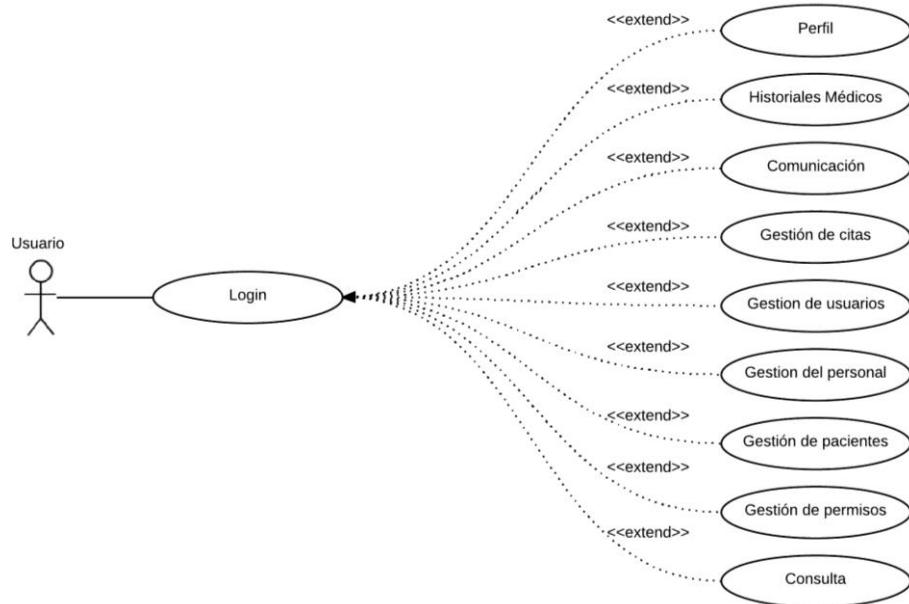


Ilustración 11. Diagrama casos de uso entrada

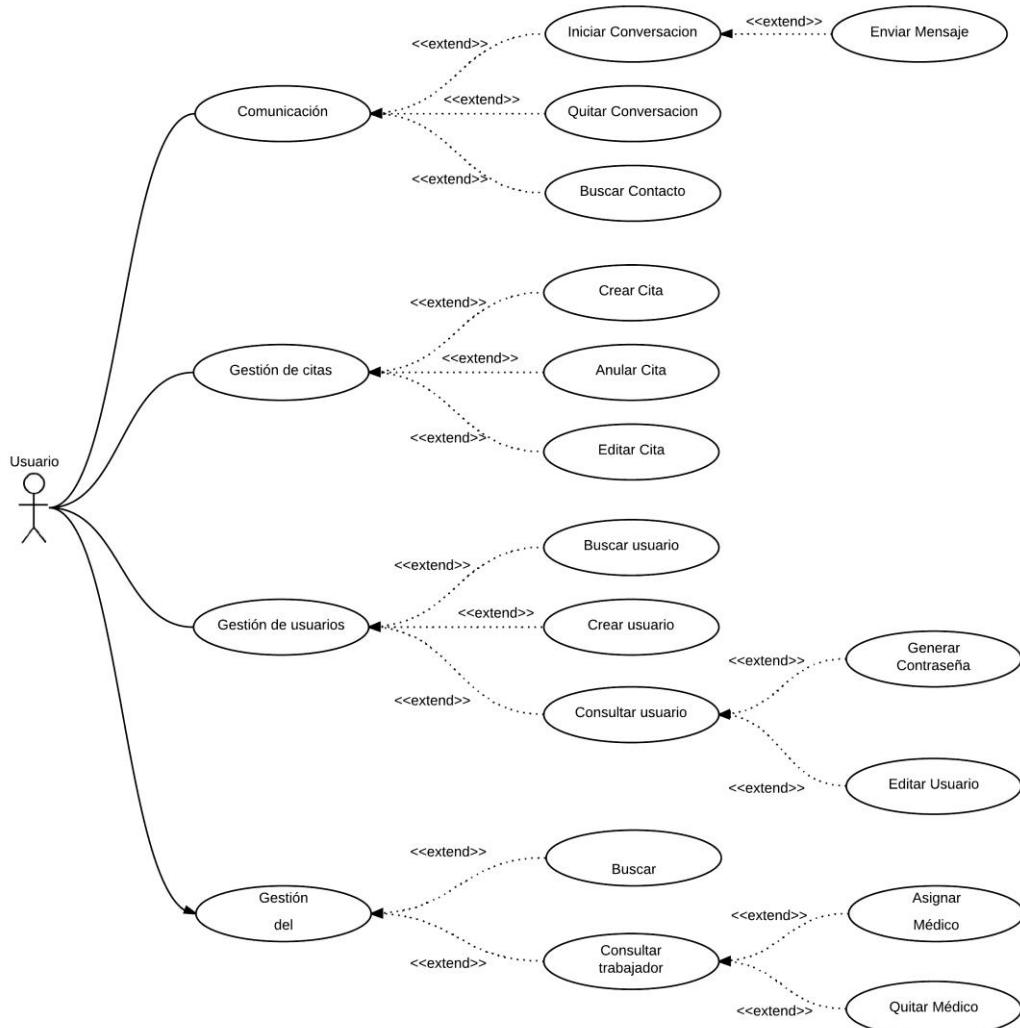


Ilustración 12. Diagrama casos de uso funcionalidades I

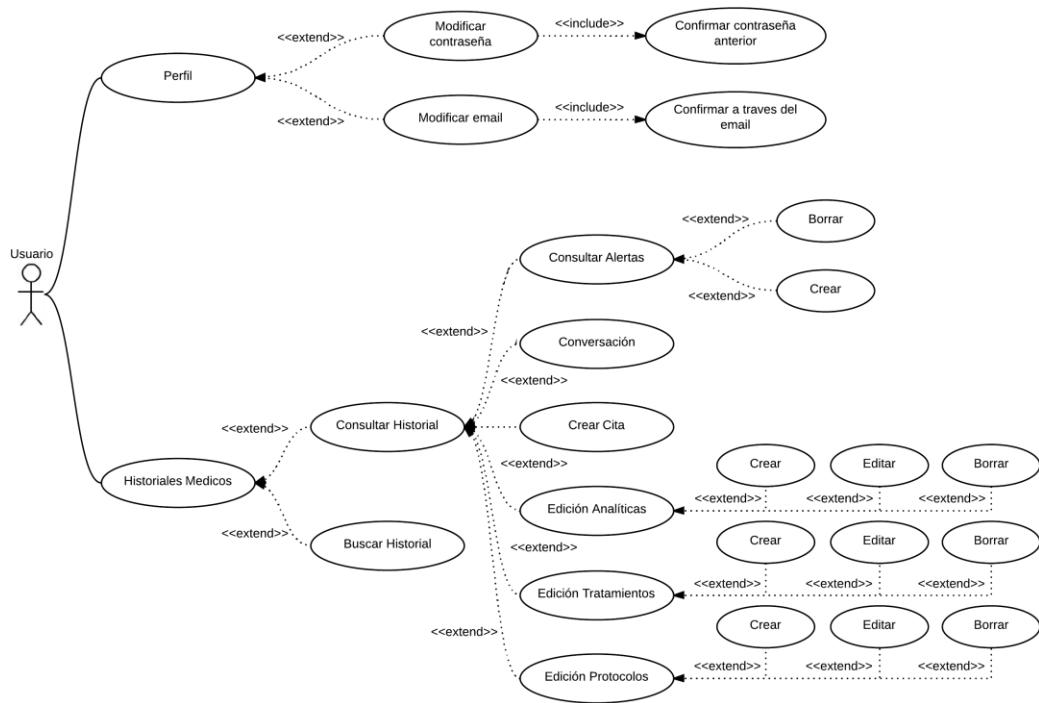


Ilustración 13. Diagrama casos de uso funcionalidades 2

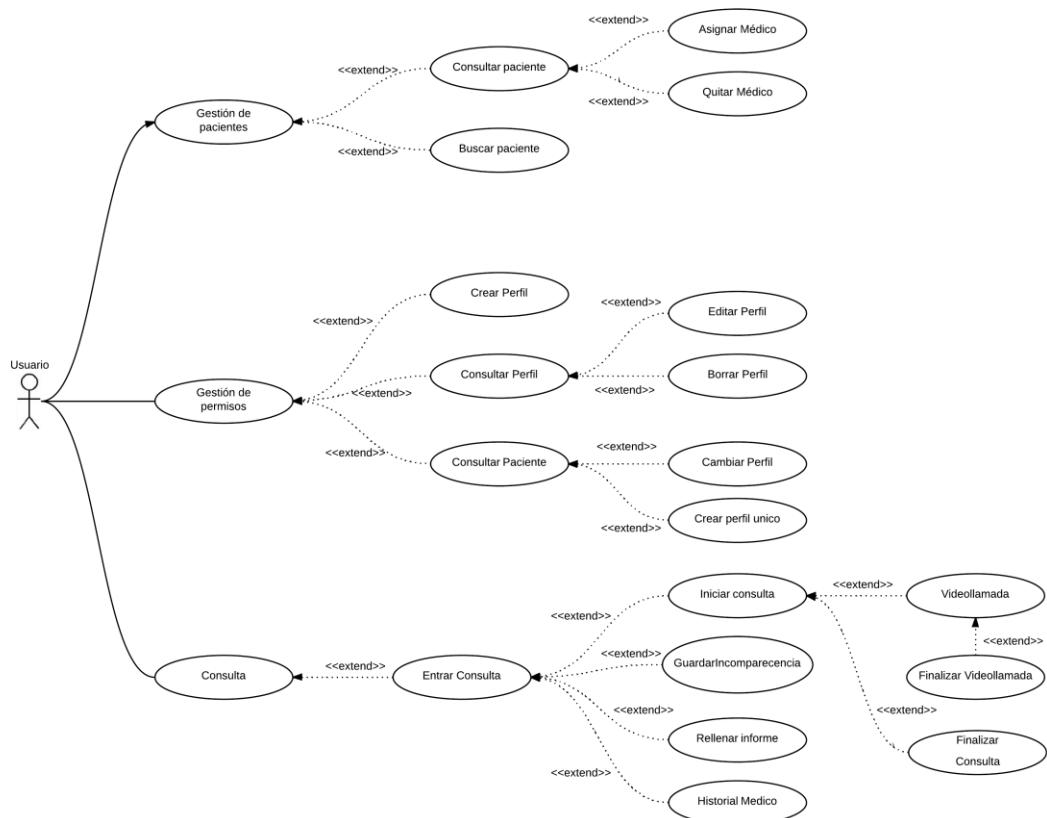


Ilustración 14. Diagrama casos de uso funcionalidades 3

3. Diseño

En este capítulo, tras el análisis realizado se procede al modelado de datos, definición de la arquitectura y diagrama de clases, así como el diseño de la interfaz y los modos de evaluación de la usabilidad y accesibilidad.

3.1 Modelado lógico de datos

En el punto 2.5 de la memoria se definió el modelo de datos conceptual, esto se va a transformar en un modelo lógico con el que se describirá la estructura de datos.

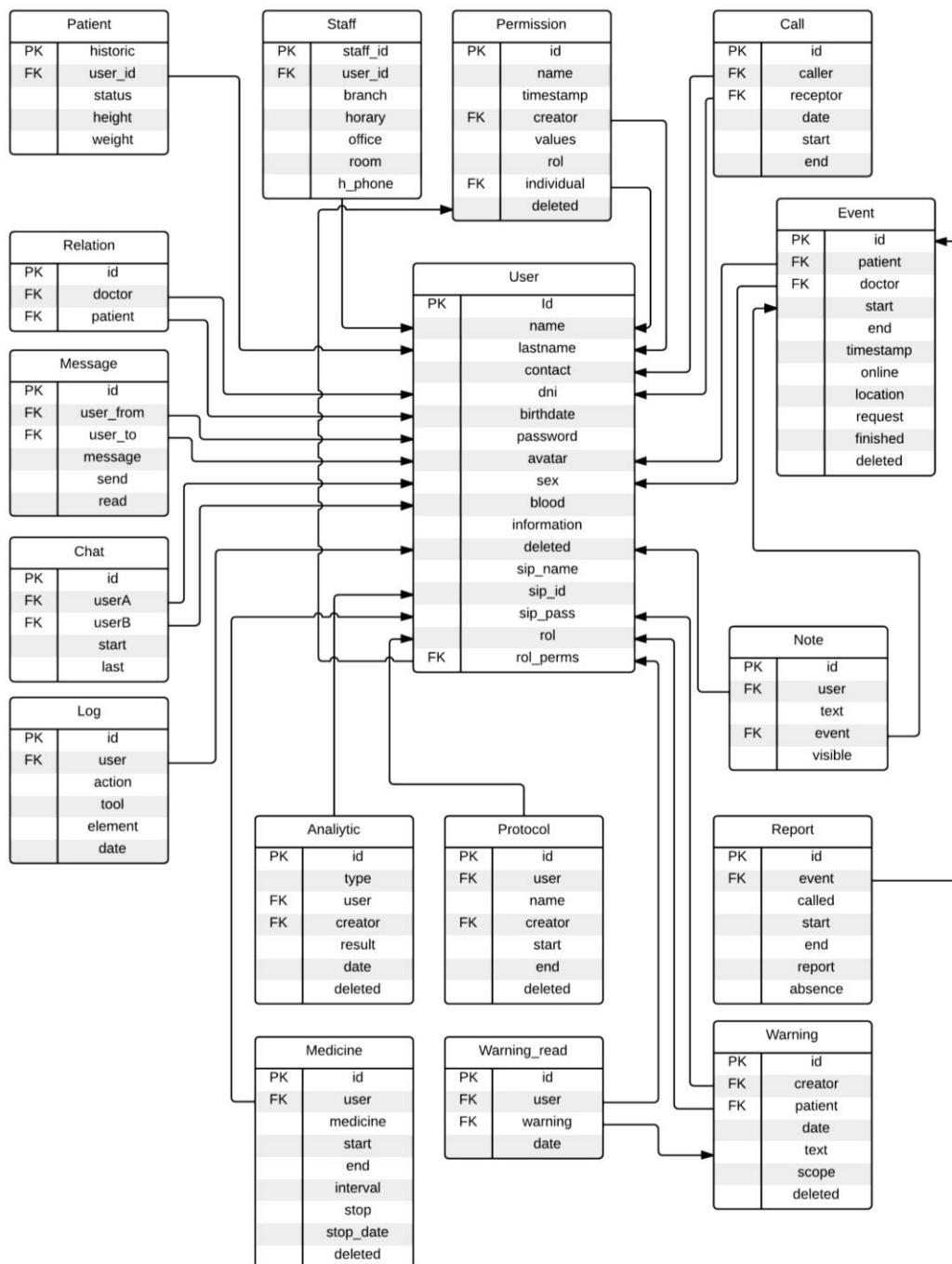


Ilustración 15. Modelo de datos lógico

3.2 Arquitectura

La arquitectura al ser una aplicación se denomina cliente-servidor, en la que tenemos una separación entre la lógica y la presentación.

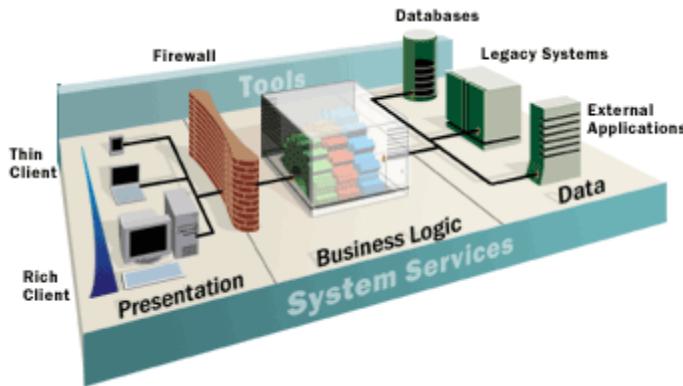


Ilustración 16. Arquitectura tres niveles

Las tres capas o niveles que representan el sistema son:

- Capa de presentación:

Es la capa que se le presenta al usuario, a través de ella se le presenta toda la información en un mínimo proceso. Recibe también el nombre de interfaz gráfica. Se comunica con la capa de negocio.

- Capa de negocio:

En ella reside toda la lógica de la aplicación, por ello también es llamada también capa lógica de negocio. Recibe las peticiones de la capa de presentación, las procesa para comunicarse con la capa de datos y devolver toda la información a la capa de presentación.

- Capa de datos:

Todos los datos de la aplicación residen aquí, y es la encargada de acceder a los mismos. Puede estar formada por uno o varios gestores de bases de datos y recibe solicitudes de almacenamiento de datos o de recuperación desde la capa de negocio.

