



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Aplicación para controlar el estado de pacientes
de cuidados intensivos

Grado en Ingeniería Informática

Autor: Iván Alonso Rubio

Tutor: Julio Pons Terol

Curso académico 2020-2021

Resumen

Se ha desarrollado una aplicación para controlar el estado de los pacientes ingresados en la Unidad de Cuidados Intensivos (UCI). De esta manera se podrá llevar a cabo una gestión más controlada, actualizada y dinámica del pase de información de los médicos entre turnos de trabajo. La idea de esta aplicación nace para ayudar a mejorar la gestión de la elevada carga existencial de los ingresos hospitalarios en dichas unidades debido a los estragos de la pandemia de 2020.

La aplicación se desarrollará como una aplicación web responsive, la aplicación usará *HTML*, *CSS*, *JavaScript* junto con las librerías *jQuery* y *Bootstrap* para la parte de frontend, mientras que utilizará *PHP* y *MySQL* para la parte de backend. Se ha utilizado *AJAX* como técnica de programación para la creación de las páginas interactivas. La información será transportada entre cliente y servidor en formato *JSON*.

Palabras clave: aplicación, cuidados intensivos, UCI, pacientes, médicos, hospital, JavaScript, MySQL, AJAX, JSON.

Abstract

The aim of this document is to explain the development of an application to control the state of the patients in the Intensive Care Unit (ICU). This will allow the doctors to give the information of the patients in a more controlled, updated and dynamic way. This application was born due to the exponential number of hospital admissions in those units as a result to the ravages of the 2020 pandemic.

The application will be developed as a responsive web application, the application will use HTML, CSS, JavaScript together with the jQuery and Bootstrap libraries for the frontend part, while it will use PHP and MySQL for the backend part. AJAX has been used as a programming technique for the creation of interactive pages. The information will be transported between client and server in JSON format.

Keywords: application, intensive care, ICU, patients, doctors, hospital, JavaScript, MySQL, AJAX, JSON.



Tabla de contenidos

Contenido

1.	Introducción	11
1.1	Motivación.....	11
1.2.	Objetivos.....	11
2.	Estado del arte	13
2.1.	Situación actual de la tecnología.....	13
2.1.1.	INURSE.....	13
2.1.2.	Aplicación móvil para la gestión de pacientes con Mieloma Múltiple.....	14
2.1.3.	Aplicación en un hospital del análisis de procesos sanitarios haciendo uso de tecnologías de minería de procesos.....	15
2.2.	Crítica del estado del arte	16
3.	Análisis del problema.....	17
3.1.	Introducción del problema.....	17
3.2.	Análisis marco legal y ético.....	18
3.3.	Análisis de riesgo	19
3.4.	Solución propuesta	21
4.	Diseño de la solución	23
4.1.	Arquitectura del sistema.....	23
4.1.1.	Fase 1: Análisis de la aplicación.....	23
4.1.2.	Fase 2: Diseño de la aplicación	23
4.1.3.	Fase 3: Desarrollo de la aplicación	24
4.2.	Diseño detallado.....	24
4.2.1.	Diseño de base de datos	24
4.3.	Tecnologías utilizadas.....	35
5.	Desarrollo de la aplicación propuesta	37
5.1.	Pantalla de inicio de sesión.....	37
5.2.	Pantalla principal.....	38
5.3.	Pantalla de categorías del paciente	40
5.3.1.	Situación del paciente.....	41
5.3.2.	Antecedentes personales	42
5.3.3.	Alergias.....	43
5.3.4.	Pruebas diagnósticas.....	44



5.4.	Pantalla de informes	45
5.4.1.	Análisis general	46
5.4.2.	Análisis categorías	47
5.4.3.	Análisis patología.....	48
5.5.	Pantalla del chat	49
5.6.	Gestión de administrador	50
6.	Implantación	53
7.	Pruebas	55
8.	Conclusiones	57
9.	Trabajos futuros.....	59
9.1.	Formación.....	59
9.2.	Versión de la aplicación.....	59
10.	Referencias bibliográficas y citas	61
11.	Apéndices	63
12.	Glosario	87

Índice de ilustraciones

Ilustración 1: Pantalla principal INURSE	13
Ilustración 2: Pantalla Mockup aplicación Mieloma Múltiple	14
Ilustración 3: Pantalla PALIA.....	15
Ilustración 4: Pantalla principal de la aplicación.....	21
Ilustración 5: diagrama de clases UML.....	28
Ilustración 6: Pantalla de inicio de sesión del usuario.....	37
Ilustración 7: Pantalla principal.....	38
Ilustración 8: modal de alta del paciente	39
Ilustración 9: Pantalla de categorías del paciente.....	40
Ilustración 10: Pantalla del sistema del paciente	41
Ilustración 11: Pantalla de antecedentes personales del paciente.....	42
Ilustración 12: Pantalla de alergias del paciente	43
Ilustración 13: Alerta que contiene alergias el paciente.....	43
Ilustración 14: Pantalla visualización de pruebas diagnosticas	44
Ilustración 15: Modal de maximización de imagen de pruebas diagnósticas	44
Ilustración 16: Pantalla de visualización de informes	45
Ilustración 17: Informe de estadísticas generales	46
Ilustración 18: Informe de estadísticas por categorías	47
Ilustración 19: Informe de estadísticas por patologías	48
Ilustración 20: Pantalla los chats a los que pertenece el usuario.....	49
Ilustración 21: Pantalla abierta de un chat.....	49
Ilustración 22: Modal de creación de chat	50
Ilustración 23: Pantalla de administración	51
Ilustración 24: : Modal de actualización de un usuario	51



Agradecimientos

Desde que tengo memoria he sentido curiosidad por cómo funcionan las cosas. En mi comunión me regalaron mi primer ordenador, un Pentium. Aunque no sabía cómo funcionaban las cosas y me tiraba más tiempo viendo la pantalla y accediendo entre carpetas, me acabó llamando mucho la atención.

Al cumplir los 16 años accedí por primera vez a un ciclo formativo de informática y por fin pude empezar a entender cómo unas pulsaciones magnéticas podían convertirse en información. Después de asistir a dos ciclos formativos (Medio y Superior) de informática y estar trabajando durante unos años, vi que no había conseguido mi objetivo. Quería aprender aún más y poder hacer mis propios desarrollos. Así que por fin accedí al grado de ingeniería informática.

Durante estos años he conocido compañeros y profesores estupendos con los que he podido desarrollarme tanto personal como profesionalmente. Siempre he querido desarrollar aplicaciones de ámbito social, para ayudar a todas las personas que estén a mi alcance.

En la pandemia por la Covid19 pude observar gracias a mi pareja Sara, médico intensivista, que existían problemas de traspaso de información en algunos hospitales, idea que motivó este trabajo. Gracias por estar siempre a mi lado, con una paciencia increíble y apoyarme en todos mis nuevos proyectos. Con su ayuda y gracias a los consejos de los profesores Julio Pons y Juan Carlos Casamayor pude orientarme para la realización de este proyecto.

Gracias a mi compañero Carlos, por todos esos momentos que hemos pasado durante las clases y fuera de ellas. Todos los días y noches que hemos invertido repasando los exámenes hasta último momento.

Agradecer a mi familia y en especial a mi padre Sergio, por ayudarme y animarme siempre en todos esos momentos en los que uno quiere rendirse e ir por el camino fácil.

Por último, agradecer a mi tío Paco, que me ayudó en toda mi estancia en Valencia y que, aunque ya no esté con nosotros, seguirá siendo una parte importante de mi vida.

Me gustaría añadir una cita de Edsger W. Dijkstra, «La informática tiene que ver con los ordenadores lo mismo que la astronomía con los telescopios».

Iván Alonso Rubio



1. Introducción

1.1 Motivación

Este proyecto surge por la necesidad de tener toda la información de los pacientes de la sala de cuidados intensivos en un único lugar, para que todos los médicos puedan observar el estado de los pacientes y así, facilitar el cambio de información entre los turnos del hospital.

1.2. Objetivos

El objetivo que se desea alcanzar es el desarrollo de una aplicación para la gestión de la sala de cuidados intensivos; esta aplicación debe responder a las necesidades del personal de la sala.

Con el desarrollo de esta aplicación se desea alcanzar los objetivos principales que serían:

- ✓ Desarrollo de una aplicación web para la gestión de la sala de cuidados intensivos. Esta aplicación debe ser capaz de ver el estado general de la sala y poder visualizar información detallada de cada paciente.

- ✓ Guardar los datos en una base de datos MySQL para el desarrollo de futuras mejoras en versiones posteriores.

- ✓ Mejorar el intercambio de información de forma segura entre los facultativos y que pueda ser extensible al personal de enfermería.

- ✓ Evitar la pérdida de información de los pacientes.

- ✓ Mejorar la gestión de la sala de cuidados intensivos, organizando mejor los recursos disponibles.



Aplicación para controlar el estado de pacientes de cuidados intensivos

- ✓ Reducir el uso de documentación en papel, evitando la pérdida de documentación y posibles errores.
- ✓ Modernizar la sala de cuidados intensivo.
- ✓ Generar con el histórico de la sala de cuidados intensivos informes para futuros análisis.

2. Estado del arte

En este capítulo se hará un análisis de la situación actual del mercado, realizando una búsqueda para encontrar desarrollos que sean similares a las funciones que se desea con nuestra aplicación. Este análisis comprobará esas aplicaciones frente a nuestra aplicación para determinar si lo que ofrece es más novedoso y presenta nuevas funcionalidades no existentes en el mercado.

2.1. Situación actual de la tecnología

2.1.1. INURSE

Es una aplicación para dispositivos móviles destinada a ser usada dentro de los hospitales, y por ello la solución propuesta ofrecerá una aplicación con dos frontend, uno para el personal sanitario y otro para los pacientes hospitalizados. Está disponible para sistemas operativos Android [1].

Ofrece diferentes servicios, todos ellos orientados a facilitar la actividad clínica del personal sanitario, incrementar la eficiencia del proceso asistencial y mejorar la atención de los pacientes.

Cada frontend ofrecerá diferentes servicios, todos ellos orientados a facilitar la actividad clínica del personal sanitario, incrementar la eficiencia del proceso asistencial y mejorar la atención de los pacientes.

Además, esta aplicación realizará una integración con el servicio Firebase Cloud Messaging para la recepción de notificaciones instantáneas que informen a los usuarios de determinados eventos en tiempo real.

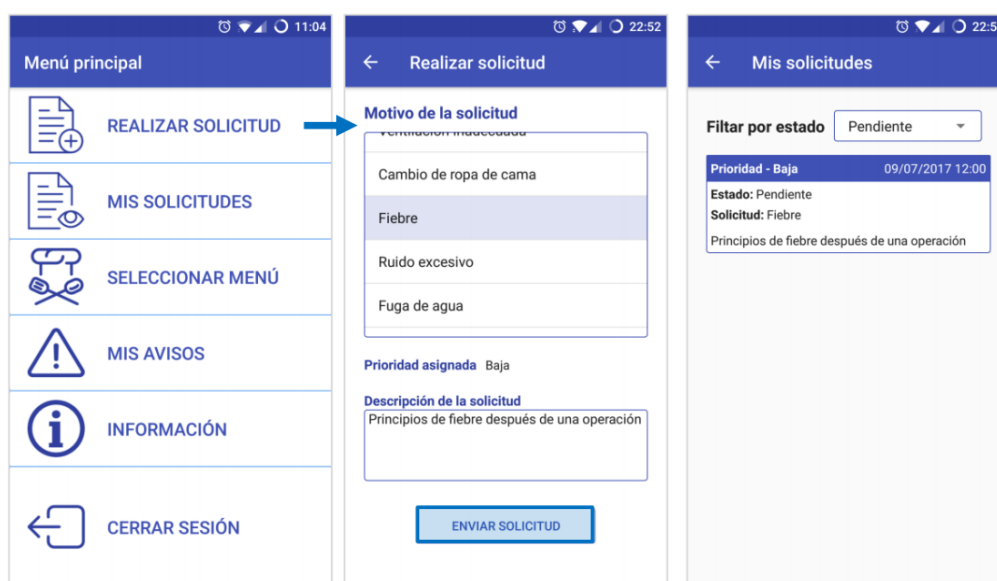


Ilustración 1: Pantalla principal INURSE

2.1.2. Aplicación móvil para la gestión de pacientes con Mieloma Múltiple.

El objetivo de este proyecto es llevar un control de los pacientes crónicos con Mieloma Múltiple y así mejorar su calidad de vida. El sistema permite introducir el peso, los síntomas, rellenar cuestionarios para llevar un control del estado anímico del paciente y la calidad de vida de este [2].

El sistema se integra con otra aplicación web, donde los profesionales sanitarios pueden consultar los datos enviados desde la aplicación móvil y posteriormente realizar diagnósticos o llevar el control del paciente.



Ilustración 2: Pantalla Mockup aplicación Mieloma Múltiple

2.1.3. Aplicación en un hospital del análisis de procesos sanitarios haciendo uso de tecnologías de minería de procesos.

En este proyecto, se ha utilizado la herramienta de minería de procesos PALIA para ayudar a los gestores del Hospital Universitario y Politécnico de la Fe a conocer y mejorar la gestión de las Consultas Externas del hospital. Tras estudiar el funcionamiento del área y los datos recogidos en ella, se obtuvieron los flujos de pacientes usuarios del área de Consultas Externas, permitiendo identificar anomalías y responder a preguntas concretas de sus gestores [3].

Usa PALIA (algoritmo de inferencia de registros basados en actividades paralelas) como herramienta de PM (process mining).

Los resultados obtenidos manifiestan el potencial de las tecnologías de minería de procesos en la gestión de las Consultas Externas de los hospitales y abren paso a futuros proyectos en este u otros procesos sanitarios.

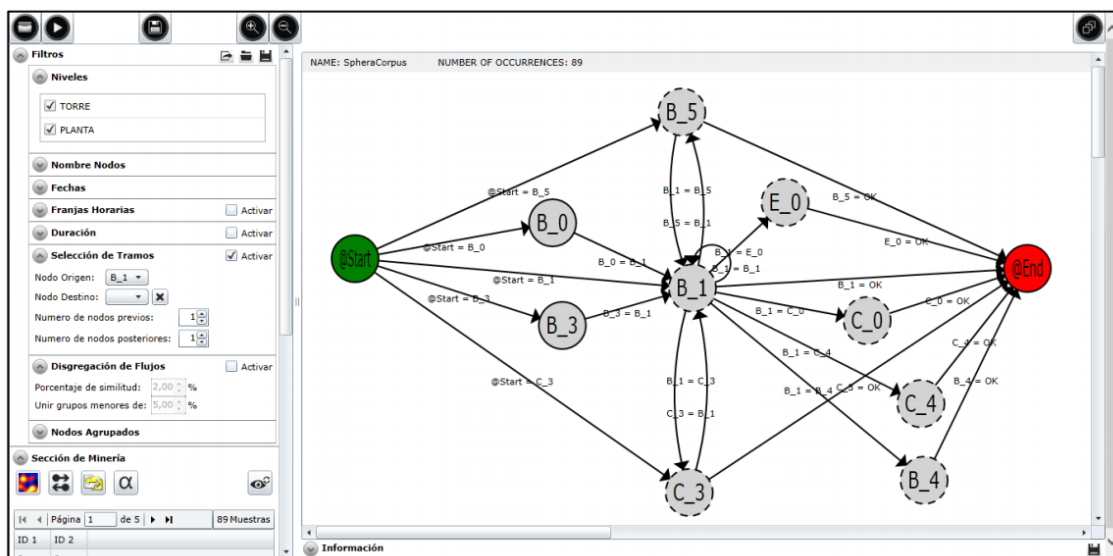


Ilustración 3: Pantalla PALIA

2.2. Crítica del estado del arte

Las aplicaciones analizadas en el apartado anterior ofrecen una serie de funcionalidades que facilitan la actividad clínica de los facultativos, residentes y personal del hospital. Mejorando las necesidades según donde sean utilizadas, pero cada una de estas aplicaciones están enfocadas en un ambiente profesional diferente.

Con nuestra aplicación se pretende proporcionar una serie de funcionalidades que no cubre ninguna de estas aplicaciones.

La primera novedad es que es una aplicación orientada a la sala de cuidados intensivos, que permite en un solo vistazo ver el estado general de la sala. Además, permite añadir información detallada de cada uno de los pacientes durante su estancia, permitiendo recibir información de manera más detallada y precisa.

La segunda novedad, es que permite insertar imágenes desde nuestra aplicación web, de esta forma podemos añadir imágenes de analíticas sanguíneas o pruebas de imágenes complementarias como pueden ser imágenes de tórax, ecografías, etc.

Por último, permite la visualización de gráficas de los pacientes que han pasado por la sala de cuidados intensivos, permitiendo al personal realizar análisis o planificaciones para futuros tratamientos.

Con estas novedades se pretende cubrir una serie de funcionalidades que no cubre ninguna aplicación actual, con la finalidad de aportar nuevas herramientas para mejorar el estado de los pacientes de cuidados intensivos.

3. Análisis del problema

3.1. Introducción del problema

Habitualmente, en las Unidades de Cuidados Intensivos, se realiza el “pase”, que consiste en el intercambio de información entre turnos de trabajo. En dichos pases, se reúnen los médicos (facultativos y residentes) e incluso a veces personal de enfermería, para exponer los cambios clínicos que se han producido en los pacientes (por ejemplo, durante la guardia), así como para debatir el manejo más adecuado de aquellos pacientes más complejos. Generalmente, se realiza a través de una tabla de Word rellenable, en la que cada día se actualizan los datos más relevantes (nombre del paciente, edad, días de estancia en UCI, y situación clínica en los diferentes sistemas: neurológico, hemodinámico, respiratorio, renal...).

Dentro de las limitaciones principales de este sistema, se incluyen:

- Ausencia de información relevante de cada paciente, ya que la extensión de la tabla de Word es limitada, por lo que a veces no se dispone de datos como las alergias medicamentosas, antecedentes médicos, o tratamiento habitual.
- No disponibilidad de material complementario como los resultados analíticos, siendo esto imprescindible en una Unidad de Cuidados Críticos, ya que la evolución de los pacientes es cambiante día a día.
- Ausencia de poder visualizar imágenes como electrocardiogramas o pruebas de imagen como radiografías o tomografías, no ya sólo por la implicación clínica con el paciente, sino por su utilidad formativa (muchos hospitales tienen residentes en formación, que se beneficiarían también de poder consultar a diario este material complementario).

Además, debido a la pandemia y la instauración de medidas de prevención como la distancia de seguridad, ha impedido que muchos médicos se puedan reunir como se hacía hasta la fecha, dado que la gran mayoría de salas de reuniones hospitalarias no cumplen con los criterios de aforo permitido.

Es por ello, por lo que surge la necesidad de crear un sistema alternativo al formato en papel habitual que por una parte permita el intercambio de información de forma segura entre los facultativos, que además pueda ser extensible a personal de enfermería (que tienen habitualmente una visión más cercana y complementaria del paciente), y que, por otra parte, garantice unos cuidados globales del paciente, sin fallos en la transmisión de la información, garantizando la seguridad en los cuidados.



Además, en muchas UCIs existe personal subespecializado en determinado tipo de patologías, existiendo grupos de trabajo que se enfocan en estudiar las últimas recomendaciones de los diferentes temas, por ejemplo, de patología infecciosa, de cuidados respiratorios, de patología cardíaca, etc. El sistema actual que se propone, al permitir la categorización de los pacientes según la patología que presentan al ingreso, puede permitir un reparto de los pacientes entre los médicos de una forma más específica, para así tener un manejo más subespecializado y con la última evidencia científica.

3.2. Análisis marco legal y ético

Para la realización de este proyecto se ha seguido la **LOPD** (Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales) como normativa a seguir en lo relativo a la protección de las personas físicas en lo que respecta al tratamiento de sus datos personales y a la libre circulación de estos datos, y completar sus disposiciones [4].

También se ha tenido en cuenta el reglamento europeo **RGPD** (Reglamento General de Protección de Datos) [5].

Las dos nacieron con el fin de preservar la intimidad de los ciudadanos en el uso que de sus datos hacen las empresas.

Sabiendo la existencia de estas dos legislaciones se tendrán en cuenta estos apartados:

- Dar de alta los ficheros en la Agencia Española de Protección de Datos.
- Elaborar y mantener actualizado el Documento de Seguridad.
- Obtener la legitimidad de los afectados.