En la aplicación la capa de presentación reside en los ordenadores o dispositivos móviles de los usuarios, y la capa de negocio y capa de datos residen en el mismo servidor, pudiéndose dar el caso de estar en distintos.

3.3 MVC

El modelo vista controlador es un patrón de arquitectura de software que separa los datos y la lógica del negocio de la aplicación.

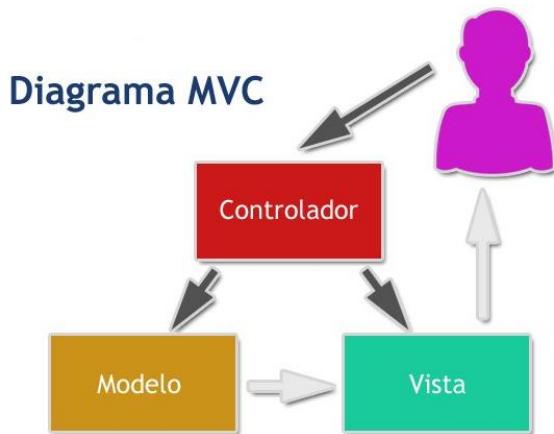


Ilustración 17. Modelo-Vista-Controlador

Este patrón se basa en la solicitud del usuario información al servidor, esta se procesa a través del controlador que hace de puente entre los modelos y las vistas, se producen las salidas y se devuelven al usuario.[31]

3.3.1 Modelo

Es la capa donde se trabaja con los datos, por tanto contendrá mecanismos para acceder a la información y también para actualizar su estado. Los datos los tendremos habitualmente en una base de datos, por lo que en los modelos tendremos todas las funciones que accederán a las tablas y harán los correspondientes selects, updates, inserts, etc.

3.3.2 Vista

Las vistas, como su nombre nos hace entender, contienen el código de nuestra aplicación que va a producir la visualización de las interfaces de usuario, o sea, el código que nos permitirá renderizar los estados de nuestra aplicación en HTML. En las vistas nada más tenemos los códigos HTML y PHP que nos permite mostrar la salida.

3.3.3 Controlador

Contiene el código necesario para responder a las acciones que se solicitan en la aplicación, como visualizar un elemento, realizar una compra, una búsqueda de información, etc.

3.4 Diagrama de clases

El diagrama de clases nos describe la estructura de un sistema mostrando sus clases, está orientado a objetos. Cada objeto representado pertenece a una clase.

A continuación tenemos el diagrama de clases de la aplicación:

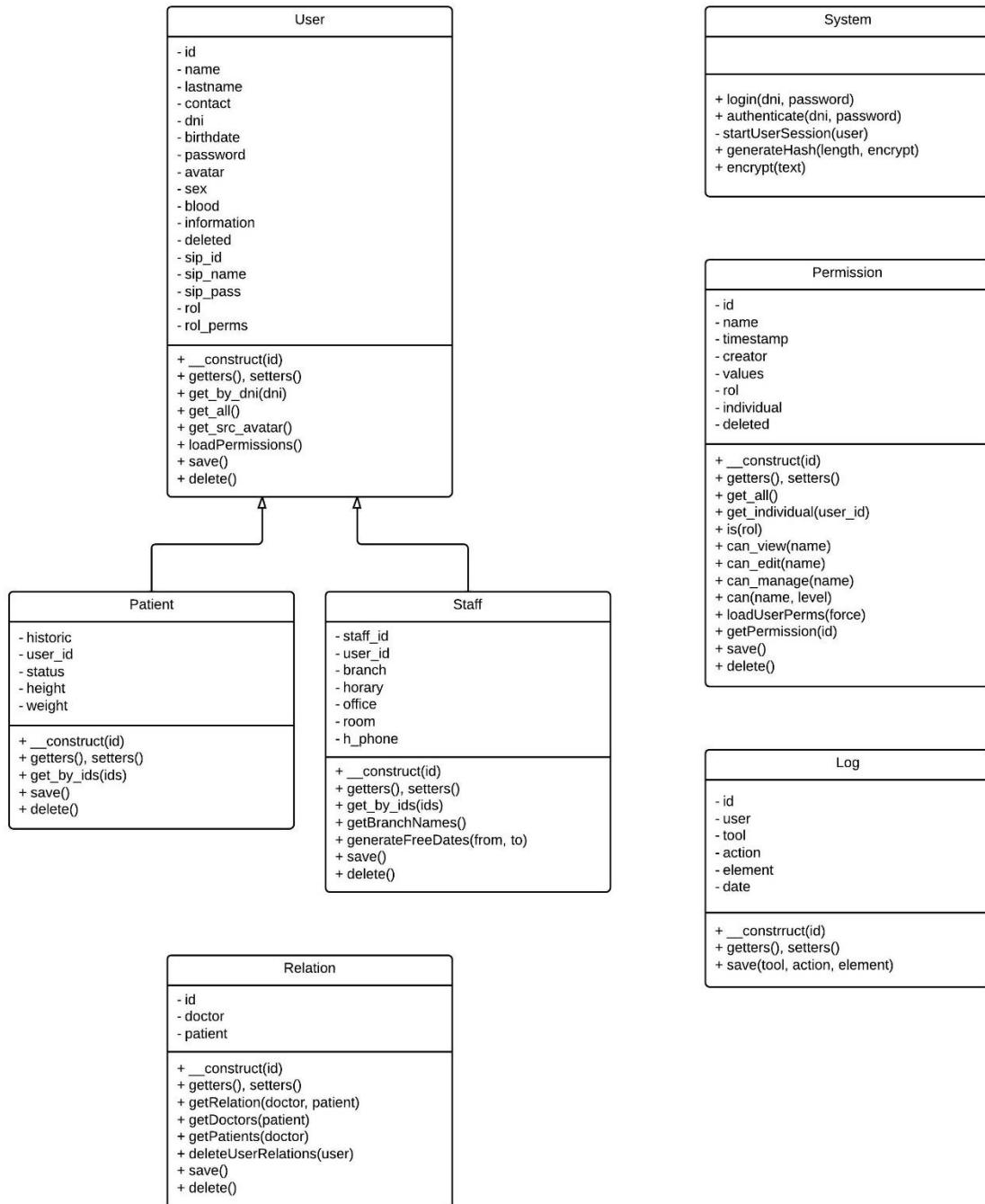


Ilustración 18. Diagrama clases parte 1

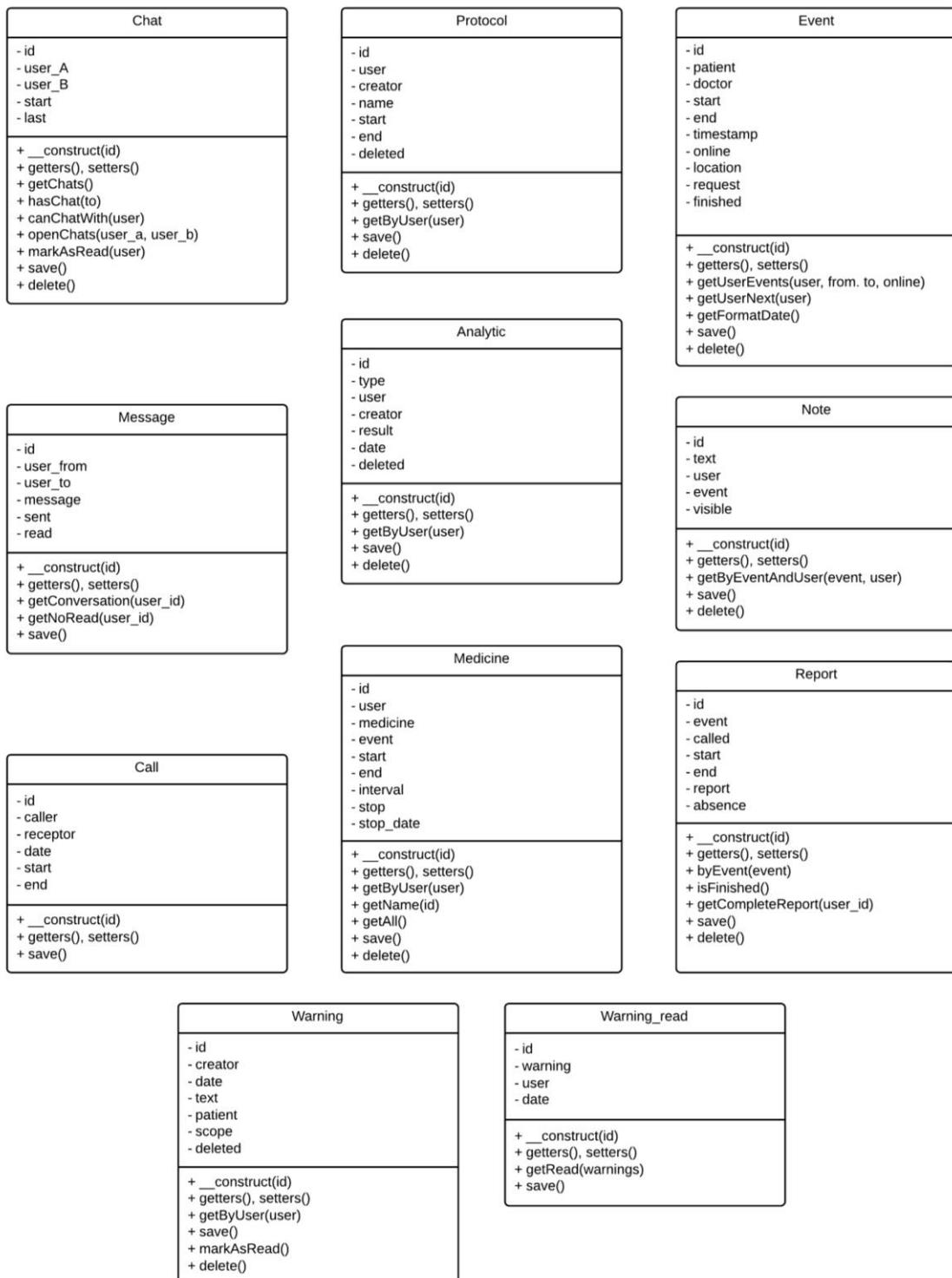


Ilustración 19. Diagrama de clases parte 2

3.5 Diseño de la interfaz grafica

El proceso de diseño de la interfaz gráfica es uno de los puntos más importantes, ya que determina en gran medida la percepción e impresión que el usuario posee sobre la plataforma. Toda la lógica estará representada por la interfaz, por ello debe funcionar correctamente y ser usable por cualquier tipo de usuario.

Para facilitar el proceso del diseño se han ido realizando prototipos de cada una de las partes de la plataforma para poder verificar que todas las funcionalidades y requisitos se cumplen.

La interfaz debe adaptarse a multitud de tamaños de pantallas distintos, por ello la estructura base de la interfaz debe ser definida lo más precisa posible. Es la siguiente:

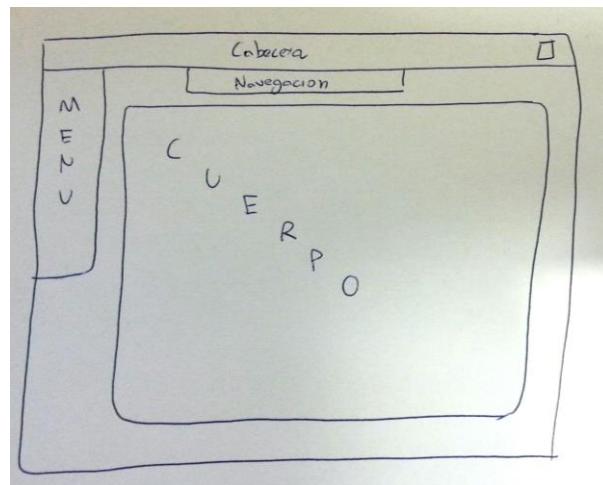


Ilustración 20. Diseño pantallas grandes

Como la plataforma debe ser adaptable a todos los dispositivos hay que diferenciar entre dos estructuras que convivan juntas y se adaptan gracias al diseño *responsive*, que siempre se ajusta completamente a la pantalla y muestra dos formas de visualización diferentes partiendo de la misma base, por ello al acceder desde un dispositivo de menor tamaño tendremos la siguiente distribución sin necesidad de tener una aplicación independiente:

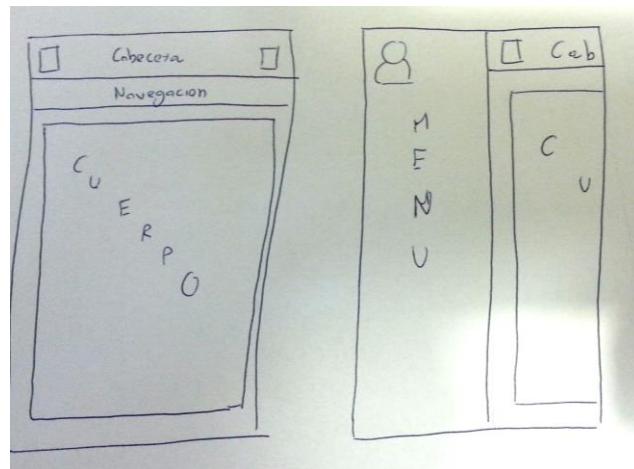


Ilustración 21. Diseño dispositivos móviles

3.6 Usabilidad

La usabilidad en el ámbito del desarrollo web es la disciplina que estudia cuál es la forma de diseñar para que los usuarios puedan interactuar con ella de la forma más fácil, cómoda e intuitiva posible. El diseño no es en sí mismo usable, lo es para usuarios específicos en contextos de uso específicos. Depende por completo de la audiencia a la que esté dirigida.

Por ello la mejor forma es realizando un diseño centrado en el usuario, diseñando cada detalle para y por el usuario. Hay dos tipos de atributos para cuantificar esta usabilidad:

- Cuantificables de forma objetiva:

Son la eficacia o número de errores cometidos por el usuario durante la realización de la tarea, y eficiencia o tiempo empleado por el usuario para la consecución de la misma

- Cuantificables de forma subjetiva:

Es la satisfacción del usuario con el uso de la plataforma, que puede ser medida con preguntas al usuario.

La usabilidad conlleva beneficios:

- Reducción de los costes de producción:

Evitando los rediseños y así reduciendo los cambios posteriores

- Reducción de los costes de mantenimiento y apoyo:

Si el sistema es usable requiere menos entrenamiento y soporte

- Reducción de los costes de uso:

Mejoran la productividad

- Mejora en la calidad del producto:

La sociedad ante un mercado tan amplio demanda productos de fácil uso.

Por ello para la evaluación de la usabilidad no hay métricas exactas, a continuación se describirán los principios en los cuales hay que basarse, y en el punto 4.4 se realizará una evaluación de la usabilidad de la plataforma.

3.6.1 Facilidad de aprendizaje

Es el tiempo requerido desde el no conocimiento de la aplicación hasta el uso productivo debe ser mínimo. Por ello debe proporcionarse esa facilidad para que cualquier usuario alcance un nivel de conocimiento y uso del sistema máximos. Los principios que se toman como referencia para la interfaz son:

- Predecible:

Hay que evitar que el usuario realice acciones incorrectas y dar la opción de deshacer acciones realizadas. Toda acción debe ser clara e informar al usuario de lo que sucederá al realizarla, y facilitar al usuario a realizar las acciones con retroalimentación de las acciones que esté realizando.

- Familiar:

Debe existir una correlación entre los conocimientos que el usuario posee de la interacción con el mundo real u otros sistemas y los conocimientos requeridos para la interacción con un sistema nuevo

- Consistente

Para que el sistema sea consistente todos los mecanismos deben ser usados siempre de la misma manera sea la acción y momento que sea. Por ello se debe mantener un estilo y estructura común en toda la aplicación.

- Fácil de recordar

Cada vez que el usuario vuelve a entrar en la aplicación no debe ser necesario tener que volver a realizar un aprendizaje. Se debe favorecer al reconocimiento sobre el recuerdo intentando así disminuir la carga cognitiva del usuario.

3.6.2 Flexibilidad

Determinada por la variedad de posibilidades con la que un usuario y la aplicación pueden intercambiar información y la multiplicidad de vías para realizar las tareas.

3.6.3 Robustez

Determinada por el nivel de apoyo que se da al usuario para el cumplimiento de sus objetivos. Hay cuatro principios que determinan este nivel:

- Observable:

El usuario debe tener la capacidad de evaluar el estado interno del sistema a partir de la representación de la interfaz.