A la hora de realizar nuestro proyecto, debemos tener en especial consideración estas legislaciones, para almacenar y tratar correctamente los datos de los pacientes [6].

Vamos a detallar los principales artículos que afectan a la protección de los datos:

- **Artículo 4:** Exactitud de los datos
- **Artículo 5:** Deber de confidencialidad
- **Artículo 6:** Tratamiento basado en el consentimiento afectado
- **Artículo 7:** Consentimiento de los menores de edad
- **Artículo 8:** Tratamiento de datos por obligación legal, interés público o ejercicio de poderes públicos.

- **Artículo 9:** Categorías especiales de datos
- **Artículo 10:** Tratamiento de datos de naturaleza penal

Propiedad intelectual

El sistema de información desarrollado en este proyecto está bajo licencia de software libre Copyleft [7].

Destacaremos que la propiedad intelectual será del autor del TFG Iván Alonso Rubio que será quien pondrá las restricciones para su distribución y modificación por parte de terceros.

3.3. Análisis de riesgo

Para poder poner los medios que minimicen o eviten el riesgo, lo primero es evaluar su incidencia, alcance y peligrosidad.

El artículo 5.1 f) de RGPD se refiere a la “integridad y confidencialidad” de los datos personales y establece que las organizaciones deben poder garantizar la protección adecuada. El objetivo es evitar que los datos personales que procesa y almacena se vean comprometidos, accidental o deliberadamente.

La inexistente, inadecuada o ineficaz seguridad de la información pone en riesgo los sistemas y servicios y puede terminar siendo la causa de daños reales a las personas.

Algunos (no los únicos) ejemplos de las consecuencias de la pérdida o violación de datos personales serían los seis siguientes:

- Fraude de identidad.
- Transacciones falsas con tarjeta de crédito.
- Intimidación.
- Exposición de las direcciones del personal de servicio, policías y funcionarios de prisiones, y de personas en riesgo de violencia doméstica.
- Solicitudes falsas de créditos fiscales.
- Fraude hipotecario.

Aunque no siempre se producen estos daños, las personas aún tienen derecho a ser protegidas de consecuencias menos graves, por ejemplo, vergüenza o inconveniencia.

Aplicar los niveles de seguridad LOPD que son básico, medio, alto y a lo que hacen referencia es el primer paso:



Aplicación para controlar el estado de pacientes de cuidados intensivos

1. **Nivel Básico:** se aplica a los documentos y ficheros que contienen datos identificativos.
2. **Nivel Medio:** se aplica a los ficheros que aporten información sobre las finanzas de la persona, como acerca de su patrimonio, crédito, solvencia, incluyendo información sobre la seguridad social.
3. **Nivel Alto:** Aquí se incluye toda la información que podría revelar la religión, orientación sexual, creencias, ideología, datos de afiliación sindical, afinidad política o salud de un individuo.

Una vez se han clasificado los riesgos, es hora de poner medidas. Estas pueden ir desde la definición de políticas y responsables, a la implementación de un sistema de encriptación de la información, métodos de autenticación de acceso o incluso la instalación de medidas de seguridad física para preservar la integridad de los ficheros no automatizados.

Estas medidas se tendrán que poner en marcha junto con el personal de sistemas del hospital. Decidiendo con ellos, según el sistema que tengan creado cuáles serían las medidas más apropiadas.

3.4. Solución propuesta

La solución propuesta es una aplicación web con posible adaptación a móvil. Está basada con la siguiente tecnología:

- En frontend con HTML, JavaScript, CSS
- En backend con PHP con una base de datos relacional en MySQL que almacenará la información de los pacientes del hospital [8].
- Técnicas de desarrollo web con AJAX [9].

La aplicación consta de una venta de login de los usuarios de la plataforma, una principal del estado de la sala de cuidados intensivos, tres tipos de ventanas para rellenar y visualizar la información de cada paciente, una ventana para visualizar los informes con gráficos, otra ventana para el chat entre los usuarios de la plataforma y si eres administrador, una última ventana para administrar los usuarios.



Ilustración 4: Pantalla principal de la aplicación

Esta es la ventana principal de la sala de cuidados intensivos, toda la aplicación mantendrá esta tipología y colores.



4. Diseño de la solución

4.1. Arquitectura del sistema

En esta parte vamos a analizar las fases para desarrollar la solución, crear la arquitectura y ver la conexión entre ellos.

4.1.1. Fase 1: Análisis de la aplicación

En la primera fase, se realizan reuniones con el personal de la sala de cuidados intensivos del hospital que ha participado en el proyecto para llevar a cabo un informe lo más exacto posible de sus necesidades.

Se han realizado encuestas de *Google form* para visualizar las necesidades del personal, entre los que detallamos qué funciones se realizan de manera manual y cuáles podría cubrir la aplicación.

Una vez recopilada la información se han analizado y se han determinado unos objetivos a cumplir.

4.1.2. Fase 2: Diseño de la aplicación

En la segunda fase, se diseñan los casos de uso y se confirman con el personal del hospital, asegurando que se cumplen las funciones requeridas.

Una vez asegurados los casos de uso se diseñan las pantallas de la aplicación, haciendo un prototipo en *Just in Mind*. De esta forma se valida la interfaz gráfica con el personal del hospital y se pueden realizar las modificaciones necesarias para cumplir las funciones deseadas.

Con este prototipo se asegura tener una aplicación lo más intuitiva y sencilla posible para los futuros usuarios de la aplicación.

Una vez se tiene el prototipo se diseña la base de datos, y se realizará un estudio de la base de datos para poder implementar el teorema de CAP (Consistencia, Disponibilidad y Tolerancia al Particionamiento) [10].



4.1.3. Fase 3: Desarrollo de la aplicación

Se han estudiado varios lenguajes de programación para el desarrollo de la aplicación [11]. En este caso se ha seleccionado PHP por las siguientes razones:

- Multiplataforma, puede correr en casi cualquier plataforma sin modificar el código fuente.
- Soporta gran cantidad de base de datos
- Es Open Source, disponible sin coste alguno.
- Permite crear páginas web dinámicas y el código es ejecutado en el lado del servidor.
- Lenguaje rápido, simple, robusto y el lenguaje más usado en programación web.

4.2. Diseño detallado

4.2.1. Diseño de base de datos

4.2.1.1. Primera fase: Análisis

En colaboración con el servicio de Medicina Intensiva del Hospital que ha participado en el proyecto se ha decidido crear un sistema de información para la gestión de la sala de cuidados intensivos.

Tras realizar varias entrevistas con personal de la sala de cuidados intensivos, se ha podido averiguar cuál es la información para tratar para la construcción de este sistema:

- El número de camas disponibles en la sala y qué paciente ocupa cada cama.
 - Número de cama.
 - Si está libre.
 - Nombre de la cama.
- Conocer de cada paciente todos los antecedentes personales que ha tenido para poder llevar a cabo el seguimiento.
 - Descripción detallada de los antecedentes personales.
 - Fecha de la última modificación de los antecedentes personales.
- Si el paciente presenta alguna alergia.
 - Descripción de las alergias.

- Fecha de la última modificación de las alergias del paciente.

- Se necesitará conocer la información de los sistemas del paciente, y se realizará a partir de una categorización de los distintos sistemas del cuerpo humano para indicar si estos están en óptimo funcionamiento o si presentan algún problema.
 - Descripción del estado del sistema.
 - Fecha de la última modificación del sistema del paciente.

- Tipos de sistemas.
 - Nombre del sistema del cuerpo humano.
 - Tipo de sistema, para mostrar los sistemas en una ventana u otra en caso de ser necesario.

- Al paciente se le realizarán una serie de pruebas diagnósticas, si se realizan se necesitará una descripción detallada de la misma y qué tipo de prueba se ha realizado.
 - Título para la prueba.
 - Descripción de los resultados.
 - Imagen del resultado de la prueba.

- Categorías de las pruebas.
 - Nombre de la categoría.
 - Tipo de categoría de prueba, para mostrar los datos en una ventana u otra en caso de ser necesario.

- Los pacientes padecerán alguna patología de las cuales se quieren categorizar para ayudar a la organización de la sala.
 - Nombre de la patología.
 - Descripción de la patología.



Aplicación para controlar el estado de pacientes de cuidados intensivos

Los usuarios dados de alta en el sistema deberán tener algún mecanismo de comunicación para facilitar el tratamiento de los datos entre ellos.

- Usuarios
 - Apodo en la plataforma
 - Contraseña
 - Nombre
 - Apellidos
 - Edad
 - Rol

- Mensaje
 - Contenido
 - Fecha

- Tipo de mensaje
 - Nombre

- Sala
 - Fecha de creación de sala

- Tipo de sala
 - Nombre

4.2.1.2. Descripción detallada del sistema

De cada cama del hospital se necesita saber si esa cama está libre y su número de cama. Cada paciente que ocupa una cama tendrá el nombre del paciente, la edad, la patología que padece y la categoría de esta patología. Cuando un paciente ingrese en el sistema se necesitará saber la fecha de entrada y la fecha de salida al darle el alta.

De los pacientes que se ingresan se revisarán los antecedentes que ha tenido con una descripción de estos.

De los distintos sistemas del cuerpo humano se sabrá el estado y la fecha de revisión, estas revisiones del sistema deberán estar asociadas a una categoría del sistema.

Al paciente estando ingresado se le realizarán una serie de pruebas diagnósticas de las cuales necesitaremos tener unas imágenes con un título y una pequeña descripción en caso de ser necesaria.

Tendremos una categorización para saber qué tipo de información tendremos que mostrar en el sistema según en la página que nos encontremos, esta categorización también nos ayudará a diferenciar qué tipo de prueba diagnóstica tenemos que guardar.

Los usuarios dados de alta en el sistema deberán tener disponible un sistema de mensajería entre ellos, para que tanto los administradores como usuarios puedan comunicarse en caso de ser necesario.

4.2.1.3. Segunda fase: Diseño conceptual

En la siguiente página representaremos gráficamente el sistema de información usando un diagrama de clases en UML [12].



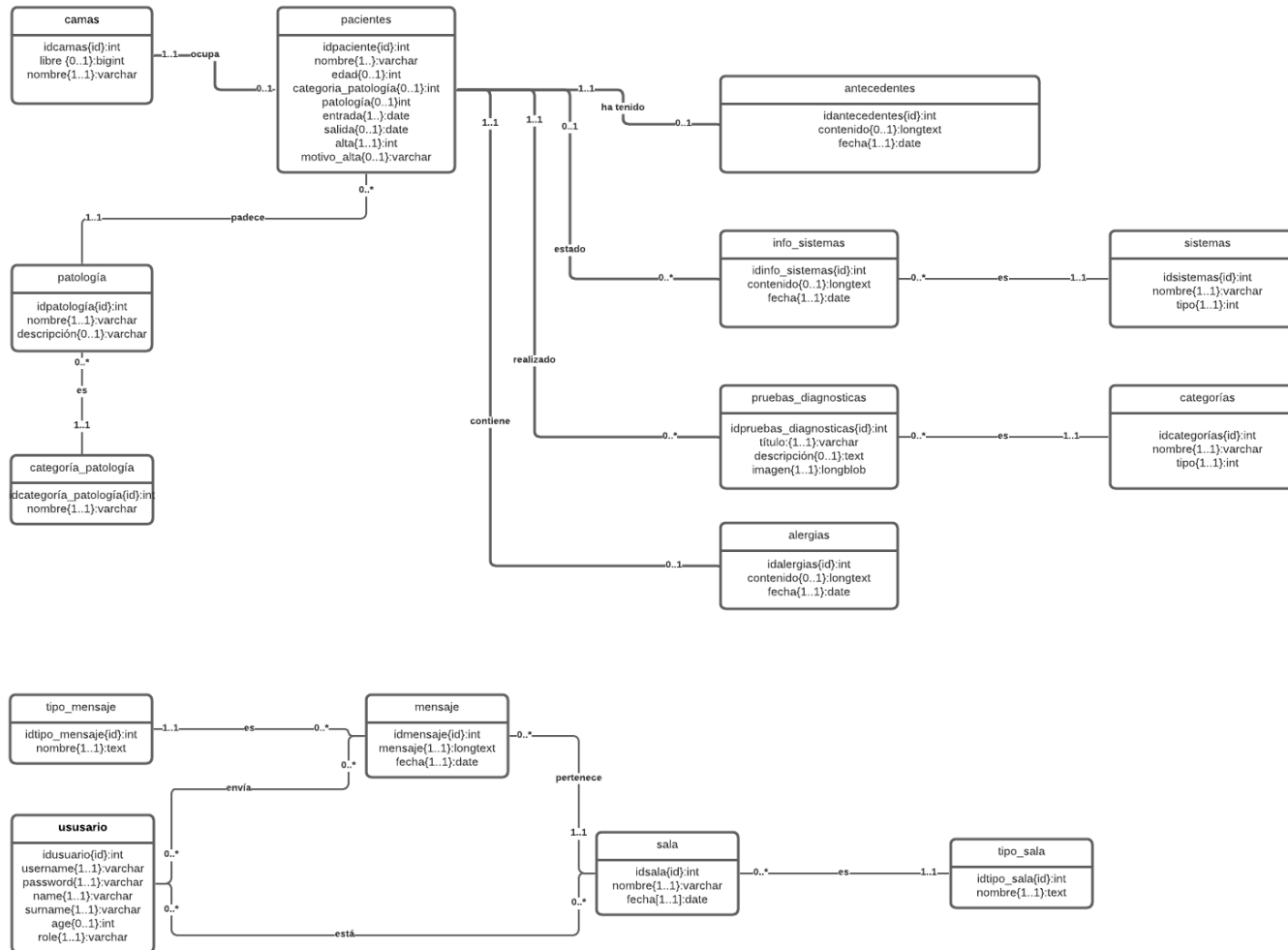


Ilustración 5: diagrama de clases UML

4.2.1.4. *Diseño lógico*

Definición de dominios, relaciones y restricciones del sistema de información [13]

Camas (idcamas: int (11), libre: bigint (1), nombre: varchar (45))

CP: {idcamas}

VNN: {nombre}

Pacientes (idpacientes: int (11), nombre: varchar (45), edad: int (3),

categoría_patología: int (11), patología: int (11), entrada: timestamp (6),

salida: timestamp (6), alta: varchar (45), motivo_alta: varchar (45),

cama: int (11))

CP: {idpacientes}

VNN: {nombre, entrada, alta}

CAj: {categoría_patología} → Patología(categoría) Borrado en cascada

CAj: {patología} → Patología(idpatología) Borrado en cascada

CAj: {cama} → Camas(idcamas) Borrado en cascada



Aplicación para controlar el estado de pacientes de cuidados intensivos

Patología (idpatologia: int (11), nombre: varchar (45), descripción: varchar (45),

categoría: int (11))

CP: {idpatologia}

VNN: {nombre}

CAj: {categoría} → Categoria_patologia(idcategoria_patologia) Borrado en cascada

Categoria_patologia (idcategoria_patologia: int (11), nombre: varchar (45))

CP: {idcategoria_patologia}

VNN: {nombre}

Antecedentes (idantecedentes: int (11), fk_cama: int (11), fk_paciente: int (11),

contenido: longtext, fecha: timestamp (6))

CP: {idantecedentes}

VNN: {fecha}

CAj: {fk_cama} → Pacientes(cama) Borrado en cascada

CAj: {fk_paciente} → Pacientes(idpaciente) Borrado en cascada

Info_sistemas (idinfo_sistemas: int (11), fk_sistemas: int (11), fk_cama: int (11),

fk_paciente: int (11), contenido: longtext, fecha: timestamp (6))

CP: {idinfo_sistemas}

VNN: {fecha}

CAj: {fk_sistema} → Sistemas(idsistemas) Borrado en cascada

CAj: {fk_cama} → Pacientes(cama) Borrado en cascada

CAj: {fk_paciente} → Pacientes(idpaciente) Borrado en cascada

Pruebas diagnósticas (idpruebas_diagnosticas: int(11), fk_categoria int (11),

fk_paciente: int (11), título: varchar (45), descripción: text,

imagen: longblob)

CP: {idpruebas_diagnosticas}

VNN: {título, imagen}

CAj: {fk_categoria} → Categorias(idcategorias) Borrado en cascada

CAj: {fk_paciente} → Pacientes(idpaciente) Borrado en cascada



Aplicación para controlar el estado de pacientes de cuidados intensivos

Alergias (idalergias: int (11), fk_cama: int (int), fk_paciente: int (11), contenido:

longtext, fecha: timestamp (6))

CP: {idalergias}

VNN: {fecha}

CAj: {fk_cama} → Pacientes(cama) Borrado en cascada

CAj: {fk_paciente} → Pacientes(idpaciente) Borrado en cascada

Sistemas (idsistemas: int (11), nombre: varchar (45), tipo: int (11))

CP: {idsistemas}

VNN: {nombre, tipo}

Categorías (idcategorias: int (11), nombre: varchar (45), tipo: int (11))

CP: {idsistemas}

VNN: {nombre, tipo}

Usuario (idusuario: int (11), username: varchar (45), password: varchar (60),

name:varchar(45), surname: varchar(45), age: int (3), role: varchar (45))

CP: {idusuario}

VNN: {username, password, name, surname, role}

Mensaje (idmensaje: int (11), mensaje: longtext, fecha: timestamp (6),

idusuario:int (11), idsala: int (11), idtipo: int (11))

CP: {idMensaje}

VNN: {mensaje, fecha}

CAj: {idusuario} → Users(idusuario) Borrado en cascada

CAj: {idsala} → Sala(idsala) Borrado en cascada

CAj: {idtipo} → Tipo_mensaje(idtipo) Borrado en cascada

Tipo_mensaje (idtipo_mensaje: int (11), nombre: varchar (45))

CP: {idtipo_mensaje}

VNN: {idtipo_mensaje, nombre}

Tipo_sala (idtipo_sala: int (11), nombre)

CP: {idtipo_sala}

VNN: {idtipo_sala, nombre}

Sala (idsala: int (11), nombre: varchar (45), fecha: timestamp (6), idtipo: int (11))

CP: {idsala}

VNN: {idsala, nombre, fecha}

CAj: {idtipo} → Tipo_sala(idtipo_sala) Borrado en cascada



Aplicación para controlar el estado de pacientes de cuidados intensivos

Sala_usuario (idsala_usuario: int (11), idsala: int (11), idusuario: int (11))

CP: {idsala_usuario}

VNN: {id_sala_usuario}

CAj: {idsala} → Sala(idsala) Borrado en cascada

CAj: {idusuario} → Users(idusers) Borrado en cascada



4.3. Tecnologías utilizadas

En este apartado vamos a explicar todas las tecnologías que hemos utilizado para el desarrollo de la solución. La gran mayoría de estas tecnologías tienen licencias gratuitas, sin ningún tipo de coste alguno.

El desarrollo se ha realizado en un ordenador personal en Windows.



El desarrollo de la base de datos se ha realizado con **Xampp** versión 3.2.4 que consiste en un sistema gestor de base de datos MySQL, servidor web Apache y los intérpretes para lenguajes script PHP y Perl.



MariaDB versión 10.4.14, es un sistema de gestión de bases de datos derivado de MySQL con licencia GPL. Contiene mejoras sobre su predecesor en velocidad, mecanismos de pruebas y su facilidad de uso.



MySQL Workbench 8.0 CE, es una herramienta visual de diseño de bases de datos que integra desarrollo de software, administración de bases de datos, diseño de bases de datos, gestión y mantenimiento para el sistema de base de datos MySQL.



Visual studio code, es un editor de código fuente desarrollado por Microsoft para Windows, Linux y macOS. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código.



Aplicación para controlar el estado de pacientes de cuidados intensivos



Justinmind, es una herramienta de creación de prototipos y wireframing para la creación de prototipos de alta fidelidad de aplicaciones web y móviles.



Se ha utilizado **PHP** versión 7.4.10 lenguaje de código abierto y que se puede incrustar en HTML.

Librerías utilizadas:

- jQuery v3.3.1, es una biblioteca multiplataforma de JavaScript que se ha utilizado para el manejo del DOM, eventos y agregar interacción con AJAX a la página web.
- Bootstrap v5.0.0, biblioteca multiplataforma usada para aplicar el diseño de la aplicación web.
- Datatables, es un complemento para la biblioteca jQuery JavaScript que agrega todas estas características avanzadas a cualquier tabla HTML.
- Google charts, es un servicio web interactivo que crea gráficos a partir de información proporcionada por el usuario.