- Capacidad de recuperación:

La aplicación debe permitir al usuario corregir una acción una vez que ha sido reconocida como errónea.

- Realimentación en caso de error:

En caso de producirse un error se debe informar al usuario de forma clara lo sucedido.

- Ajuste de tareas:

El usuario es el que debe llevar el control sobre la aplicación permitiendo suspender acciones, realizarlas en la medida de lo posible en el orden deseado.

3.7 Accesibilidad

Una web accesible es aquella que independientemente de las características del usuario, ya sean habilidades cognitivas, motoras, capacidades, cultura o la infraestructura de acceso pueda utilizarla.

Por ello se intenta que la interfaz tenga un diseño universal. La accesibilidad Web hace referencia a un diseño Web que va a permitir que estas personas puedan percibir, entender, navegar e interactuar con la Web, aportando a su vez contenidos. La accesibilidad Web también beneficia a otras personas, incluyendo personas de edad avanzada que han visto mermadas sus habilidad a consecuencia de la edad.

La accesibilidad Web engloba muchos tipos de discapacidades, incluyendo problemas visuales, auditivos, físicos, cognitivos, neurológicos y del habla. El documento "Cómo utilizan la Web personas con discapacidad" muestra la forma en la que diferentes discapacidades pueden dificultar la utilización de la Web e incluye algunos escenarios de personas con discapacidad utilizando la Web.[32]

El máximo organismo dentro de la jerarquía de Internet que se encarga de promover la accesibilidad es el World Wide Web Consortium (W3C), en especial su grupo de trabajo Web Accessibility Initiative (WAI). En 1999 el WAI publicó la versión 1.0 de sus pautas de accesibilidad Web. Con el paso del tiempo se han convertido en un referente internacionalmente aceptado. En diciembre del 2008 las WCAG 2.0 fueron aprobadas como recomendación oficial.[33]

A continuación se mostraran los principios y pautas definidos por el WCAG 2.0 que en el punto 5 serán evaluados. [34]

3.7.1 Principio 1: Perceptible

La información y los componentes de la interfaz de usuario deben ser presentados a los usuarios de forma que ellos puedan percibirlos.

Algunas pautas son:

1. Alternativas Textuales

Proporcionar alternativas textuales para todo contenido no textual de modo que se pueda convertir a otros formatos que las personas necesiten, tales como textos ampliados, braille, voz, símbolos o en un lenguaje más simple.

2. Medios dependientes del tiempo

Proporcionar alternativas para los medios dependientes del tiempo.

3. Adaptable

Crear contenido que pueda presentarse de diferentes formas (por ejemplo, con una disposición más simple) sin perder información o estructura.

4. Distinguible

Facilitar a los usuarios ver y oír el contenido, incluyendo la separación entre el primer plano y el fondo.

3.7.2 Principio 2: Operable

Los componentes de la interfaz de usuario y la navegación deben ser operables.

Algunas pautas son:

1. Accesible por teclado

Proporcionar acceso a toda funcionalidad mediante el teclado

2. Tiempo suficiente

Proporcionar a los usuarios el tiempo suficiente para leer y usar el contenido

3. Convulsiones

No diseñar el contenido de modo que se sepa que podría provocar ataques, espasmos o convulsiones

4. Navegable

Proporcionar medios para ayudar a los usuarios a navegar, encontrar contenido y determinar donde se encuentran

3.7.3 Principio 3: Comprensible

La información y el manejo de la interfaz deben ser comprensibles.

Algunas pautas son:

1. Legible

Hacer que los contenidos textuales resulten legibles y comprensibles

2. Predecible

Hacer que las páginas web aparezcan y operen de manera predecible

3. Entrada de datos asistida

Ayudar a los usuarios a evitar y corregir los errores

3.6.4 Principio 4: Robusto

El contenido debe ser suficientemente robusto como para ser interpretado de forma fiable por una amplia variedad de aplicaciones de usuario, incluyendo las ayudas técnicas.

Algunas pautas son:

1. Compatible

Maximizar la compatibilidad con las aplicaciones de usuario actuales y futuras, incluyendo las ayudas técnicas.

4. Implementación

En este capítulo, partiendo del análisis y el diseño realizado anteriormente se procede al desarrollo de la aplicación. En primer lugar, se mostrarán partes del código fuente, el control de versiones que se ha seguido y por último, se verá la interfaz de usuario y su evaluación.

4.1 Código fuente

Una vez resueltos todos los detalles anteriormente mencionados, se procede a programar la solución.

4.1.1 Estructura

La estructura del código se ha realizado agrupando contenidos en función a su uso. Tenemos la carpeta principal llamada pfg que contiene lo siguiente.

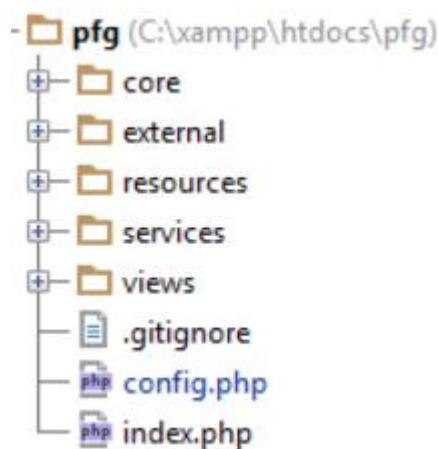


Ilustración 22. Estructura código 1

Como se puede ver se tienen cinco carpetas que se verán en detalle más adelante y tres ficheros:

- **gitignore:**
En este fichero se especifican las rutas o ficheros que no se quieren trackear en el repositorio de Git.
- **config.php**
Archivo básico de php que se incluye en todos los ficheros que incluye valores básicos y el control de la sesión del usuario.
- **index.php**
Fichero que contiene la portada de la plataforma y el control de acceso del usuario.

Las carpetas que forman cada parte del proyecto son las siguientes

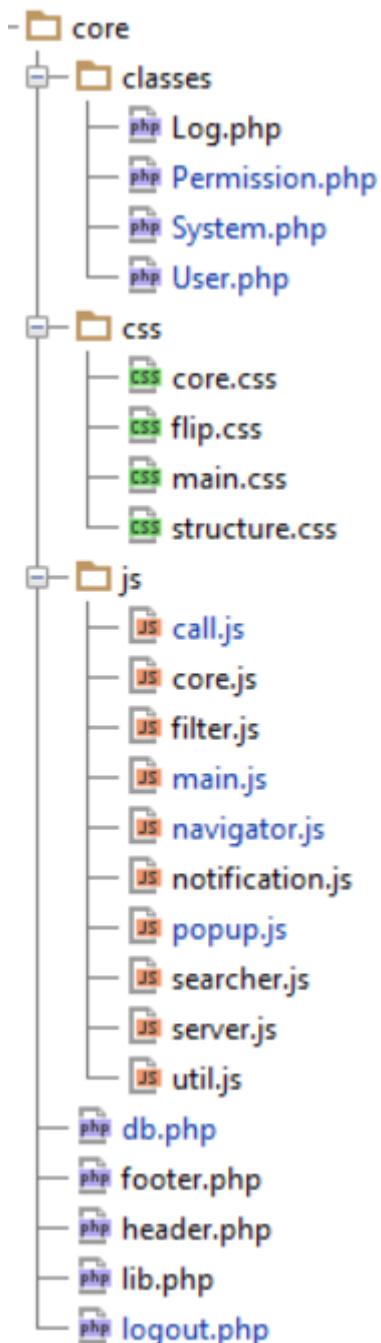


Ilustración 23. Estructura código 2

La carpeta *core* agrupa todos los elementos que parte principal de la plataforma.

Contiene tres carpetas:

- *classes*:

Están las cuatro clases principales, *Log* (encargada de registrar toda acción en la plataforma), *Permision* (control de permisos sobre las distintas herramientas), *System* (agrupa las funciones críticas de autentificación) y *User* que es la clase principal de los usuarios.

- *css*:

Contiene los ficheros de *css* con elementos comunes en todas las vistas.

- *js*:

Funciones u objetos útiles en cualquier lugar de la plataforma.

- Ficheros:

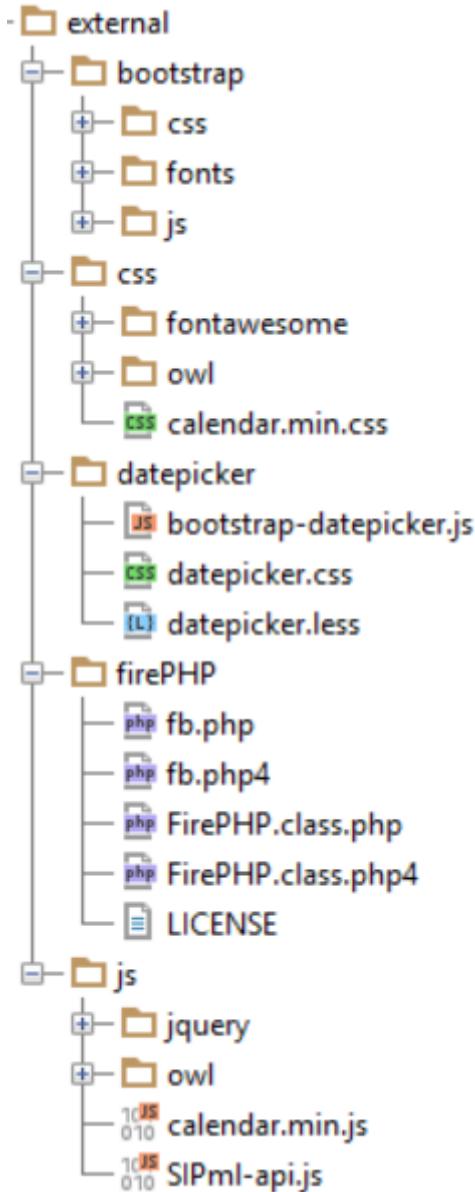
db.php: todas las funciones de ejecución contra la base de datos.

footer.php: contenedor común a todas las vistas con la parte inferior de la página.

header.php: contenedor superior común a todas las vistas con la parte superior de la página (etiqueta `<head>`).

lib.php: funciones de *php* útiles en cualquier lugar del código.

logout.php: fichero con la utilidad de deslogear al usuario.



La carpeta *external* agrupa todos los ficheros que correspondan a librerías, frameworks o plugins no creados por el alumno. Donde tenemos:

Bootstrap: Framework visto anteriormente.

Fontawesome: Fuente con los iconos usados.

Owl: Plugin visto anteriormente.

Calendar: plugin para la visualización de un calendario

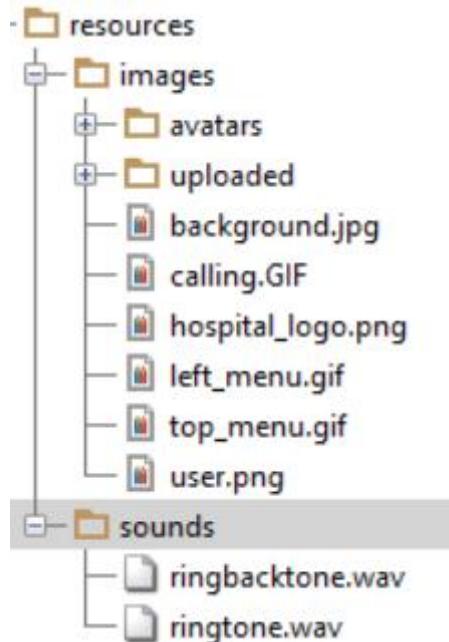
datepicker: Plugin para completar campos de formato fecha

firePHP: Extensión para depurar código

jQuery: Biblioteca de JavaScript

SIPml: Api de uso del protocolo SIP para videollamadas

Ilustración 24. Estructura código 3



La carpeta *resources* agrupa todos los elementos multimedia usados en la plataforma:

images: contiene todas las imágenes, ya sean fijas o subidas

sounds: Sonidos útiles para las videollamadas

Ilustración 25. Estructura código 4

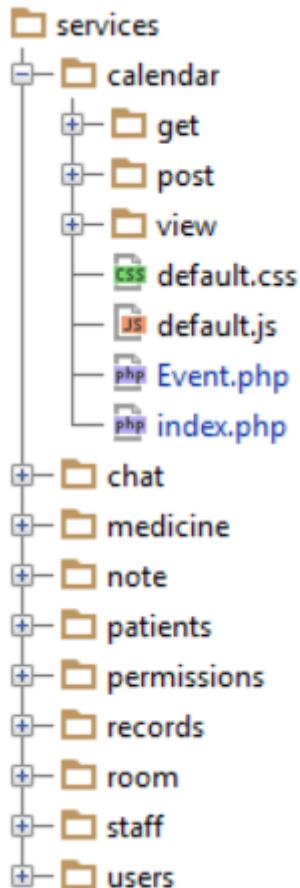


Ilustración 26. Estructura código 5

La carpeta *services* agrupa las distintas aplicaciones de la plataforma , la estructura de cada una es como la mostrada de calendar donde tenemos:

get: todas las peticiones Ajax de tipo get están en esta carpeta, ya puede ser para obtener citas, pacientes, grupos del chat, etc... Siempre devuelve un Json con el estado de la respuesta (success true o false) y la info de la solicitud

post: todos los scripts que implican acciones y son invocados como post están en esta carpeta. Como por ejemplo guardar un mensaje, borrar un protocolo, etc... Al igual que el get devuelve un Json con el mismo formato

view: contiene todos los elementos relativos a vistas

Classes: las clases específicas que necesita cada aplicación están dentro de su carpeta

Index.php: Todas las herramientas tienen su index que muestra la pantalla inicial de cada aplicación

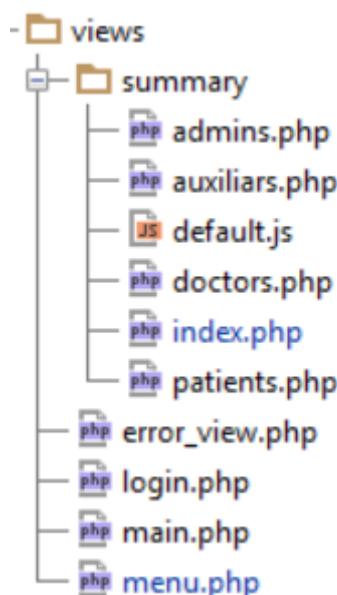


Ilustración 27. Estructura código 6

La carpeta *views* agrupa vistas genéricas de la aplicación.

Summary: contiene las vistas principales para cada rol, así como el index que se encarga de incluir la adecuada, el fichero javascript tiene las funcionalidades de dicha portada

error_view.php: vista usada cuando el usuario entra en una vista inexistente o que no tiene permisos para acceder

login.php: formulario con los campos para logearse en la aplicación

main.php: contiene la estructura del body de la aplicación.

menu.php: contiene el menú izquierdo de la aplicación.

4.1.2 Desarrollo de las principales funcionalidades

De los casos de uso que se han visto anteriormente se han elegido dos para mostrar cómo es su funcionamiento.

4.1.2.1 Creación perfil de un rol

A continuación se describe el proceso necesario para la creación de un rol dentro de la aplicación.

En primer lugar se debe entrar en la herramienta de “Gestión de Permisos”, al pulsar en un enlace del menú para entrar en una herramienta se produce la siguiente secuencia:

1. Se guarda el histórico de navegación en el navegador con:

```
history.pushState(view, "Tool " + view, "?v=" + view);
```

Ilustración 28. Ejemplo código 1

2. Se solicita al servidor la portada de la herramienta con la función dentro del objeto _Server de JavaScript:

```
get_view : function(url, data, on_success, on_error){
    $.get(url, data)
        .done(function(html) {
            if(on_success && typeof(on_success) == "function")
                on_success(html);
        })
        .fail(function() {
            if(on_error && typeof(on_error) == "function")
                on_error();
        })
},
```

Ilustración 29. Ejemplo código 2

Pasándole la dirección de la portada de la herramienta.

3. Se comprueba en el servidor si el usuario tiene permisos suficientes como para ver la herramienta, y en caso de que no los tenga se le presenta una página de error para informar al usuario que está entrando en una zona que no tiene acceso.

```
if(!Permission::can_view(PERMISSION_ROLES)) {
    require_once($CONFIG->dir . 'views/error_view.php');
    die();
}
```

Ilustración 30. Ejemplo código 3

4. Una vez dentro si el usuario tiene permisos para crear nuevos roles tendrá habilitado un botón para realizar dicha acción.