5. Desarrollo de la aplicación propuesta

En el siguiente capítulo se va a implementar la propuesta con las tecnologías explicadas en el capítulo anterior. Nos centraremos en el desarrollo de la interfaz gráfica.

La primera versión de la aplicación es una aplicación web de escritorio que se conecta a una base de datos relacional cuyo diseño hemos visto en la Ilustración 2. Es un modelo relacional de base de datos que se hospeda en el servidor que definan los administradores de sistemas del hospital. Al estar implementada con *MariaDB* es multiplataforma, facilitando la implementación para los administradores.

Para la creación de las pantallas de la aplicación se ha utilizado *Just in Mind*.

A continuación, se detallan las pantallas principales.

5.1. Pantalla de inicio de sesión

Para cumplir con la legislación, los datos de los usuarios deben estar protegidos. Por este motivo se ha creado una pantalla de inicio de sesión.

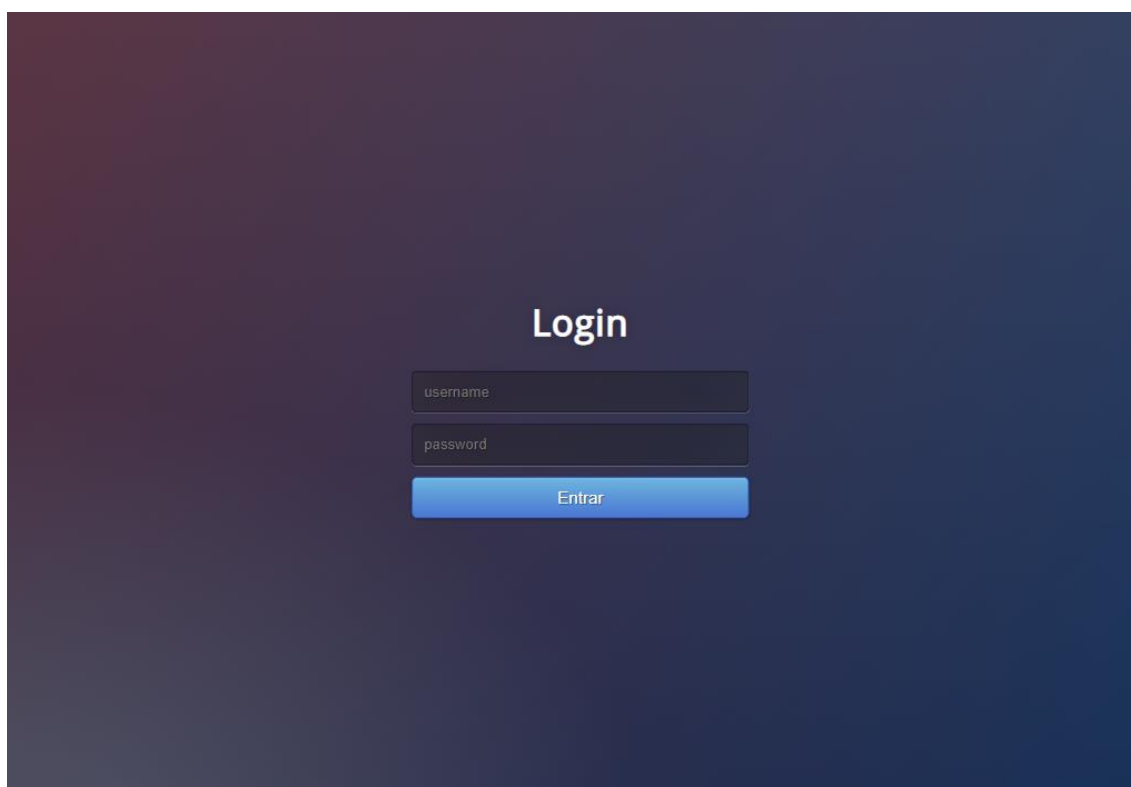


Ilustración 6: Pantalla de inicio de sesión del usuario

5.2. Pantalla principal

Una vez el usuario se ha validado con su autenticación, se procede a acceder a la pantalla principal de la aplicación.

En esta pantalla se podrá ver el estado resumido de la sala de Cuidados Intensivos. Se verán tantas camas como tenga el hospital con un breve resumen de cada paciente en cada cama: el identificador de la cama, su nombre, edad, categoría de la patología y la patología.



Ilustración 7: Pantalla principal

En cada cama se verán dos botones “Más información” y “Alta”.

El primero permite añadir o visualizar la información más relevante de este paciente.

El segundo se utilizará para dar de alta al paciente que puede ser por 3 motivos:

1. **Curación:** El paciente ya se ha estabilizado y se le dará el alta de la sala de cuidados intensivos.
2. **Éxito:** El tratamiento instaurado no ha sido efectivo y el paciente fallece.
3. **Traslado:** El paciente se ha trasladado a otro centro hospitalario.

Una vez se da de alta al paciente se le borra, quedando la cama disponible para el próximo.



Ilustración 8: modal de alta del paciente

En la parte superior se podrá visualizar el número de camas disponibles en la sala y los botones para acceder a diferentes funciones de la aplicación, como puedan ser informes, el chat y el panel de administración si se dispone de permisos.

5.3. Pantalla de categorías del paciente

Cuando accedemos a “Más información” de un paciente, esta pantalla permitirá visualizar e insertar información relevante sobre el paciente [14].

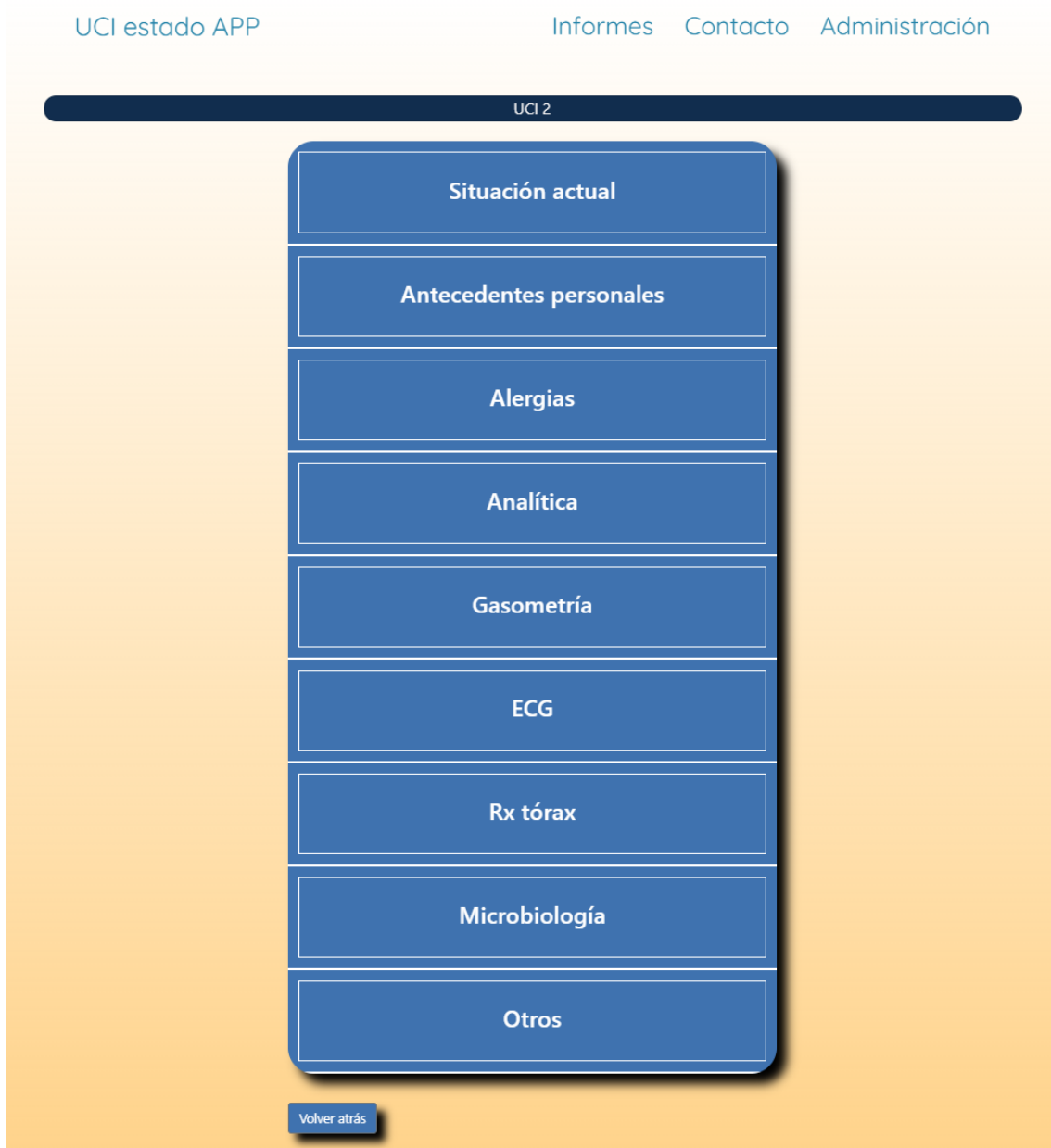


Ilustración 9: Pantalla de categorías del paciente

Toda la información está por categorías que en nuestro caso son:

5.3.1. Situación del paciente

Un resumen de la situación del cuerpo humano del paciente por sistemas: Neurológico, Hemodinámico, Respiratorio, Renal, Infeccioso, Abdomen/Metabólico, Otros; que contendrá las anotaciones realizadas y permitirá modificar esta información en caso de ser necesario.

The screenshot displays a web application interface for 'UCI estado APP'. At the top, there are navigation links for 'Informes', 'Contacto', and 'Administración'. Below these is a dark blue header bar with the text 'UCI 1'. The main content area is a vertical stack of seven blue-bordered boxes, each representing a medical system. Each box has a title and a text input field containing a status note. At the bottom of the stack are two buttons: 'Volver atrás' and 'Guardar'.

Sistema	Estado
Neurológico	Consciente y orientado
Hemodinámico	Sin drogas vasoactivas
Respiratorio	Con gafas nasales a 2 LPM
Renal	Función renal estable.
Infeccioso	Sin fiebre ni antibióticos
Abdomen/Metabólico	Tolerando dieta oral
Otros	alta próximamente

Ilustración 10: Pantalla del sistema del paciente

5.3.2. Antecedentes personales

Si el paciente contiene algún antecedente personal relevante se podrá visualizar o añadir, facilitando a tomar las medidas necesarias y ayudar al estado del paciente.



Ilustración 11: Pantalla de antecedentes personales del paciente

5.3.3. Alergias

En este apartado se podrán visualizar y añadir las alergias del paciente.



Ilustración 12: Pantalla de alergias del paciente

En caso de que el paciente presentase alguna alergia, el cuadro de la categoría “Alergias” se marcaría en rojo para notificar a los usuarios de este antecedente.

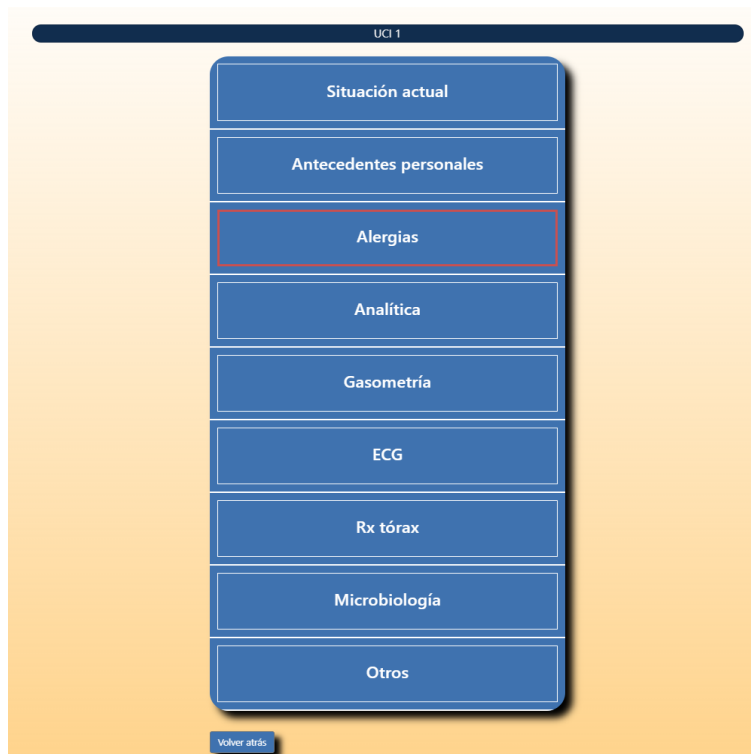


Ilustración 13: Alerta que contiene alergias el paciente

5.3.4. Pruebas diagnósticas.

En esta pantalla se podrá añadir imágenes de las pruebas complementarias, así como un título y una breve descripción de los hallazgos. A medida que se vayan añadiendo imágenes se verán en formato cuadrícula.

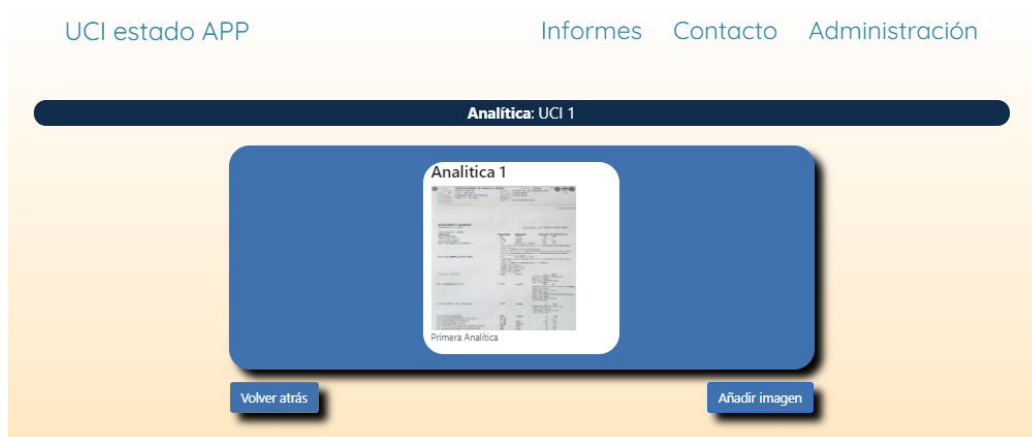


Ilustración 14: Pantalla visualización de pruebas diagnósticas

Al pulsar en la imagen, ésta se maximizará para visualizar mejor el contenido. Al maximizar, también se podrá eliminar la imagen desde el botón.

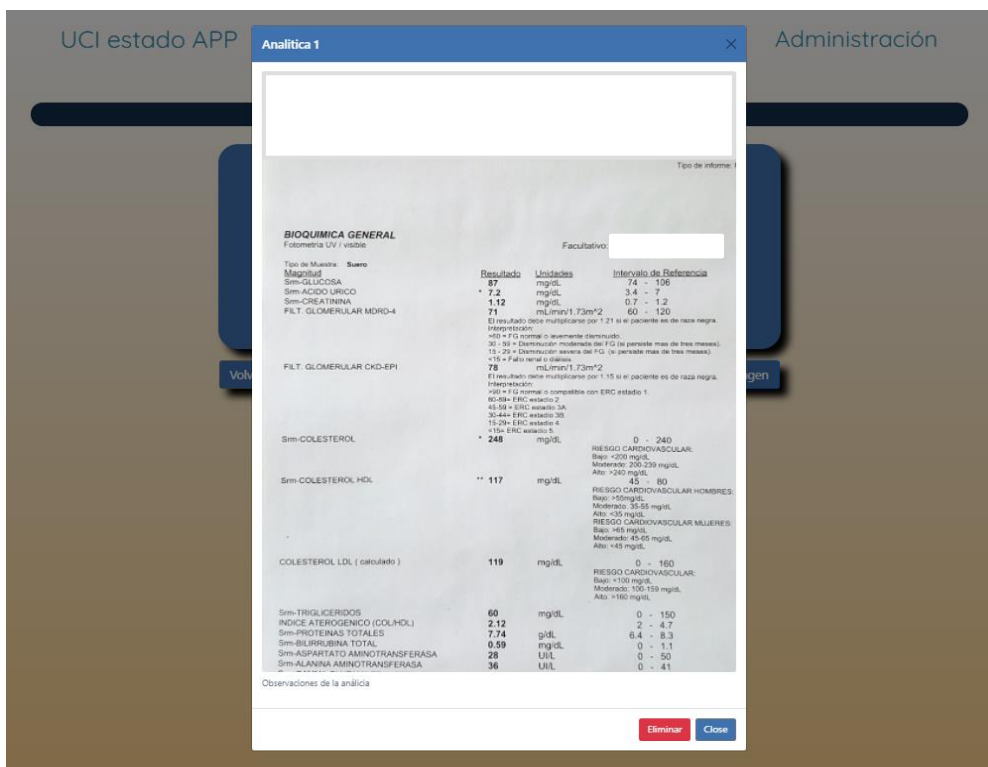


Ilustración 15: Modal de maximización de imagen de pruebas diagnósticas

5.4. Pantalla de informes

En esta pantalla se podrán visualizar gráficas detalladas sobre el estado de la sala de cuidados intensivos. Se podrán ver según la pestaña seleccionada por uno de los siguientes análisis: General, Categorías y Patologías.

Todos los análisis se podrán filtrar por rango de fechas para facilitar el análisis. En caso de no seleccionar ningún rango de fechas no se hará ningún filtrado, mostrándose desde que se empezó la aplicación hasta la fecha actual.

UCI estado APP Informes Chat Administración

Análisis general **Análisis categorías** Análisis patología

Estadísticas Generales

uciAPP

Fecha inicio:

Fecha fin:

Ilustración 16: Pantalla de visualización de informes

5.4.1. Análisis general

Mostrará tres tipos de gráficos.

1. El primero, será un gráfico de sectores que mostrará cuántos, y qué porcentaje de pacientes han sido dados de alta, filtrando por motivos de alta de los pacientes, que son curación, éxitus y traslado. Este gráfico permitirá saber de todos los pacientes que han ingresado en la sala de cuidados intensivos, cuántos se curan, fallecen o se trasladan.
2. El segundo, será un gráfico de barras que mostrará cuántos pacientes se han tenido de cada categoría (Coronarios, Sépticos, Miscelánea y Covid19), clasificándolos por el motivo del alta. Este gráfico permitirá saber de cada categoría, cuantos se curan, fallecen o se trasladan.
3. El tercero, será otro gráfico de barras, pero recogerá las 10 patologías más frecuentes en la sala de cuidados intensivos, mostrando el porcentaje pacientes que se han tenido de cada patología.

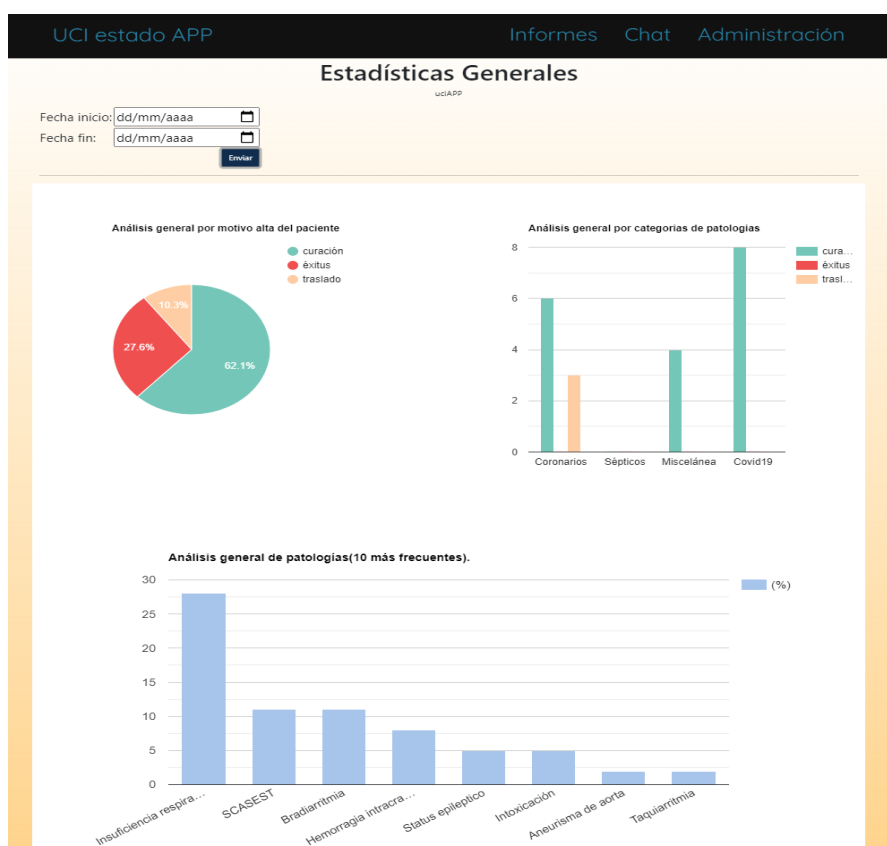


Ilustración 17: Informe de estadísticas generales

5.4.2. Análisis categorías

El análisis por categorías, además del filtro por rango de fechas, permite seleccionar cada una de las categorías (Coronarios, Sépticos, Misceláneas, Covid19).