5. Cuando el usuario pulse en el botón se le abrirá un “popup” donde se insertara la vista de creación de perfiles. Esto será realizado de la siguiente manera:

```
var popup = new Popup();
popup.setTitle("Crear permisos:");
popup.setModal(true);
```

Ilustración 31. Ejemplo código 4

6. Las vistas internas a una herramienta son solicitadas al servidor una única vez, son templates (de jquery template) que se rellenan con datos de variables que en este caso se solicitan al servidor cada vez que se abre esta misma vista. Esto agiliza la carga de las vistas y si los datos a llenar en el template están ya en local no es necesario solicitar nada al servidor para mostrar la vista.

```
_Permissions._get.perms(false, function(perms_data) {
    _Server.get_tmpl("services/permissions/view/edit_permission.php", function(tmpl) {
```

Ilustración 32. Ejemplo código 5

7. Una vez llenados los datos del perfil de permisos para el rol se procede a enviarlos al servidor para guardarlos, el proceso una vez más se realiza por ajax.

```
_Server.post_data('services/permissions/post/save_perm.php', data, function(response) {
    alert('success', 'Guardado correctamente');
    popup.close();
    if(on_create && typeof(on_create) == "function")
        on_create(response);
}, function(error) {
    alert('error', error);
})
```

Ilustración 33. Ejemplo código 6

Se envía una petición post con un Json con la información a guardar y se espera a la respuesta, tanto en caso de error como de éxito se notifica al usuario con un mensaje.

8. El servidor recoge la información enviada, que en primer lugar comprueba si el usuario tiene permisos para poder realizar la acción:

```
if(!Permission::can_edit(PERMISSION_ROLES)) {
    echo json_encode(array(
        "success" => false,
        "info"     => 'No tienes permisos'
    ));
    die();
}
```

Ilustración 34. Ejemplo código 7

9. Una vez recogida la información en el servidor se crea el objeto, se guarda en la base de datos y se devuelve un Json con el resultado.

```

$Permission = new Permission($id);
$Permission->setName($name);
$Permission->setRol($rol);
$Permission->setCreator($USER->id);
$Permission->setIndividual(0);
$Permission->setValues($values);

if($Permission->save()){
    echo json_encode(array(
        "success" => true,
        "info"     => $Permission
    ));
} else{
    echo json_encode(array(
        "success" => false,
        "info"     => 'Error al guardar'
    ));
}

```

Ilustración 35. Ejemplo código 8

4.2.2.1 Consulta a través de videollamada

A continuación se describe el desarrollo de una videollamada en la consulta entre un paciente y el profesional sanitario. Se parte de una cita ya asignada entre los dos usuarios.

1. Cuando el usuario entra en la aplicación se envía una solicitud de registro al servidor SIP. Esto se hace a través de un socket, con la tecnología de WebSocket que proporciona un canal de comunicación bidireccional en un único socket TCP. Gracias a la API de SIPml5 esta acción es muy sencilla:

```

_Sip.oSipStack = new SIPml.Stack({
    realm          : _User._sip.url,
    impi           : _User._sip.name,
    impu           : _User._sip.id,
    password       : _User._sip.pass,
    display_name   : _User.name,
    websocket_proxy_url : null,
    outbound_proxy_url : null,
    ice_servers    : null,
    enable_rtcweb_breaker : false,
    events_listener : { events: '*', listener: _Sip.onSipEventStack },
    enable_early_ims : true,
    enable_media_stream_cache : false,
    bandwidth       : null,
    video_size      : null,
    sip_headers: [
        { name: 'User-Agent', value: 'IM-client/OMA1.0 sipML5-v1.2014.03.10' },
        { name: 'Organization', value: 'Doubango Telecom' }
    ]
});

```

Ilustración 36. Ejemplo código 9

2. Cada usuario está identificado por un identificador privado (impi) que será el necesario para que un usuario se comunique con otro, luego está la identidad pública (impu) y su contraseña (password) necesarias para conectarse al servidor (realm). Esto abre el siguiente WebSocket.

The screenshot shows the Network tab in the developer tools of Google Chrome. A request to `ws://ns313841.ovh.net:12060/` is selected. The Headers tab is active, displaying the following request headers:

- Request URL: `ws://ns313841.ovh.net:12060/`
- Request Method: GET
- Status Code: 101 Switching Protocols
- Request Headers CAUTION: Provisional headers are shown.**
- Cache-Control: no-cache
- Connection: Upgrade
- Host: ns313841.ovh.net:12060
- Origin: `http://localhost`
- Pragma: no-cache
- Sec-WebSocket-Extensions: `permessage-deflate; client_max_window_bits, x-webkit-deflate-frame`
- Sec-WebSocket-Key: `PA0GwubS4xCyNGE0zuRPHA==`
- Sec-WebSocket-Protocol: `sip`
- Sec-WebSocket-Version: 13
- Upgrade: websocket
- User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome

Below the request headers, the response headers are listed:

- Connection: Upgrade
- Content-Length: 0
- Sec-WebSocket-Accept: `BNa8z1CkTQibK2ZIfs1514XW8tA=`
- Sec-WebSocket-Protocol: `sip`
- Sec-WebSocket-Version: 13
- Upgrade: websocket

Ilustración 37. Ejemplo código 10

Headers	Frames
Data	
Binary Frame (Opcode 2)	804 01:04:01
REGISTER sip:sip2sip.info SIP/2.0 Via: SIP/2.0/WS df7jal23ls0d.inva...	811 01:04:01

Ilustración 38. Ejemplo código 11

- Como se puede observar desde la consola de Google Chrome tenemos un Websocket dirigido a un host (que en este caso es un servidor gratuito SIP), donde el protocolo utilizado es SIP y se está enviado una solicitud de registro.
- Una vez registrados ya podemos utilizar todas las características que nos ofrece esta API: videollamadas, mensajería instantánea, presencia, compartir pantalla, etc...
- Cuando el profesional sanitario y el paciente estén conectados a la aplicación se puede proceder a la consulta.
- El profesional sanitario deberá entrar a su consulta donde podrá visualizar las citas que tiene para ese día, para ello en el servidor se consultara a la base de datos las citas en las que está involucrado ese usuario para el día actual:

```
SELECT * FROM event WHERE doctor = :p1 AND (UNIX_TIMESTAMP(`start`) >= UNIX_TIMESTAMP(:p2))
                                         AND (UNIX_TIMESTAMP(`start`) <= UNIX_TIMESTAMP(:p3))
                                         ORDER BY start ASC
array(
  [':p1'] => 2
  [':p2'] => '2014-06-18 00:00:00'
  [':p3'] => '2014-06-18 23:59:59'
)
```

Ilustración 39. Ejemplo código 12

7. La sala virtual creada para la cita únicamente podrán entrar el profesional sanitario asignado y el paciente una vez haya sido llamado. La variable de sesión sirve para controlar el acceso.

```
if ($Event = new Event($event_id)) {
    ...
    if ($USER->rol == ROL_USER && $Event->getUser() == $USER->id) {
        ...
    } else if ($USER->rol == ROL_DOCTOR && $Event->getDoctor() == $USER->id) {
        ...
    }
}
```

Ilustración 40. Ejemplo código 13

8. Una vez en la sala virtual el profesional sanitario dispone de la opción de llamar al paciente, al pulsarlo se solicitará al servidor el id de SIP necesario para realizar la conexión con el paciente y se registrara que se ha realizado la llamada. Y con la api de SIPml5 se realizará la llamada

```
$Patient = new Patient($Event->getUser());

$Call = new Call();
$Call->setCaller      ($USER->id);
$Call->setReceptor    ($Patient->getId());

if ($Call->save()) {
    echo json_encode(array(
        'success' => true,
        'info'     => array(
            'call_id' => $Call->getId(),
            'sip_id'  => $Patient->getSip_id()
        )
    ));
    die();
}
```

Ilustración 41. Ejemplo código 14

```
sipCall : function(sip_name) {
    var s_type = "call-audiovideo";
    _Sip.oSipSessionCall = _Sip.oSipStack.newSession(s_type, _Sip.oConfigCall);

    if (_Sip.oSipSessionCall.call(sip_name) != 0) {
        _Sip.oSipSessionCall = null;
        alert('error', 'Error al inicial la conexión');
        return false;
    }
},
```

Ilustración 42. Ejemplo código 15

9. Si consultamos de nuevo el WebSocket que comentamos en puntos anteriores ahora tendremos la petición de conexión con el usuario al que estamos llamando

Headers	Frames		
Data		Length	Time
ACK sip:denispfg2@sip2sip.info SIP/2.0 Via: SIP/2.0/WS df7jal23ls0d.inv...		362	01:58:49
Binary Frame (Opcode 2)		599	01:58:49
Binary Frame (Opcode 2)		493	01:58:49
Binary Frame (Opcode 2)		487	01:58:49
INVITE sip:denispfg2@sip2sip.info SIP/2.0 Via: SIP/2.0/WS df7jal23ls0d.i...		4048	01:58:49
ACK sip:denispfg2@sip2sip.info SIP/2.0 Via: SIP/2.0/WS df7jal23ls0d.inv...		389	01:58:49
Binary Frame (Opcode 2)		662	01:58:49
INVITE sip:denispfg2@sip2sip.info SIP/2.0 Via: SIP/2.0/WS df7jal23ls0d.i...		3838	01:58:49

Ilustración 43. Ejemplo código 16

10. Una vez acepte el paciente la videollamada se conectarán y comenzará el flujo de video, hasta que uno de los dos la finalice, tras esto se registrará en el sistema su duración.

4.2.2.2 Solución ante vulnerabilidades

Se han evitado las principales vulnerabilidades que una aplicación web puede presentar. A continuación se describe cada caso y cuál ha sido la solución implementada:

- Sql Injection:

Es una técnica que permite la ejecución de queries no deseadas realizadas por el usuario, lo que permite la consulta de datos, así como su modificación o borrado.

Para evitar esta vulnerabilidad toda comunicación con la base de datos se realiza utilizando PDO (PHP Data Objects) que es una extensión de PHP que define una interfaz de acceso a la base de datos donde toda query se ejecuta mediante sentencias preparadas que los parámetros se introducen a posteriori evitando así introducir valores no deseados en los parámetros:

```
$query = $db_connection->prepare($sql);

foreach($params as $i => $param) {
    $query->bindValue($i, $param, PDO::PARAM_STR);
}

$query->execute();
```

Ilustración 44. Protección frente a Sql injection

- XSS

Un ataque XSS (Cross Site Scripting) consiste en que el atacante ingresa código HTML o javascript en los formularios y si estos no están correctamente filtrados mostrarán este código en la página alterando el contenido original.

Para evitar esta vulnerabilidad todo parámetro que envie el usuario al servidor es limpiado utilizando la función strip_tags de php que elimina todas las etiquetas html:

```
function get_param($name, $default = false) {
    return strip_tags(isset($_POST[$name]) ? $_POST[$name] :
        (isset($_GET[$name]) ? $_GET[$name] : $default));
}
```

Ilustración 45. Protección frente a XSS

Todo parámetro que se recoja en los scripts se realiza con esta función que se encarga de limpiar todos los valores.

- RFI

Un ataque RFI (Remote File Inclusion) permite el enlace de archivos remotos situados en otros servidores, esto permite la ejecución de código directamente en el servidor, que dejaría al descubierto toda la información.

Para evitar esta vulnerabilidad se ha evitado la inclusión de ficheros que no sean rutas definidas directamente por el desarrollador a ficheros conocidos.

- Session Hijacking

Otra posible brecha de seguridad es cuando una persona malintencionada intenta suplantar a otro usuario de cara al servidor. Esto es fácil de realizar si se obtiene el identificador de la cookie del usuario y el atacante realiza peticiones al servidor con dicha cookie. Para evitar esto lo ideal es utilizar una conexión con HTTPS.

Para evitar esta vulnerabilidad se realizan una serie de comprobaciones para identificar correctamente al usuario y evitar que sea suplantado.

En primer lugar al logearse se guarda en la sesión la IP del usuario e identificador del navegador, y por cada solicitud que realice al servidor se comprobara que esta información se mantiene siempre:

```
if($_SESSION['REMOTE_ADDR'] != $_SERVER['REMOTE_ADDR'] ||
$_SESSION['HTTP_USER_AGENT'] != $_SERVER['HTTP_USER_AGENT']){
    session_destroy();
    header('Location: index.php');
    die();
}
```

Ilustración 46. Protección frente a Session Hijacking 1

Otro parámetro a comprobar dentro de la sesión es el tiempo que ha pasado desde la última acción del usuario , si han pasado 5 minutos desde su última acción se regenera la sesión con un identificador nuevo:

```
$now = strtotime('now');
if($_SESSION['last_action'] + 300) < $now
| session_regenerate_id(true);

$_SESSION['last_action'] = $now;
```

Ilustración 47. Protección frente a Session Hijacking 2

Otro elemento de seguridad es bloquear que el identificador de sesión se pueda transmitir por la url, por ello se activa que únicamente se pueda por cookies. Y como última medida se bloquea que la cookie sea accesible desde JavaScript:

```
ini_set('session.use_only_cookies', 1);
ini_set('session.cookie_httponly', 1);
```

Ilustración 48. Protección frente a Session Hijacking 3

4.2 Control de versiones

Como se describió en el apartado 2.4.3.3 el control de versiones se ha realizado mediante Git.

Se ha trabajado en una única rama llamada “master” que ha ido evolucionando linealmente a través de commits según se iban añadiendo funcionalidades o elementos relevantes al código.

Las correcciones de bugs o cambios que han ido surgiendo durante las reuniones entre el tutor y el alumno se han ido introduciendo durante todo el desarrollo.

Como únicamente ha sido el alumno el desarrollador el trabajo con Git ha sido muy básico siendo este el funcionamiento:

1. Modificación o creación en local de ficheros
2. Commit local con un nombre representativo
3. Subida al repositorio del commit

El repositorio está alojado en:

https://bitbucket.org/Denisvs/pfg_hospital_virtual

Y el histórico de commits es el siguiente:

Author	Commit	Message	Date
Denis Vaillo	q1j3yoh	bugs	2014-06-06
Denis Vaillo	tatyn5z	Permisos en vistas	2014-06-06
Denis Vaillo	xmyj4jf	Errores usuarios	2014-06-05
Denis Vaillo	hmhaboa	Correcciones consulta	2014-06-02
Denis Vaillo	m4ibzq3	Proceso consulta	2014-06-01
Denis Vaillo	r956g34	Historiales medicos v3	2014-05-30
Denis Vaillo	vaxxxnw8	Protocolos	2014-05-26
Denis Vaillo	bz79uec	Gestion de analiticas	2014-05-26
Denis Vaillo	w358q1n	Videollamada	2014-05-25
Denis Vaillo	a9aubkd	Log de acciones	2014-05-21
Denis Vaillo	w8r4cii	Sala de espera	2014-05-18
Denis Vaillo	x96qv8	Consulta medica	2014-05-16
Denis Vaillo	5p8qejj	Acciones en historial	2014-05-10
Denis Vaillo	xwf8m1l	Cambio historiales	2014-05-09
Denis Vaillo	zzpzg15	Historiales medicos v2	2014-05-09
Denis Vaillo	fpa4q9x	Portadas por rol	2014-05-05
Denis Vaillo	ejbvq32	Historiales medicos	2014-05-04
Denis Vaillo	np2fknv	Fallos consulta	2014-04-26
Denis Vaillo	6kakedr	Informes consulta	2014-04-26
Denis Vaillo	p0f2hbk	Campos usuario	2014-04-25
Denis Vaillo	y6mw14i	Mejora calendarios	2014-04-25
Denis Vaillo	2xicdyi	Relaciones usuarios	2014-04-21
Denis Vaillo	v3gzhhu	Bugs mensajeria	2014-04-16
Denis Vaillo	mec9sx4	Mensajeria	2014-04-15
Denis Vaillo	z1bcclu	Consulta medica	2014-04-15
Denis Vaillo	oon9iz8	Calendarios	2014-04-12
Denis Vaillo	4xkz1kn	Control permisos	2014-04-09
Denis Vaillo	2ejepgj	Gestion roles	2014-04-06
Denis Vaillo	5e11y47	Bugs medicos	2014-04-06
Denis Vaillo	w3h7621	Gestion pacientes	2014-04-05
Denis Vaillo	as77dfr	Funciones usuario	2014-04-05
Denis Vaillo	r96x6xc	Gestion usuarios	2014-04-04
Denis Vaillo	7brmijo	Elementos core	2014-03-30
Denis Vaillo	2bay577	Acciones movil	2014-03-28
Denis Vaillo	n8gy74v	Estructura pagina	2014-03-26
Denis Vaillo	r0da9op	Pruebas videollamada	2014-03-23
Denis Vaillo	5yx0zu	Manejo sesiones	2014-03-22
Denis Vaillo	vstsokr	Conexion BD	2014-03-21
Denis Vaillo	gy4immg	Portada inicial	2014-03-21
Denis Vaillo	b07fc00	Creacion repositorio	2014-03-21

Ilustración 49. Histórico de commits

Dado este historial de commits se puede ver cuál ha sido el orden de creación de las distintas funcionalidades.

4.3 Interfaz de usuario

El desarrollo de la interfaz de usuario es uno de los objetivos importantes del proyecto, ante la necesidad de adaptarse a cualquier pantalla los elementos no pueden tener tamaños fijos.

Primero se muestra cual es la estructura base en toda la aplicación, como ejemplo hemos cogido la herramienta de historiales médicos:

Nombre	Apellidos	Historial	Edad	Sexo	Altura	Peso
Gloria	Carter Gomez	2222222222054R	55	female	0	0
Maria	Marques Munoz	1654130	39	female	0	0
Oliver	Pata Sanchez	4565454	26	male	178	86
Pedro	Rodriguez Cano	5445300	48	male	0	0
Laura	Sanchez Sanz	36843287	34	female	0	0
	paciente2					
	paciente1					
	paciente3					

Ilustración 50. Ejemplo interfaz web

Nombre	Apellidos	Historial
Gloria	Carter Gomez	2222222222054R
Maria	Marques Munoz	1654130

Juan
Pepito Perez

- █ HISTORIALES MEDICOS
- █ MENSAJERIA
- █ SALA DE ESPERA
- █ MI AGENDA
- █ SALIR

Ord
Hor
Nor

Ilustración 51. Ejemplo móvil 1

Ilustración 52. Ejemplo móvil 2

Las funcionalidades desde todas las visualizaciones con exactamente las mismas, los elementos relevantes de la página se caracterizan y diferencian entre visualizaciones por estos detalles:

- Cabecera:

La cabecera azul de la página se mantiene en todas las vistas, con una pequeña diferencia de visualización.

En dispositivos grandes en la esquina superior izquierda se mostrará la imagen del usuario y su nombre mientras que en dispositivos pequeños aparecerá un botón para acceder al menú de navegación.

- Navegación

Se tienen dos elementos de navegación en la aplicación, el menú izquierdo que sirve para cambiar entre las distintas herramientas, y debajo de la cabecera una navegación interna a la herramienta que se está visualizando.

En todos los dispositivos tendremos la navegación superior siempre visible diferenciándose por el ancho que ocupa. Este menú irá cambiando según la herramienta si tiene una o varias vistas.

Mientras que el menú izquierdo lo tendremos siempre visible en dispositivos grandes y en dispositivos pequeños estará oculto hasta que el usuario lo requiera.

- Contenido

El contenido se presenta siempre en la zona central en tarjetas con fondo blanco sobre el fondo de la página gris, según el tamaño del dispositivo se presentara en una o varias columnas, y la información mostrada será mayor cuanto mayor sea la pantalla. La información que no se presente de forma inicial en la vista siempre se podrá consultar en vistas más específicas al elemento.

4.4 Evaluación de la usabilidad

Tras la finalización de la interfaz se procede a la evaluación de la usabilidad, los puntos a evaluar se definieron en el capítulo 3.6.

Como todavía no está implementado en escenarios reales la evaluación ha sido realizada por el desarrollador, con las limitaciones que esto conlleva.

- Facilidad de aprendizaje:

La evaluación de la facilidad de aprendizaje se ha realizado mediante los principios definidos

- o Predecible:

Se puede afirmar que la aplicación es predecible debido a:

- Cualquier acción que realice el usuario se puede deshacer o modificar, por ejemplo al asignar un profesional sanitario a un paciente dispone de la posibilidad de borrarla.
- Al realizar una acción que implique un cambio sobre la base de datos se notifica al usuario en caso de confirmación u error.
- Para evitar que el usuario introduzca valores erróneos en los campos de fecha y pese a que los campos en html5 de tipo fecha deberían solo admitir un formato se le proporciona al usuario un calendario que permite realizar la elección de la fecha de forma más sencilla.
- En formularios para evitar datos incorrectos se notifica que valores debe corregir para que toda la información sea adecuada.

- o Familiar

Se puede afirmar que la aplicación es familiar debido a:

- Se utilizan iconos representativos en el mundo real para todas las acciones. Ejemplos:



- Colores de botones que representen confirmaciones en color verde, rojo para acciones peligrosas.

- o Consistencia

Se puede afirmar que la aplicación tiene consistencia debido a:

- Look & feel común entre todas las vistas. Los contenedores son comunes en todas las vistas, los estilos de campos de texto , botones y todos los elementos tienen el mismo tamaño y forma, y están situados en la misma posición con respecto a su contenedor

- Los tipos de letra y tamaños son iguales cuando tienen el mismo uso
- Fácil de recordar
 - Se puede afirmar que la aplicación es fácil de recordar debido a:
 - La interacción con la interfaz es siempre la misma, se representan los datos en formato listado y pulsando sobre el elemento se entra más en detalle.
 - La colocación de los elementos de las acciones sobre un elemento es siempre la misma, situado con botones al final.
 - La edición de un elemento se realiza sobre los valores actuales para evitar tener que memorizarlos.
- Flexibilidad
 - Se puede afirmar que la aplicación tiene flexibilidad debido a:
 - Acciones iguales se pueden realizar desde lugares diferentes, algunos ejemplos:
 - Se pueden crear citas tanto desde el gestor de citas como desde el historial médico de un paciente.
 - Se puede añadir una medicación al paciente tanto desde el historial médico como en la consulta con el paciente.
 - Se pueden asignar profesionales sanitarios al paciente o viceversa
 - Buscador y filtrado sobre las listas de elementos.
 - Acciones probables de realizar en vistas que no sean de ella tienen acceso directo. Ejemplo: En la realización de una cita se puede tener acceso a dar citas nuevas al paciente o enviarle mensajes directos.
- Robustez:
 - La evaluación de la robustez se ha realizado mediante los principios definidos
 - Observable:
 - Se puede afirmar que la aplicación es observable debido a:
 - Al editar cualquier tipo de dato inmediatamente se ve reflejado en todo lugar de la aplicación
 - En la edición de campos del historial médico se notifica al usuario los valores modificados, así como a la hora de guardar el número de registros que se van a modificar.
 - Capacidad de recuperación:
 - Se puede afirmar que hay capacidad de recuperación debido a:

- Si se introduce un valor no deseado por el usuario se tiene la capacidad de modificarlo a posteriori.
- Al llenar valores del usuario si alguno es incorrecto se notifica y se ofrece al usuario la modificación del mismo.
- Realimentación en caso de error
 - Se puede afirmar que hay realimentación en caso de error debido a:
 - Si se produce un error en el servidor por una acción producida por el usuario se notifica con una alerta que el usuario debe leer y aceptar para continuar
 - Ante valores erróneos en un formulario se notifica si hay algún fallo antes de ser enviado al servidor.
- Ajuste de tareas
 - Se puede afirmar que se proporciona ajuste de tareas debido a:
 - Se permite al usuario que la creación de pacientes sea independiente de asignarle profesionales médicos y pueda realizarlo más tarde
 - En una cita se permite llenar el informe, modificar protocolos, medicación o analíticas en el orden deseado.

4.5 Evaluación de la accesibilidad

Tras la finalización de la interfaz se procede a la evaluación de la accesibilidad, las pautas a evaluar se definieron en el capítulo X.

- Perceptible

Se puede afirmar que la aplicación cumple la pauta perceptible debido a que sigue los principios de:

- Alternativas textuales

Las pocas imágenes que aparecen en la plataforma contienen texto alternativo al igual que todos los iconos.

- Medios dependientes del tiempo:

No existen elementos multimedia pregrabados, únicamente se podría aplicar a las videollamadas que depende del uso del usuario, no podrían existir subtítulos pero las conversaciones podrían tener diferentes alternativas como el lenguaje de signos o conversación mediante mensajería.

- Adaptable:

La estructura de la página permite un orden de navegación lógico e intuitivo, los campos de texto están etiquetados superiormente por un título descriptivo adaptado al orden de lectura.

- Distinguible

El color no se utiliza como elemento determinante para transmitir información, en algunos casos se utiliza para mejorarla. No existe ningún audio automático ni imágenes que representen texto

- Operable

Se puede afirmar que la aplicación cumple la pauta de operable dado que sigue los principios de:

- Accesibilidad mediante teclado

Todo elemento de la interfaz es accesible mediante teclado y la entrada de datos no depende de entradas externas al teclado. No existe ningún bloqueo del foco del teclado.

- Suficiente tiempo

No existe ningún elemento que tenga un tiempo impuesto por el contenido. Ni información que tenga desplazamiento automáticamente.

- Convulsiones

No existe ningún elemento que produzca destellos

- Navegable

Todas las páginas tienen títulos representativos de la vista actual, los botones y enlaces tienen un tamaño destacado.

- Comprendible

Se puede afirmar que la aplicación cumple la pauta de comprendible dado que sigue los principios de:

- Legibilidad

El texto proporcionado por la página es comprendible por cualquier usuario, su tamaño permite ser leído y los contenidos más técnicos solo son accesibles por usuarios de alto conocimiento en la materia.

- Predecible

Cuando un elemento recibe el foco no supone ningún cambio relevante en la página y los cambios en la interfaz solo son producidos por acciones de los usuarios. Los mecanismos de navegación están distribuidos siempre en el mismo orden

- Asistencia en la introducción de datos

Los campos que requieran ser rellenados por usuarios se identifican y proporciona realimentación al usuario en caso de error

- Robusta

Se puede afirmar que la aplicación cumple la pauta de comprendible dado que sigue el principio de:

- Compatible

El lenguaje de marcas utilizado contiene todas las etiquetas correctamente cerradas y no existen elementos con identificadores repetidos o etiquetas no válidas.

5. Conclusiones y futuros trabajos

Después del desarrollo y uso de la aplicación, se han extraído las conclusiones y posibles trabajos futuros para mejorar y continuar con el trabajo realizado.

5.1 Conclusiones

Al inicio del proyecto se definieron unos objetivos para el desarrollo de un servicio de hospital virtual para pacientes con VIH los cuales se han cumplido satisfactoriamente.

El desarrollo de todas las funcionalidades necesarias para el cumplimiento de dichos objetivos ha ido avanzando desde el cumplimiento mínimo hasta alcanzar un nivel aceptable marcado por los requisitos.

Los objetivos se han ido modificando según se producía el desarrollo de la aplicación, para así satisfacer las necesidades del sistema.

Se ha dado importancia a un desarrollo de calidad siguiendo todos los pasos con sumo cuidado para finalmente tener una aplicación usable, fiable, fácil de mantener y con una interfaz sencilla y agradable para los usuarios.

A nivel personal, la realización de este proyecto me ha permitido el aprendizaje de nuevas tecnologías, la mejora de los conocimientos previos aprendidos durante mi formación como Ingeniero de Software y la colaboración con las personas que se han involucrado en llevar adelante este TFG. Un aspecto muy positivo es la libertad en la toma de decisiones que he tenido a diferencia de los trabajos previos realizados durante la carrera.

Por todo ello, considero que he realizado un buen trabajo, basado en el gran apoyo recibido por mi tutor y mi esfuerzo para cumplir los objetivos desde el inicio.

5.2 Trabajos futuros

En el inicio de este trabajo de fin de grado se estudiaron todos los detalles que una plataforma web con el objetivo de dar servicio real a un hospital necesitaba. Ante la gran cantidad de objetivos que se podían presentar se eligió realizar un primer paso construyendo desde cero una plataforma que poco a poco fuese cumpliendo metas y cubriera la mayor cantidad de dichos objetivos.

Por ello, tras la finalización del desarrollo se pueden definir una serie de mejoras y de funcionalidades a desarrollar en un futuro:

5.2.1 Mejoras en funcionalidades existentes

A continuación, se listan posibles mejoras para el estado actual de la aplicación:

- En la gestión de citas, una característica a mejorar sería que se pudiese sincronizar el calendario interno de citas del hospital con externos, como por ejemplo Google Calendar.
- Otra característica a mejorar de la gestión de citas sería poder realizar un cambio automático de las citas de un profesional sanitario a otro, debido a un cambio de personal o baja del mismo.
- Para que los usuarios estén actualizados de todas las novedades entorno a su tratamiento un punto a mejorar sería la notificación de eventos relevantes como analíticas nuevas, asignación de citas, mensajes nuevos, etc... por correo electrónico para que el usuario esté informado en todo momento de las novedades.
- Para la parte de administración de la aplicación una posible mejora sería poder tener una visualización de los logs de actividad que se recogen de los usuarios.
- Para las videollamadas y mensajería se está utilizando un servidor SIP externo, sería recomendable la instalación de un servidor Sip propio.

5.2.2 Nuevas funcionalidades propuestas

A continuación se listan posibles funcionalidades a implementar en el futuro:

- Una característica importante en el tratamiento de un paciente de VIH es tener un control preciso de la adherencia a su tratamiento, por ello la implementación de una nueva funcionalidad para controlar todas las tomas y aviso a los pacientes para evitar olvidos.
- Los profesionales sanitarios no suelen tener horarios fijos de atención a los pacientes, por ello una mejora sería poder gestionar planillas de horarios de forma individual.
- Dentro de la gestión de la medicación en el historial médico del paciente una futura funcionalidad sería disponer de un servicio de creación de recetas. Así el paciente una vez el profesional médico se la asigne podría descargar la receta de dicha medicación.

- Una posible forma de valorar la satisfacción de los pacientes con el servicio de consultas sería implementando unos formularios de satisfacción que llenaría el paciente en la finalización de la consulta.
- Dado que los usuarios de la plataforma pueden preferir visualizar la página en otros idiomas, una posible mejora sería poder tener la aplicación en múltiples lenguajes para que el usuario pueda elegir según su idioma.
- Una forma de que los pacientes estén bien informados sobre el VIH sería tener una apartado donde especialistas en la materia pusieran a disposición del paciente información relevante sobre la enfermedad. Esto podría apoyarse por foros de debate donde los pacientes podrían comentar de forma anónima.

6 Anexos

6.1 Manual de usuario

A continuación se incluye un manual de usuario de la aplicación. Se detallan todas las funcionalidades y los pasos a seguir para su correcta utilización. Dado que la aplicación es compatible con cualquier tamaño de pantalla las capturas que se adjuntan son de diferentes dispositivos.

6.1.1 Inicio

Para acceder a la aplicación tenemos una página de inicio donde disponemos de la opción de logearse y la de recordar contraseña.

Para la primera acción deberá introducir su número de DNI con letra (sin espacios ni guiones), y en el caso que desee recordar la contraseña deberá pulsar sobre el link habilitado para tal donde se le habilitara un formulario para recordar su contraseña a través de su DNI o email asociado.



Ilustración 53. Login

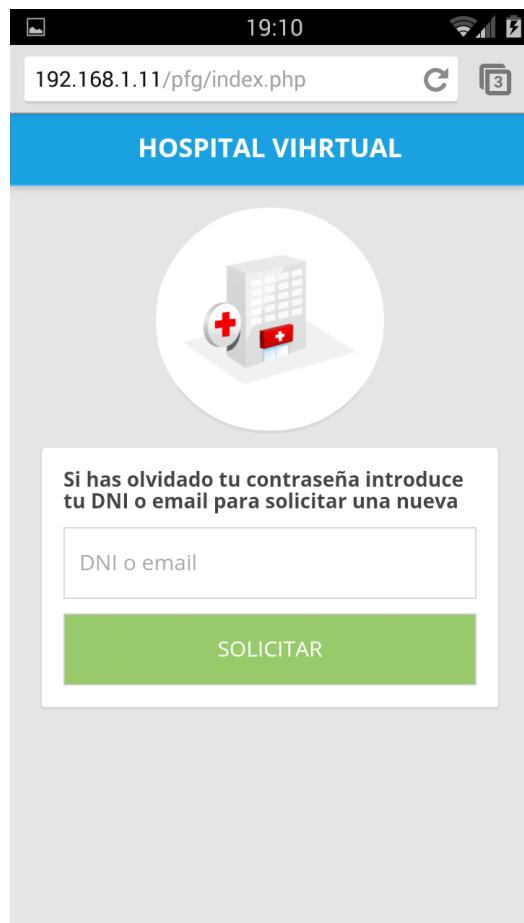


Ilustración 54. Recordar contraseña

En el proceso para recordar contraseña el usuario recibirá un email donde a través de un link podrá establecer una nueva contraseña.