En esta pestaña se verán dos tipos de gráficas.

1. La primera, será una gráfica de sectores por la categoría seleccionada mostrando el motivo de altas de los pacientes por esa categoría.
2. La segunda, será una gráfica de barras filtrada por la categoría seleccionada mostrando las patologías más frecuentes de esa categoría.



Ilustración 18: Informe de estadísticas por categorías

5.4.3. Análisis patología

En el análisis por patología además del filtro por rango de fechas y seleccionar la categoría (Coronarios, Sépticos, Misceláneas, Covid19) permite también seleccionar la patología.

Aquí se mostrará únicamente una gráfica, siendo una gráfica de sectores que muestra de la categoría y patología seleccionada el motivo de alta de los pacientes.



Ilustración 19: Informe de estadísticas por patologías

Con estos análisis se quiere facilitar el tratamiento de los pacientes, ayudando a la clasificación y revisando los casos más frecuentes en la sala.

5.5. Pantalla del chat

Esta pantalla permite la comunicación entre los usuarios de la plataforma, se pueden crear chat privados o de grupo. En la ventana principal se verá una tabla con el nombre del chat y su tipo. Se visualiza un cuadro de búsqueda para facilitar la búsqueda de un chat en concreto en caso de ser necesario.

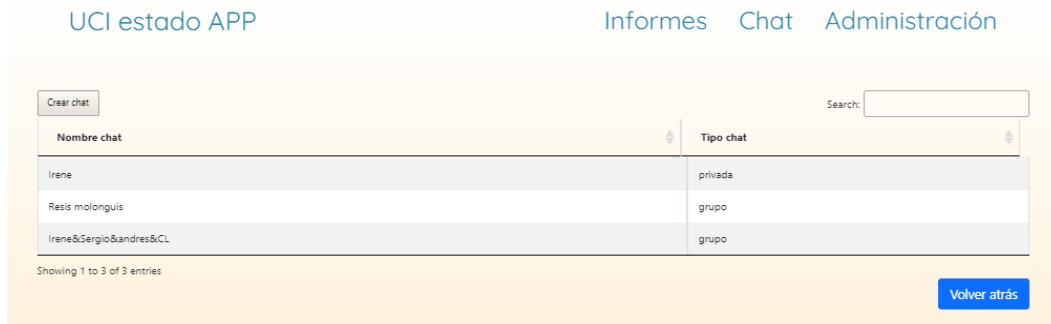


Ilustración 20: Pantalla los chats a los que pertenece el usuario

Una vez se pulse en el nombre del chat, aparecerá un chat en la parte inferior de la pantalla.

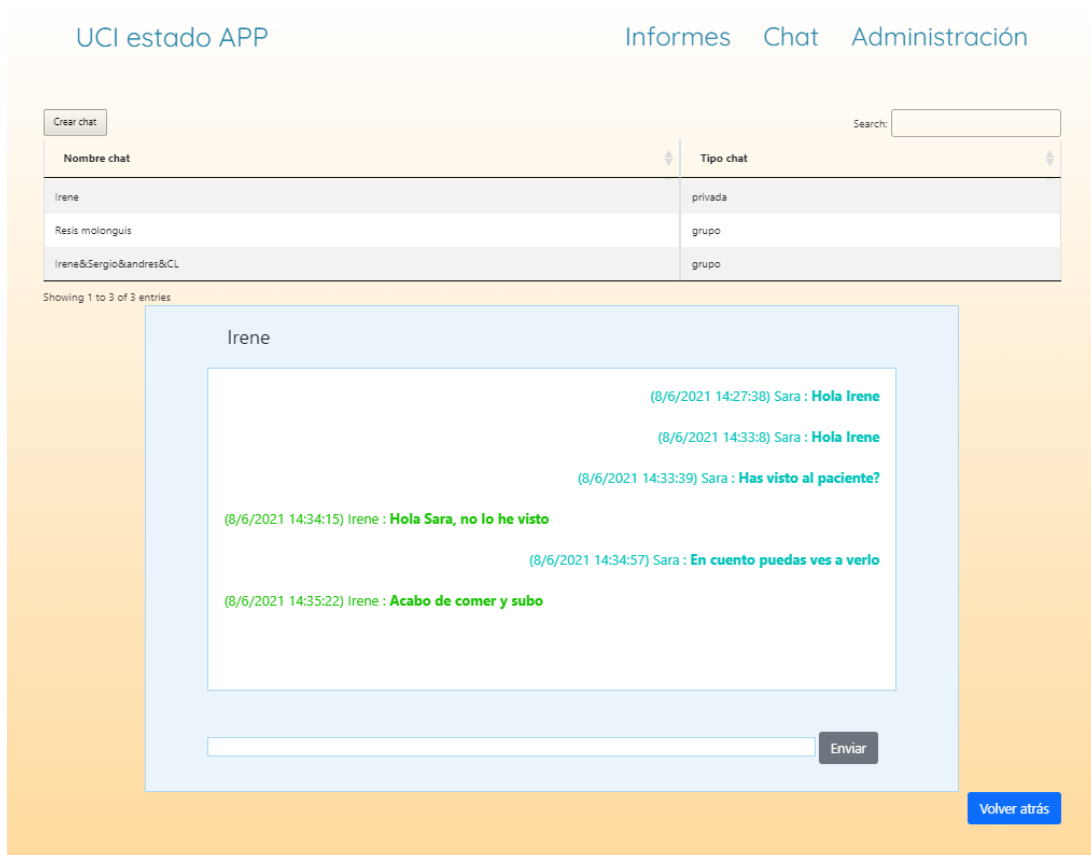


Ilustración 21: Pantalla abierta de un chat

En la parte superior existe un botón “Crear chat” que permite la creación de un chat entre los usuarios de la plataforma, pudiendo ser este privado o de grupo.

Cuando se pulse este botón, aparecerá un modal que pedirá un nombre del chat, su tipo, y se tendrá que añadir a los usuarios que serán miembros del chat.

Si se escoge más de un usuario, el tipo de chat pasará a ser de grupo y no podrá ser privado hasta existir sólo un usuario seleccionado.

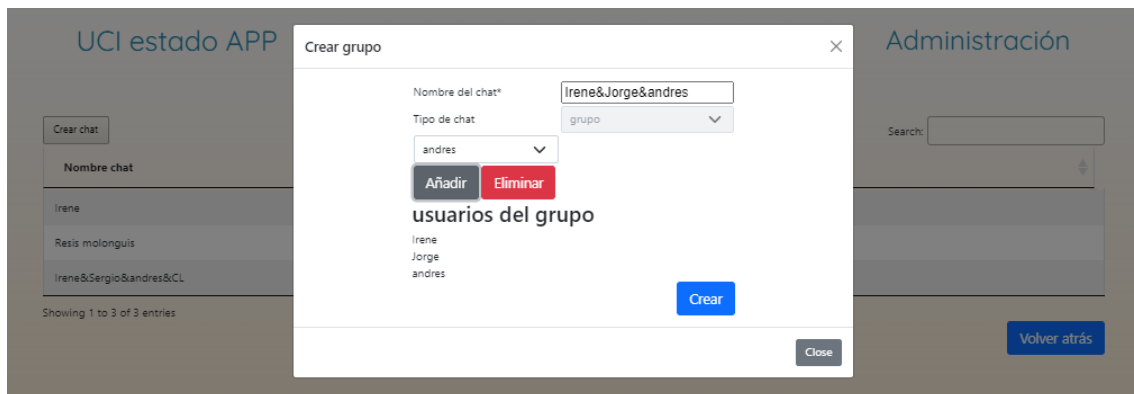


Ilustración 22: Modal de creación de chat

5.6. Gestión de administrador

En esta pantalla se podrá dar de alta, modificar y eliminar a los usuarios de la aplicación. Sólo será visible para los usuarios con el role de administrador.

Se visualiza una tabla con los datos los siguientes datos de los usuarios: nombre de usuario, nombre, apellidos, edad y role del usuario.

Añadir			Actualizar			Eliminar			Search: <input type="text"/>		
	#	Usuario	Nombre	Apellidos	Edad	Role					
<input type="checkbox"/>	1	21008263V	Sara	Ribes	29	admin					
<input type="checkbox"/>	2	Iriss	Irene	Sotos	28	admin					
<input checked="" type="checkbox"/>	3	Jorge	Jorge	Vasco	29	usuario					
<input type="checkbox"/>	5	Raxefon	Ivan	Alonso	30	admin					
<input type="checkbox"/>	6	Sico	Sergio	Alonso Rubio	32	gestor					
<input type="checkbox"/>	8	doc	andres	salazar	30	usuario					
<input type="checkbox"/>	9	Carlos	Espejo	Díaz	24	usuario					

Showing 1 to 7 of 7 entries 1 row selected

Volver atrás

Ilustración 23: Pantalla de administración

En cada fila se mostrará un botón de *checked* que permitirá actualizar o eliminar el usuario seleccionado.

Si se tiene seleccionado a un usuario y se pulsa en el botón de “Actualizar”, se mostrará un modal que permite actualizar el usuario seleccionado ya sea el nombre, apellido, contraseña, edad o role.