6.1.2 Navegación

Dentro de la aplicación existen dos barras de navegación, el menú izquierdo para la navegación a otras herramientas. Que dependiendo del dispositivo estará oculto o no.



Ilustración 55. Menú dispositivos móviles



Ilustración 56. Menú grande

Y la barra superior en cada herramienta que nos permite cambiar entre las funcionalidades disponibles.



Ilustración 57. Barra superior navegación

En la barra superior de la aplicación está situado en el lado derecho un botón para ir a la página inicial de la aplicación, y en dispositivos móviles en el lado izquierdo está situado el botón para abrir el menú de navegación.



Ilustración 58. Cabecera pagina

6.1.3 Página principal

Una vez logeado en la aplicación el usuario entra en una página inicial donde tendrá acceso al menú de la aplicación, así como a una vista resumen con las principales funcionalidades dependiendo de cada rol. Y todos tienen en común la parte de noticias.

6.1.3.1 Paciente

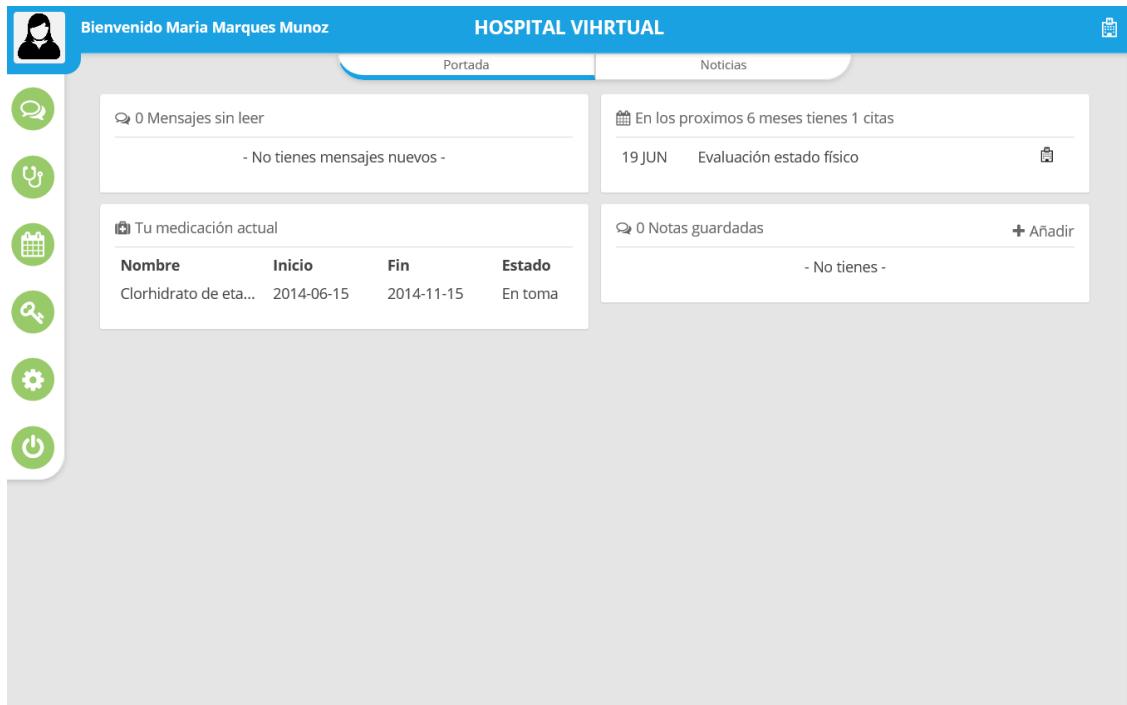


Ilustración 59. Vista resumen paciente

El paciente tendrá acceso a cuatro funcionalidades principales:

- Mensajes sin leer:

Mostrará un listado de mensajes sin leer, al pulsar sobre uno de la lista accederá directamente al chat con dicho usuario.

- Citas:

Mostrará un listado con las citas que tiene asignadas el usuario en los próximos 6 meses, si pulsa sobre una de ellas le llevará a la gestión de citas.

- Medicación:

Mostrará un listado con la medicación activa del usuario

- Notas:

El usuario podrá gestionar ver las notas que tiene guardadas o crear nuevas. Si picha sobre una de ellas se le abrirá la información.

6.1.3.2 Profesional sanitario

Apellidos, Nombre	Nº Historial	Dni
Marques Munoz, Maria	1654130	paciente1
Pata Sanchez, Oliver	4565454	47255641L
Sanchez Sanz, Laura	36843287	paciente2

Ilustración 60. Vista resumen profesional sanitario

En el caso del personal sanitario tendrá disponible el listado de pacientes que tiene asignados, los mensajes que tiene sin leer y las citas para el día actual.

Cada una de estas tres vistas enlaza con su funcionalidad:

- **Tus pacientes:**

Permite la búsqueda del paciente por su nombre, numero de historial o DNI, y al pulsar sobre un paciente de la lista te lleva directamente a su historial médico.

- **Mensajes nuevos:**

Muestra los mensajes que tenemos actualmente sin leer, si pulsamos sobre uno de ellos nos lleva directamente al chat con dicho usuario para poder ver los mensajes que tenemos.

- **Citas de hoy:**

Muestra las citas que tiene el médico para el día actual, si pulsamos sobre alguna de ellas nos lleva a la sala de espera de la consulta del médico.

6.1.3.3 Personal administrativo y administradores

Estos dos roles comparten el mismo tipo de página de inicio, donde podrán ver los mensajes nuevos que tienen.

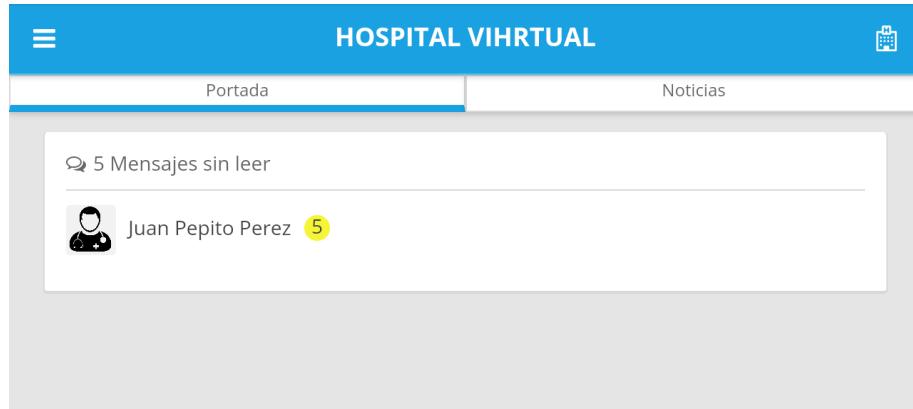


Ilustración 61. Vista resumen otros

6.1.4 Gestión de roles

La gestión de roles de la aplicación permite configurar los permisos que van a tener los usuarios sobre cada una de las herramientas disponibles. Existen cuatro roles en los cuales se pueden ir creando perfiles de dichos roles para ello disponemos de la opción de añadir nuevos o pulsando sobre cualquiera de la lista se permite editarlos:

 A screenshot of the role management interface. It displays four categories in a grid:

- Pacientes**: Includes options for "Por Defecto", "Urgencias", and "A distancia".
- Personal Sanitario**: Includes options for "Por Defecto", "Enfermeros", and "Becarios".
- Personal Administrativo**: Includes options for "Por Defecto", "Citaciones", and "Organizador medicos".
- Administradores**: Includes options for "Por Defecto" and "Admin Usuarios y permisos".

 Each category has a "+ Añadir" button to the right.

Ilustración 62. Gestión de roles

Desde esta herramienta también se permite cambiar el perfil a usuarios o editar el de un usuario de forma individual. Para ello tenemos un listado de todos los usuarios, el cual se puede filtrar por rol o buscar a un usuario en concreto, y pulsando sobre la lista nos permite su edición.

Listado de Usuarios			
Cualquier Rol	Nombre o identificador	Rol	Perfil
Vaillo Sanchez, Denis		Administrador	Por defecto
Pepito Perez, Juan		Sanitario	Por defecto
Perez Lopez, David		Aministrativo	Por defecto
Marques Munoz, Maria		Paciente	Otro

Ilustración 63. Gestión de roles de usuarios

La edición de los perfiles se realiza desde un *popup* que se superpone a la vista actual, donde podemos ir activando cada herramienta y el nivel de permisos sobre ella.

Crear permisos:

Nombre:	Por Defecto	Rol:	Pacientes
Logearse	Si	Perfil del usuario	Editar Suyo
Noticias	Ver	Historiales	Ver
Mensajería	Hablar relacion	Agenda	Ver Agenda
Sala de espera	Entrar	Gestion de usuarios	Deshabilitado
Gestion de permisos	Deshabilitado	Gestion del personal	Deshabilitado
Gestion de pacientes	Deshabilitado		

Guardar

Ilustración 64. Vista edición perfil permisos

Desde esta misma vista se permite borrar el perfil del rol siempre y cuando no sean los permisos por defecto. En el caso de que se borre un rol y existan usuarios que lo tengan asignados se les pasará al perfil por defecto.

6.1.5 Gestión de usuarios

La herramienta de gestión de usuarios nos permite la creación, modificación o borrado de usuarios de la aplicación. Dispone de un listado donde podremos buscar usuarios en concreto y realizar operaciones de modificación o borrado del mismo, así como un botón para crear un nuevo usuario

Todos los usuarios		+ Crear usuario	
Nombre usuario o historial		Buscar	
Apellidos, Nombre		Rol	Identificador
Barbas Tones, Santiago		Personal Sanitario	1000000056451
Bollo Galdes, Tamara		Personal Sanitario	88888520
Carter Gomez, Gloria		Paciente	2222222222054RO 3684526R

Ilustración 65. Listado de usuarios

Tanto la vista de edición como la de creación abren un *popup* donde podremos llenar toda la información del usuario. En el caso de que sea la edición dispondremos de un botón para borrarlo.

Vista usuario

Rol del usuario:

Rol del usuario	Perfil del usuario
-----------------	--------------------

Datos Personales:

Nombre*	Apellidos*	
<input type="text" value="Nombre"/>	<input type="text" value="Apellidos"/>	
DNI*	Sexo*	Fecha de nacimiento*
<input type="text" value="DNI"/>	<input type="text" value="Sexo"/>	<input type="text" value="Fecha de nacimiento"/>

Datos de contacto:

Domicilio	Ciudad	C.P
<input type="text" value="Domicilio"/>	<input type="text" value="Domicilio"/>	<input type="text" value="Codigo Postal"/>
Telefono	Email	
<input type="text" value="Telefono"/>	<input type="text" value="Email"/>	

Datos clínicos:

Grupo sanguíneo*
<input type="text" value="Grupo sanguíneo"/>
Información adicional
<input type="text" value="Informacion adicional"/>

Guardar Cancelar

Ilustración 66. Vista creación usuario

Dependiendo del rol elegido dispondremos de unos campos diferentes a llenar en el apartado de Datos Clínicos. En el caso de usuarios ya creados dispondremos de un botón para generar una nueva contraseña, que creara una contraseña aleatoria y notificara al usuario.

6.1.6 Gestión del personal

La herramienta de gestión del personal nos permite tener separada la gestión de los trabajadores de los usuarios. Se compone de un listado de todo el personal con la posibilidad de filtrar por la rama y/o horario del trabajador, así como la búsqueda por su nombre o identificador.

Apellidos, Nombre
Pepito Perez, Juan
Perez Lopez, David
Lopez Marin, Manuel
Manter Gil, Sergio
Norto Garrido, Pilar

Ilustración 67. Vista gestión del personal

Pulsando sobre un elemento de la lista se abrirá un popup con la información que se puede editar del trabajador, y en el caso de que se trate de que tenga rol de profesional sanitario se listaran los pacientes que tienen asignados, permitiendo asignar nuevos o quitarlos.

The screenshot shows a modal window titled 'Personal Hospital'. The 'Datos Clínicos' section contains the following fields:

- Nombre:** Juan Pepito Perez
- Identificador:** 1000000002
- Rama:** Fisiatria
- Turno:** Noche
- Consulta:** 666
- Oficina:** 41
- Telefono Hospital:** 112

A green 'Guardar' button is located at the bottom right of the form.

Below the form, there is a section titled 'Sus pacientes' with a search bar labeled 'Añade pacientes (nombre o historial)'. A table lists three patients:

Nombre	NºHistorial	
Laura Sanchez Sanz	36843287	
Maria Marques Munoz	1654130	
Oliver Pata Sanchez	4565454	

Ilustración 68. Vista edición personal

Los datos clínicos podrán ser modificados todos menos el nombre del usuario, el identificador debe ser único entre todos los trabajadores. Esta información se guardara pulsando el botón inferior.

Se podrán asignar pacientes nuevos con el buscador, y borrar los actuales presionando el icono de la basura situado a la derecha de cada elemento de la lista. Estos cambios se guardarán automáticamente.

6.1.7 Gestión de pacientes

La gestión de pacientes es una herramienta que nos permite acceder a información clínica del paciente, así como los médicos que tiene asignados.

Se compone de un listado con un buscador de todos los pacientes del hospital que pulsando sobre uno de ellos nos mostrará en un *popup* la información clínica básica para poder editarla , así como los médicos que permite la asignación de nuevos o quitar los actuales.

Apellidos, Nombre	Nº Historial	Dni	Telefono
Carter Gomez, Gloria	22222222222054RO	3684526R	63300245
Marques Munoz, Maria	1654130	paciente1	650210074
Pata Sanchez, Oliver	4565454	47255641L	
Rodriguez Cano, Pedro	5445300	paciente3	666222547
Sanchez Sanz, Laura	36843287	paciente2	

Ilustración 69. Vista listado pacientes

Los datos clínicos están compuestos por el nombre, numero de historial (que debe ser único), altura (cm) y peso del paciente (kg); los cuales, se guardan con el botón de la parte inferior derecha.

Para la asignación de personal sanitario se dispone de un buscador que se puede filtrar por su rama. Tanto la asignación como el borrado se guardan automáticamente,

Paciente

Datos Clínicos:

Nombre:	Número de historial:
Gloria Carter Gomez	22222222222054RO
Altura:	Peso:
165	62

Personal sanitario asignado

Cualquier Rama	Añade personal (nombre o identificador)																
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Apellidos, Nombre</th> <th>Identificador</th> <th>Horario</th> <th>Rama</th> </tr> </thead> <tbody> <tr> <td>Tamara Bollo Galdes</td> <td>88888520</td> <td>Mañana Tarde</td> <td>Pediatria</td> </tr> <tr> <td>Manuel Lopez Marin</td> <td>2147483647</td> <td>Mañana Noche</td> <td>Neumologia</td> </tr> <tr> <td>Sergio Manter Gil</td> <td>5224751210</td> <td>Tarde Noche</td> <td>Oncologia</td> </tr> </tbody> </table>		Apellidos, Nombre	Identificador	Horario	Rama	Tamara Bollo Galdes	88888520	Mañana Tarde	Pediatria	Manuel Lopez Marin	2147483647	Mañana Noche	Neumologia	Sergio Manter Gil	5224751210	Tarde Noche	Oncologia
Apellidos, Nombre	Identificador	Horario	Rama														
Tamara Bollo Galdes	88888520	Mañana Tarde	Pediatria														
Manuel Lopez Marin	2147483647	Mañana Noche	Neumologia														
Sergio Manter Gil	5224751210	Tarde Noche	Oncologia														

Ilustración 70. Vista edición pacientes

6.1.8 Ajustes

Para la cuenta del usuario se permite realizar dos modificaciones, el email asociado y la contraseña. Por ello, entrando en la herramienta ajustes tenemos

The screenshot shows a user profile page with the following sections:

- Datos Usuario:** Displays basic information: Nombre: Denis, Apellidos: Vaillo Sanchez, Dni: admin, Dirección: C/Sin nombre 25 (Mostoles, 28938), Teléfono: 656456292.
- Tu email:** Allows changing the email. Fields: Tu Email (denisvaillo@gmail.es), Nuevo email (ejemplo@dominio.es), Repite nuevo Email (ejemplo@dominio.es). A green "Guardar" button is present.
- Tu contraseña:** Allows changing the password. Fields: Nueva contraseña (minimum 6 characters), Repite nueva contraseña (repite contraseña), Confirma tu contraseña actual (*****). A green "Guardar" button is present.

Ilustración 71. Vista ajustes cuenta

Cada usuario puede revisar si sus datos básicos son correctos, así como dos funcionalidades:

- Cambiar email:

Se solicita al usuario introducir el email y la confirmación para evitar fallos. Al pulsar guardar se enviará un email a la nueva cuenta para verificar que el usuario tiene acceso a la misma.

- Cambiar contraseña:

El usuario deberá introducir su nueva contraseña, confirmarla y por ultimo introducir la contraseña actual. Tras esto si pulsa guardar y todos los datos son correctos tendrá la contraseña elegida.

6.1.9 Gestión de citas

La herramienta de gestión de citas permite la consulta, creación, modificación o anulación de citas entre un profesional médico y un paciente.

El personal administrativo tendrá el control sobre todas las agendas de los pacientes o profesionales sanitarios, por ello cuando un usuario con este rol o el de administrador tendrán un buscador de agendas de usuarios donde introduciendo su nombre o identificador podrá consultar su agenda.

La visualización de las agendas se compone de un botón para solicitar una cita, un listado de las citas futuras, un botón para ver las citas pasadas y un calendario donde poder visualizar cuando tiene las citas de manera más sencilla.

Ilustración 73. Solicitud de cita

Ilustración 72. Vista listado citas

Al pulsar sobre solicitar una cita se abrirá un *popup* donde se elegirá con que profesional sanitario o paciente tendrá la cita (según para quien se pida la cita), cuando, el tipo (online o presencial) y un campo libre para exponer el motivo.

Para elegir cuando es la cita tenemos tres opciones:

- Manual:

Se inserta un día y nos muestra cuales son las citas disponibles para ese día.

- Próxima libre:

Aparece un desplegable con las próximas citas libres.

- Próxima libre a partir de un día:

Se elige un día y aparecen las citas más próximas a partir de ese día.

La edición y eliminación de citas se realiza pulsando sobre uno de los eventos, el cual, abrirá un *popup* igual al anterior dando la opción de guardar los cambios o anular la cita.

6.1.10 Mensajería

La herramienta de mensajería permite el envío de mensajes directos entre usuarios. Se compone de dos vistas, una con las conversaciones abiertas y otra con la lista de contactos.

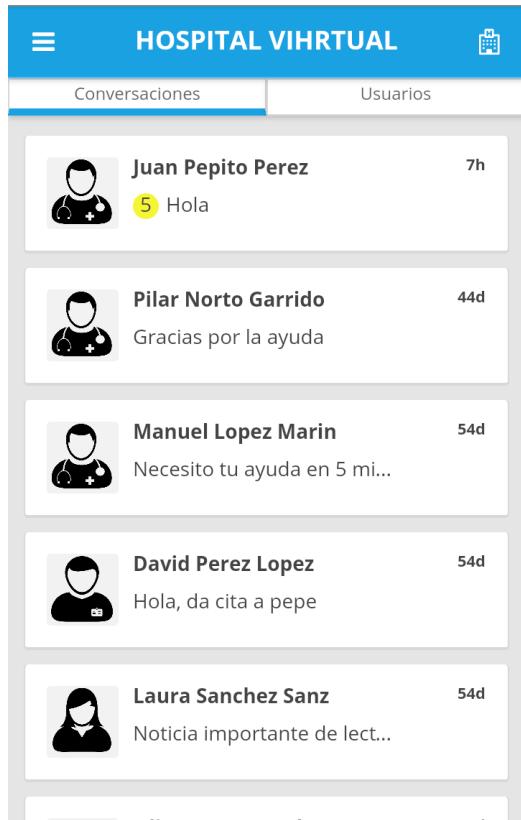


Ilustración 74. Listado de conversaciones

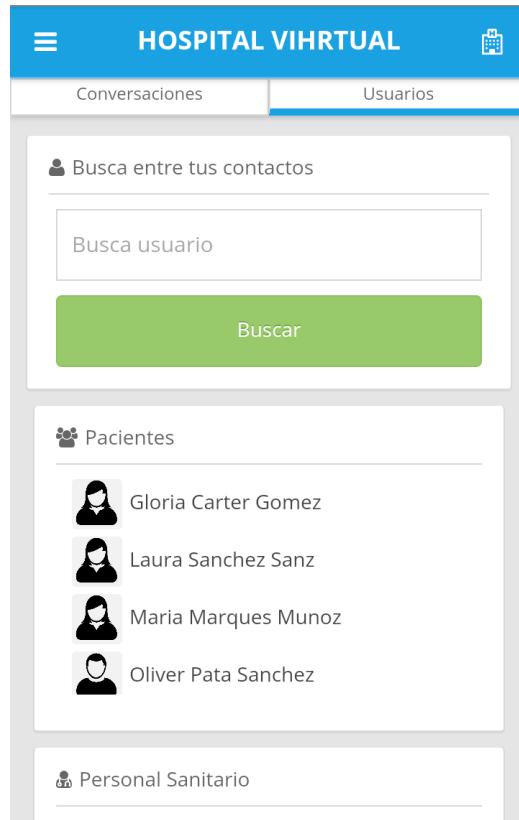


Ilustración 75. Lista de contactos

La primera recopila el listado de conversaciones con el último mensaje y su fecha, en el caso de que tengamos nuevos mensajes se notificara con un número a la izquierda. El orden de la lista es por fecha.

La segunda nos muestra los contactos con los que podemos comunicarnos, se compone de un buscador por nombre o identificador, y un listado por cada rol con los usuarios del mismo que podemos iniciar una conversación.

Al pulsar sobre una de las conversaciones recientes o sobre un usuario del listado de contactos nos llevará a la vista de conversación, donde tendremos todos los mensajes con dicho usuario. Se notificarán los mensajes nuevos con una separación entre los viejos, y se permitirá enviar nuevos mensajes.

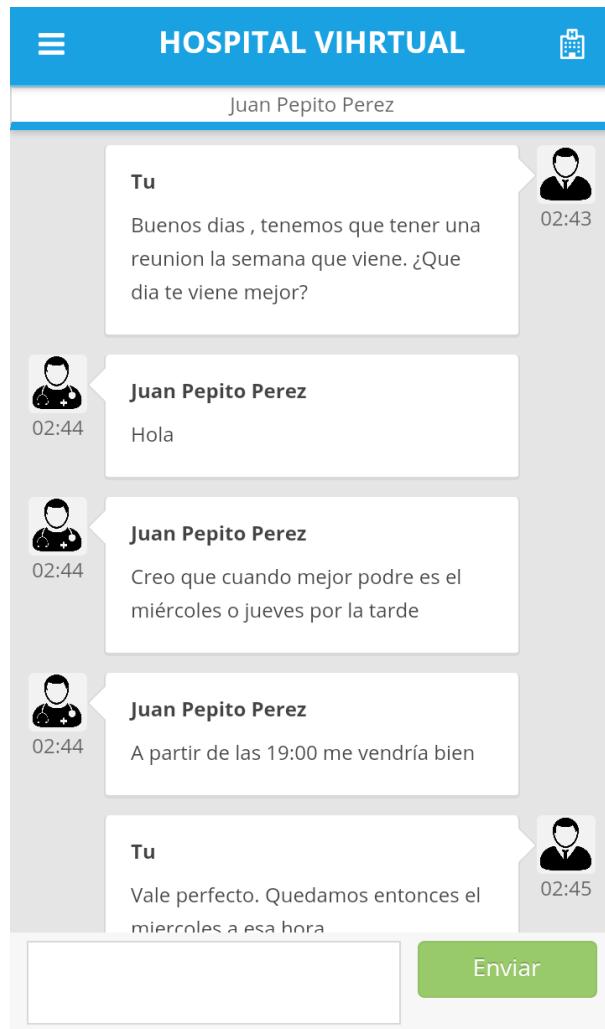


Ilustración 76. Vista de conversación

6.1.11 Historiales médicos

La recopilación de toda la información clínica de los pacientes está disponible en esta herramienta. Al acceder a ella se muestra un listado de todos los historiales médicos a los que el usuario tiene acceso, sobre los cuales podrá aplicar filtros, ordenar o realizar búsquedas por nombre, número de historial o dni.

Historiales Medicos						
Ordenado	Hombres y Mujeres	Cualquier edad	Nombre, historial o dni			
	Nombre: Gloria Apellidos: Carter Gomez Historial: 2222222222054RO DNI: 3684526R	Edad: 55 Sexo: female Altura: 0 Peso: 0		Nombre: Maria Apellidos: Marques Munoz Historial: 1654130 DNI: paciente1	Edad: 39 Sexo: female Altura: 0 Peso: 0	
	Nombre: Oliver Apellidos: Pata Sanchez Historial: 4565454 DNI: 47255641L	Edad: 26 Sexo: male Altura: 178 Peso: 86		Nombre: Pedro Apellidos: Rodriguez Cano Historial: 5445300 DNI: paciente3	Edad: 48 Sexo: male Altura: 0 Peso: 0	
	Nombre: Laura Apellidos: Sanchez Sanz Historial: 36843287 DNI: paciente2	Edad: 34 Sexo: female Altura: 0 Peso: 0				

Ilustración 77. Vista listado de historiales

Si se pulsa sobre cualquiera de los historiales accedemos a la vista completa del mismo, donde se mostraran todos los detalles del historial.

The screenshot displays a detailed medical history interface. On the left, under 'Paciente', personal information is listed: Nombre: Oliver, Apellidos: Pata Sanchez, N°Historial: 4565454, DNI: 47255641L, F.Nacimiento: 1988-01-05 (26 años), Sexo: Masculino, Altura: 178 cm, Peso: 86 kg, G.Sanguíneo: A+. Below this are three buttons: Cita, Chat, and Alertas. To the right, under 'Informe evolutivo', two entries are shown: > 2014-05-02 Urgencia revisión médica and > 2014-05-06 Revisión. The main area is divided into sections: 'Analíticas' (with tables for 'Subpoblaciones Linfocitarias' and 'Cargas Virales'), 'Tratamientos' (listing Atazanavir and Clotrimazol), and 'Protocolos' (listing Fase 33, Fase 2, and Inicio Tratamiento). Each section has an 'Editar' button.

Ilustración 78. Vista historial médico

Como se puede observar el historial se compone de 5 secciones:

- Paciente:

Muestra la información básica del paciente, con tres acciones posibles a realizar:

- Dar una cita,
- Abrir el chat con el usuario
- Gestionar las alertas.

Las dos primeras son funcionalidades vistas anteriormente, las alertas permiten establecer mensajes que serán mostrados al usuario que abra el historial de forma que deba verla obligatoriamente. Las alertas pueden ser dirigidas a cualquiera de los cuatro roles. Permitiendo ser borradas o añadir nuevas.

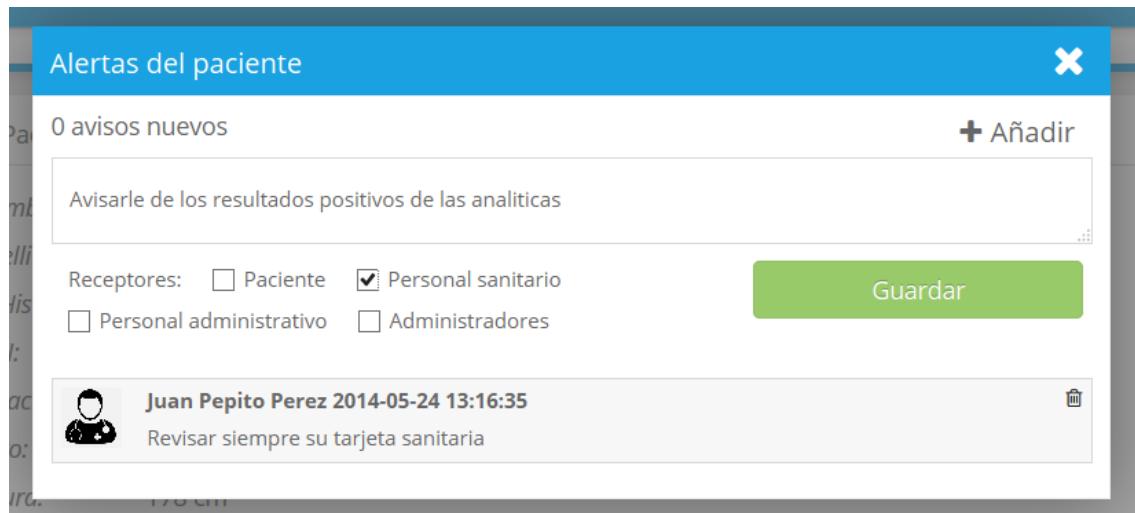


Ilustración 79. Vista gestión alertas del paciente

- Informe evolutivo:

Recopila todos los informes rellenados por profesionales sanitarios en las citas finalizadas con el paciente. Pulsando sobre los títulos de la fecha de la cita nos permite visualizar el informe completo

Ilustración 80. Vista informe evolutivo

- Analíticas:

Todas las analíticas realizadas al paciente serán listadas en orden de más recientes a antiguas. Existen dos tipos de analíticas “Subpoblaciones Linfocitarias” y “Cargas virales”.

Pulsando el botón de editar se nos permite añadir nuevas analíticas, editar las existentes o borrarlas, para ello tenemos la siguiente vista:

<u>Subpoblaciones Linfocitarias</u>						<u>Cargas Virales</u>			
Fecha	Linf	%T4	T4 Abs	%T8	T8 Abs	Fecha	Carga	Dif	Log
2014-05-25 23:1	22	118	22	78	55	2014-03-01 11:0	999	99.3689	23.55
2014-05-13 23:1	0.27	0.59	6.33	6.21	9.26	2014-03-01 10:0	12	44	8745
2014-05-13 23:1	0.334	25	2.66	33	55	2014-03-01 10:0	55	44	11

[Añadir Nueva](#) [Guardar](#) [Añadir Nueva](#) [Guardar](#)

Ilustración 81. Vista edición analíticas

Cualquier modificación sobre el listado será reflejado sobre el campo de texto, y si se intenta volver sin guardar será notificado al usuario.

- Tratamientos

El tratamiento del paciente será gestionado de forma que se permite añadir nuevos tratamientos o suspender los activos, si queremos suspender el tratamiento actual se deberá incluir el motivo de la suspensión.

Nombre	Inicio	Fin	Estado	Dosis
Atazanavir	2014-05-01	2014-05-31	Finalizado	1 cada 24 hora/s
Clotrimazol	2014-05-15	2014-05-17	Finalizado	4 cada 1 dia/s

Ilustración 82. Vista tratamientos

- Protocolos

Los protocolos aplicados al paciente serán listados y a través de la vista de edición se podrán crear nuevos, editar los existentes o borrarlos.

Nombre	Inicio	Fin
Fase inicial	2014-05-31	2014-06-15
Inicio Tratamiento	2014-05-08	2014-05-29

Ilustración 83. Vista protocolos

6.1.12 Sala de espera

La sala de espera consta de dos vistas para la sala inicial antes de entrar a consulta, una para los profesionales médicos y otra para los pacientes.

- Profesionales sanitarios:

Disponen del listado de pacientes citados para el día actual, se muestra la hora y el tipo de cita (presencial u online). Pulsando sobre la cita accederemos a la consulta virtual.

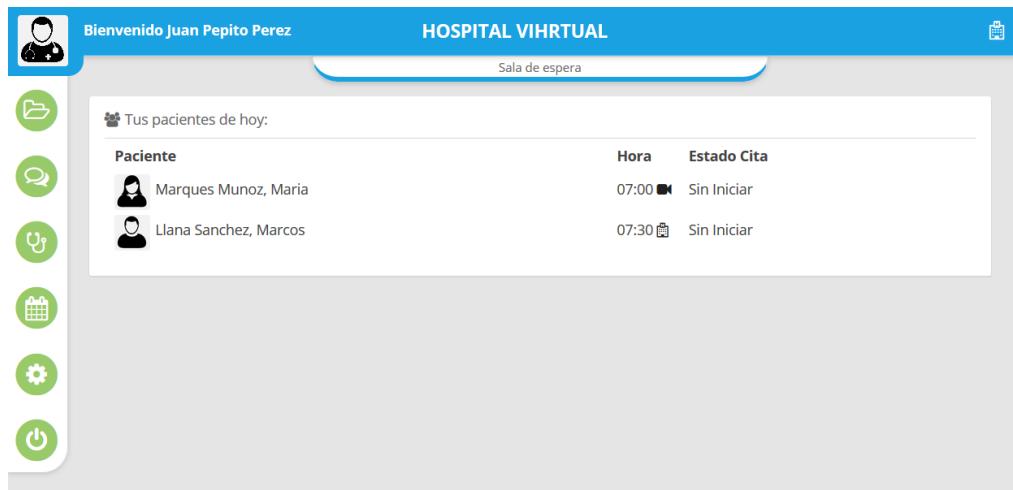


Ilustración 84. Sala espera prof.sanitario

- Pacientes:

En el caso de que tengan cinta ese día dispondrán de la siguiente vista donde podrán esperar a ser llamados.

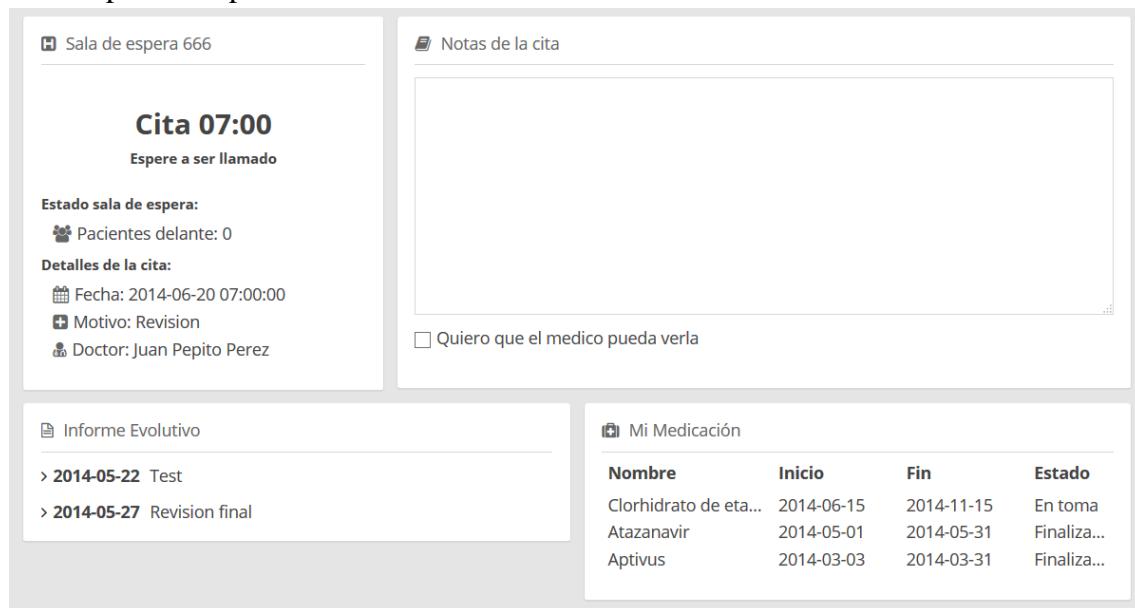


Ilustración 85. Sala espera paciente

Tienen información sobre la cita, acceso directo a su informe evolutivo y al histórico de medicación. La funcionalidad de las notas de la cita permite al usuario guardar un texto que posteriormente en la consulta podrá ver y si

selecciona que lo vea el profesional médico esta será visible para él. Todos los cambios sobre esta nota son guardados automáticamente.

Una vez dentro de la consulta tanto el medico como el paciente visualizaran el historial médico del paciente, añadiéndose la nota del paciente y en el caso del médico tendrá disponible el nuevo informe a llenar.

Para que se empiece a desarrollar la consulta es necesario que el profesional sanitario llame al paciente, por ello si la cita es online le llamara para iniciar la videollamada o si es presencial deberá iniciarla.

Las acciones disponibles según el tipo y estado de la consulta son:

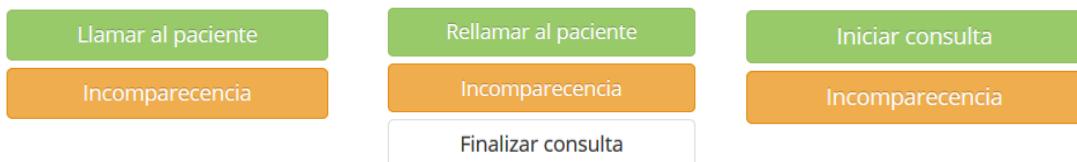


Ilustración 86. Acciones consulta

El primer caso es en una cita online que no ha sido iniciada, se permite llamar al paciente o marcar que no ha comparecido el paciente. El segundo caso es una vez iniciada la consulta online se permite volver a llamar al paciente, marcar como no comparecida o finalizar la consulta, que esto último cerraría el informe. El último caso es en una cita presencial, se podría iniciar la consulta o marcar como no comparecida.

El informe se habilita para ser llenado una vez se inicie la cita, y su información será guardada automáticamente hasta que se dé la cita por finalizada.

La funcionalidad de la videollamada consta de 4 pasos:

1. El medico pulsa en llamar al paciente, lo que provoca el inicio de la llamada, y el navegador solicitará acceso a la cámara y micrófono:

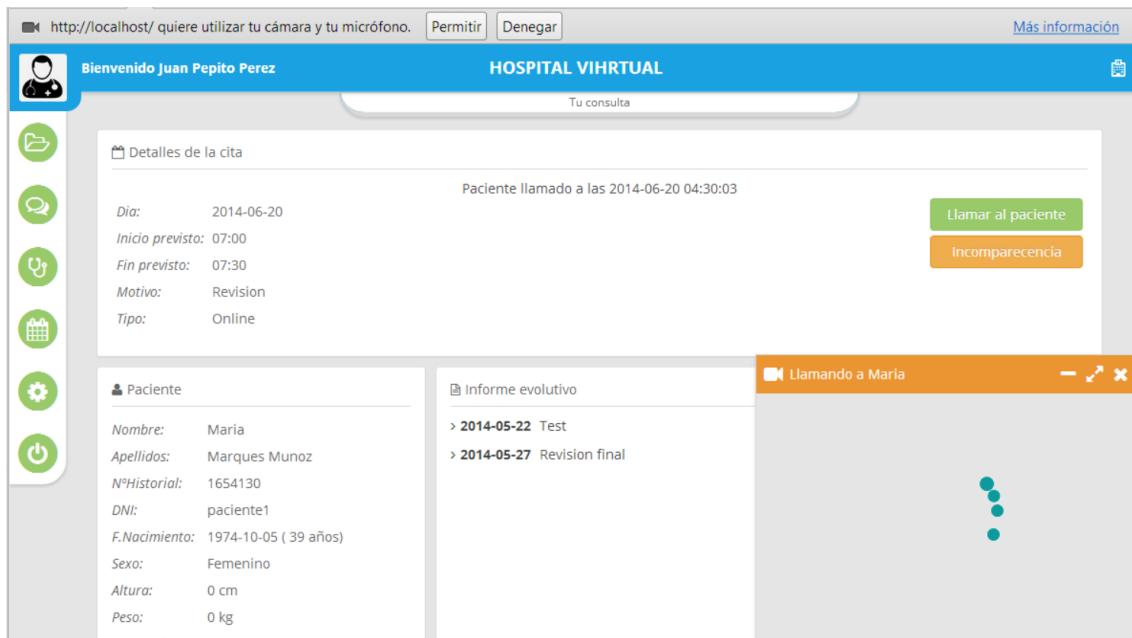


Ilustración 87. Inicio videollamada

2. El paciente recibirá la llamada en su dispositivo, la cual deberá aceptar y tras ello, el navegador igual que al profesional sanitario le solicitará acceso a la cámara y micrófono.

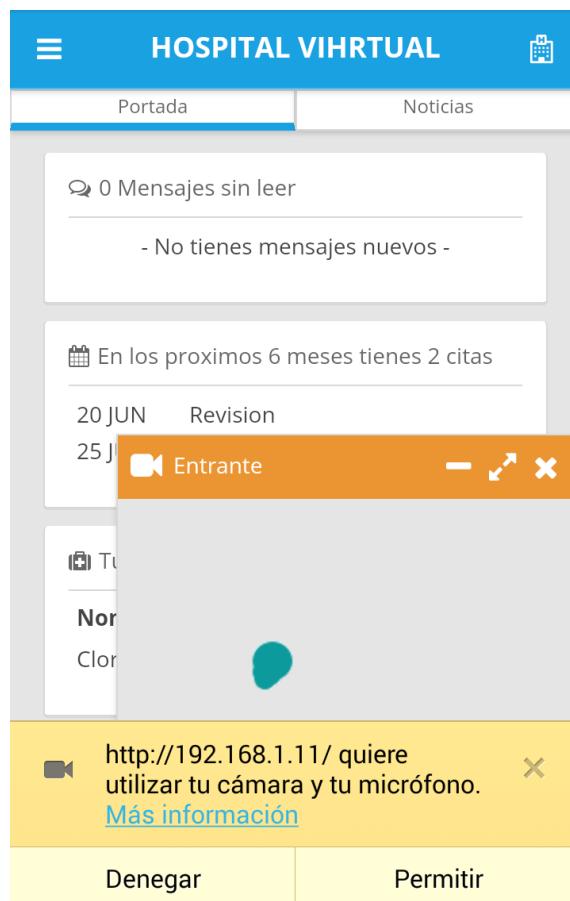


Ilustración 88. Recepción de videollamada

3. Una vez aceptado se iniciará la videollamada entre los dos usuarios

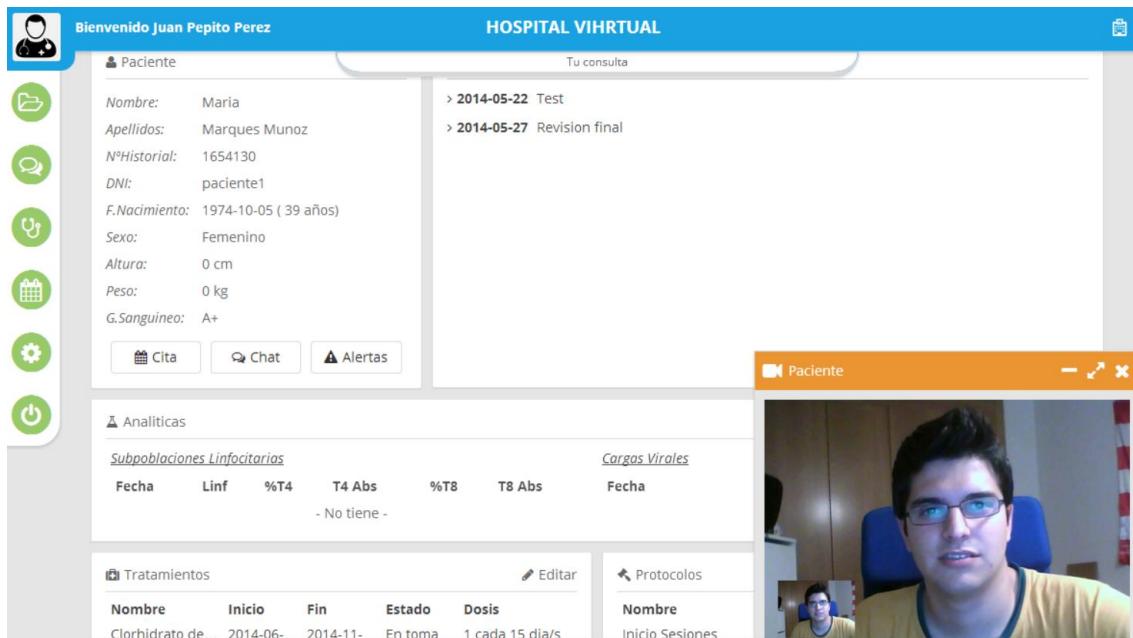


Ilustración 89. Videollamada en funcionamiento

4. Esta vista de videollamada tiene disponibles tres acciones:

a. Minimizar:

Se minimizara la videollamada pero seguirá estando disponible por voz.

b. Maximizar:

Abre a pantalla completa la videollamada

c. Cerrar:

Cuelga la llamada

Una vez dada por finalizada o no comparecida la cita ya no se podrá acceder a la misma sala.

7 Bibliografía

En este punto se incluye la bibliografía que ha sido consultada:

- [1] Recuso Web sobre con información sobre el VIH
<http://www.nlm.nih.gov/medlineplus/spanish/hivaids.html>
- [2] Recurso Web con información sobre las fases del VIH
<http://www.aids.gov.br/es/pagina/sintomas-y-fases-de-la-enfermedad>
- [3] Recurso Web con información sobre los tratamientos posibles del VIH
<http://www.cruzroja.es/vih/Tratamiento-VIH.html>
- [4] Informe anual elaborado en 2013 sobre la epidemiología mundial del VIH en 2012
http://www.unaids.org/en/media/unaids/contentassets/documents/epidemiology/2013/gr2013/201309_epi_core_en.pdf
- [5] Recurso Web con información del estado mundial del SIDA en 2012
<http://blogs.worldbank.org/opendata/es/el-estado-mundial-del-sida-en-cuatro-graficos>
- [6] Recurso Web con información sobre la Telemedicina
<http://es.wikipedia.org/wiki/Telemedicina>
- [7] Recurso Web con información sobre la calidad en la información de la salud
<http://portal.guiasalud.es/web/guest/calidad-informacion-salud>
- [8] Documento sobre el proyecto de VIH a domicilio a través de la televisión
Mairer RL, Clark R. Leveraging the Cable Network for Home-based HIV Disease Management. New Orleans, LA : Louisiana State University Health Sciences Center.
- [9] Documento sobre el proyecto de televisita de pacientes VIH
Eng RH, Provenzo R. Experience from a pilot telemedicine clinic for management of HIV patients in the VA New Jersey Health Care System (VANJHCS) . 2002.
- [10] Noticia sobre la presentación del Hospital VIHrtual en el Clínic de Barcelona
<http://www.webconsultas.com/sida/un-pionero-sistema-de-telemedicina-para-atender-pacientes-con-vih-2981>
- [11] Recurso Web con estadísticas sobre el uso de dispositivos móviles
<http://www.smartinsights.com/mobile-marketing/mobile-marketing-analytics/mobile-marketing-statistics/>

- [12] Recurso Web con información sobre metodologías de desarrollo de software
http://es.wikipedia.org/wiki/Metodolog%C3%ADa_de_desarrollo_de_software
- [13] Recurso Web con información sobre el desarrollo software de forma iterativa incremental
<http://www.proyectosagiles.org/desarrollo-iterativo-incremental>
- [14] Recurso Web con información sobre PHP
<http://www.desarrolloweb.com/articulos/392.php>
- [15] Recurso Web con información sobre PHP
http://es.wikipedia.org/wiki/PHP#Caracter%C3%ADsticas_de_PHP
- [16] Recurso Web con información sobre MySQL
<http://www.sinemed.com/recursos/docs/MySQL.pdf>
- [17] Recurso Web con información sobre MySQL
<http://www.gridmorelos.uaem.mx/~mcruz//cursos/miic/MySQL.pdf>
- [18] Recurso Web con información sobre el protocolo SIP
http://www.quarea.com/es/sip_session_initiation_protocol
- [19] Recurso Web con información sobre HTML
<http://es.wikipedia.org/wiki/HTML>
- [20] Recurso Web con información sobre HTML5
<http://es.wikipedia.org/wiki/HTML5>
- [21] Recurso Web con información sobre CSS
http://librosweb.es/css/capitulo_1.html
- [22] Recurso Web con información sobre JavaScript
<http://www.desarrolloweb.com/javascript/>
- [23] Recurso Web con información sobre Ajax
http://es.wikipedia.org/wiki/Ajax_framework
- [24] Recurso Web con información sobre el protocolo WebRTC
<http://www.3cx.es/voip-sip/webrtc/>
- [25] Recurso Web con información sobre el framework de Bootstrap
<http://jorgelessin.com/que-es-bootstrap-y-como-funciona-en-el-diseno-web/>
- [26] Recurso Web con información sobre jQuery
<http://es.wikipedia.org/wiki/JQuery>
- [27] Recurso Web con información sobre los servidores Apache

http://es.wikipedia.org/wiki/Servidor_HTTP_Apache

[28] Recurso Web con información de la aplicación Xampp

<http://es.wikipedia.org/wiki/Xampp>

[29] Recurso Web con información sobre Git

<http://git.micronautas.com/>

[30] Recurso Web con información sobre Git

<http://barradevblog.wordpress.com/2013/01/21/que-es-gitgithub/>

[31] Documento Web con información sobre PHP y MVC

<http://www.etnassoft.com/biblioteca/poo-y-mvc-en-php/>

[32] Recurso Web con información sobre accesibilidad

<http://www.w3c.es/Traducciones/es/WAI/intro/accessibility>

[33] Recurso Web con información sobre accesibilidad (WCAG 2.0)

<http://www.w3.org/TR/UNDERSTANDING-WCAG20/intro.html#introduction-fourprincs-head>

[34] Recurso Web con información sobre accesibilidad (WCAG 2.0)

<http://olgacarreras.blogspot.com.es/2007/02/wcag-20.html>