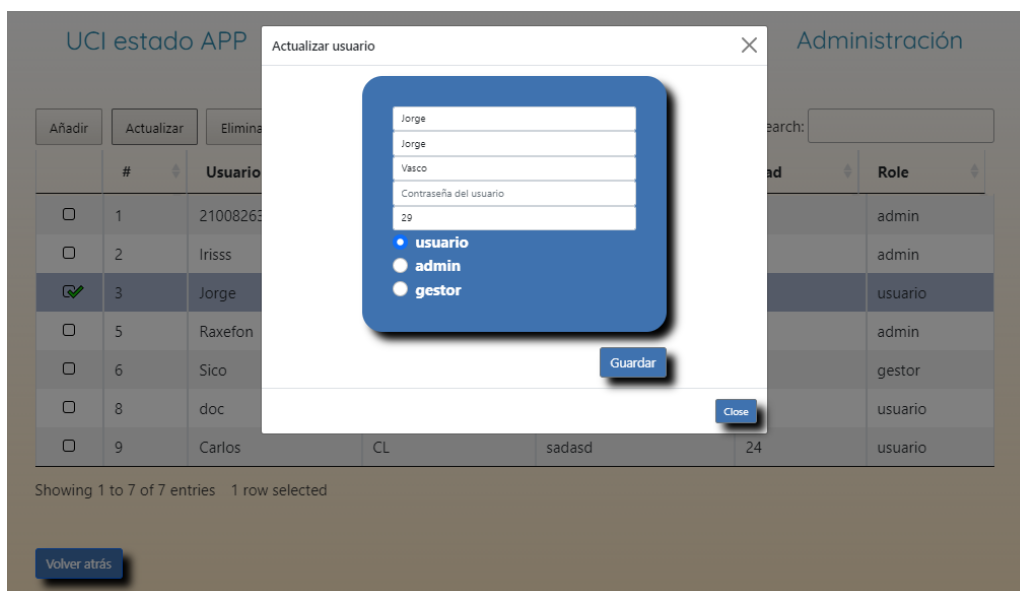


Ilustración 24: : Modal de actualización de un usuario

Aplicación para controlar el estado de pacientes de cuidados intensivos

Una vez modificado los datos se aplicarán dando al botón de “guardar” del modal, actualizando la información en base de datos.



6. Implantación

Para el desarrollo de la aplicación no se ha utilizado ningún tipo de framework. Como se ha comentado se han utilizado varias tecnologías para el desarrollo (*HTML, CSS, JavaScript, PHP, Bootstrap, jQuery, Google charts, AJAX*).

Durante la fase de diseño para mostrar un funcionamiento más intuitivo se optó por usar el conjunto de técnicas desarrollo web *AJAX*, que permite que el intercambio de datos y la capa de presentación funcionen simultáneamente sin que interfieran la una con la otra.

Estas técnicas utilizan *HTML/XHTML* para el lenguaje principal y la presentación. El *DOM* para la visualización dinámica y su interacción. *JSON* para el intercambio de los datos y su manipulación. *XMLHttpRequest* para la comunicación asincrónica y JavaScript para unir todas las tecnologías.

El modelo de *AJAX* viene a ser el siguiente:

1. El navegador crea una llamada de JavaScript que luego activará *XMLHttpRequest*.
2. En segundo plano, el navegador web crea una solicitud HTTP al servidor.
3. El servidor recibe, recupera y envía los datos al navegador web.
4. El navegador web recibe los datos solicitados que aparecerán directamente en la página.
No se necesita recargar.

En el desarrollo de la aplicación casi toda la comunicación que interactúa con la Base de datos *MySQL* lo hemos realizado a través de *AJAX*, para poder usar las ventajas explicadas anteriormente.

Para el procesamiento de las imágenes, se convertirán en datos binarios y estas imágenes no podrán ser superiores a 2Mb.

Para mostrar los gráficos hemos utilizado la librería *Google charts* que a través de los datos en *JSON* nos hace su representación gráfica.

Para mostrar las tablas en la aplicación se ha utilizado *DataTable* que es una librería de *jQuery* con *JavaScript* que agrega características avanzadas a cualquier tabla *HTML*, permitiendo la búsqueda, la ordenación, entre otros aspectos.



Aplicación para controlar el estado de pacientes de cuidados intensivos

Para que todo funcione correctamente se han tenido que insertar los links de enlaces a las librerías de *Bootstrap* y *jQuery*. En el momento de entregar el proyecto, se incluirán los archivos '.min' de dichas librerías para evitar problemas de dependencias.

El desarrollo de la aplicación, al estar desarrollado en *PHP* que no es un lenguaje compilado se entregara todo el proyecto al cliente en su entrega.



7. Pruebas

Al finalizar la implantación de la aplicación se pondrá en funcionamiento en un servidor con datos de muestra, y se le concederá acceso al personal del hospital, junto con una formación, comprobando que la aplicación es sencilla e intuitiva.

Una vez realizado estos pasos, se comenzará a utilizar la aplicación y se confirmará el correcto funcionamiento, comprobando que no existen errores de inconsistencia de datos [15].

Los pasos que se realizar serán:

- 1. Provisión de entornos necesarios:** El Equipo deberá seguir el protocolo necesario para disponer de los entornos requeridos con completa disponibilidad y con la configuración necesaria.
- 2. Obtención de los productos (software y documentación):** El Equipo descargará los productos necesarios para la ejecución de los servicios.
- 3. Verificación del proceso de Compilación:** La entrega es sometida al servicio 'Verificación del proceso de Compilación', basándose en la documentación para la instalación y configuración, y en las fuentes entregadas. Se certifica la compilación del software garantizando la generación de los compilados y la adecuada gestión de dependencias de otras aplicaciones, componentes externos y librerías.
- 4. Verificación del proceso de despliegue:** Este paso tiene como objetivo asegurar que la entrega se podrá implantar en cualquier entorno con las características especificadas, siguiendo las recomendaciones del Manual de Instalación.
- 5. Generación y evolución de planes de prueba:** El Equipo deberá validar y evolucionar los distintos Planes de Prueba (tanto técnicos, como funcionales) que serán puestos en práctica durante la certificación del aplicativo.
- 6. Despliegue desde cero:** En el caso de entregas que necesiten ser desplegadas en un entorno limpio, se procederá a la instalación de la aplicación desde cero en dicho entorno, siguiendo las indicaciones recogidas en la documentación de soporte a la implantación.



- 7. Análisis de defectos detectados:** El Equipo deberá valorar los defectos detectados tras la ejecución de los servicios anteriores, y decidir si continuar la ejecución del resto de los servicios comprometidos o proponer una reentrega. Para ello, se apoyará en los informes de resultados de los servicios que han sido ejecutados y que contienen los defectos detectados (no solucionados).
- 8. Verificación y validación de la usabilidad:** Se medirá la facilidad con la que un usuario inexperto puede utilizar la aplicación.
- 9. Verificación funcional y generación de pruebas de regresión:** Se ejecutarán las pruebas funcionales que aseguren que el producto software se ajusta a las necesidades solicitadas por el usuario.
- 10. Elaboración informe de resultados:** Una vez finalizada la ejecución de cada paso, el equipo deberá elaborar un informe en el que se recojan los resultados de todas las verificaciones realizadas. Una vez elaborado, deberá incorporarlo en la herramienta de gestión documental para que el Gestor de Proyecto pueda conocer los resultados de las pruebas realizadas.

8. Conclusiones

En este proyecto se ha detallado el proceso de diseño e implementación de una aplicación web para ver el estado de la sala de cuidados intensivos, para ayudar a recibir la información relevante de los pacientes, acceso al material complementario y visualización de las pruebas.

De esta manera los médicos (facultativos y residentes) y enfermería dispondrán de la información más precisa de los cambios clínicos de los pacientes. Permitiendo facilitar la organización y el manejo más adecuado de los pacientes.

Se han alcanzado los siguientes:

- ✓ Desarrollo de una aplicación web para la gestión de la sala de cuidados intensivos.
- ✓ Guardar los datos en una base de datos MySQL.
- ✓ Mejorar el intercambio de información de forma segura entre los facultativos y que pueda ser extensible al personal de enfermería.
- ✓ Evitar la pérdida de información de los pacientes.
- ✓ Generar informes con el histórico de la sala de cuidados intensivos para futuros análisis.

Esto permitirá, cuando se ponga en producción el sistema, y hace un uso correcto de ella:

- Mejorar la gestión de la sala de cuidados intensivos, organizando mejor los recursos disponibles.
- Reducir el uso de documentación en papel, evitando la pérdida de documentación y posibles errores.
- Modernizar la sala de cuidados intensivos.

Con esto se ha completado todos los objetivos planteados en el proyecto



9. Trabajos futuros

9.1. Formación

En las primeras etapas de puesta en marcha de la aplicación se realizarán varias sesiones de formación para el personal de la sala de cuidados intensivos.

Esta formación se realizará de manera presencial o en teleformación según sea necesario para el servicio y según las condiciones de aforo lo permitan.

El objetivo es que únicamente con la guía de usuario y la formación a los usuarios de la plataforma se alcance un aprovechamiento óptimo de las funciones de la aplicación.

9.2. Versión de la aplicación

En versiones posteriores de la aplicación se podrá diseñar en la pantalla de informes un acceso para exportar la información de la aplicación según criterios que se seleccionen como, por ejemplo, los mostrados en los gráficos, entre otros. Además, se podrá exportar dicha información en varios tipos de formatos, como PDF, CSV, Excel, etc.

También se ha pensado desarrollar una opción de agregar o quitar camas de la pantalla principal para modificar el tamaño de la sala de cuidados intensivos, según la necesidad de camas en función de la demanda asistencial.

En la aplicación se deberá crear una forma de mantener la sesión. Actualmente cada vez que se ingresa a la aplicación se tiene que poner el usuario y contraseña.



10. Referencias bibliográficas y citas

- [1] D. Fenollar Onrubia, Aplicación móvil para la gestión de pacientes con Mieloma Múltiple, Universitat Politècnica de València., 2019-10-14.
- [2] D. Fenollar Onrubia, Aplicación móvil para la gestión de pacientes con Mieloma Múltiple, Universitat Politècnica de València., 2019-10-14.
- [3] M. Mares García, Aplicacion en un hospital del análisis de procesos sanitarios haciendo uso de tecnologías de minería de procesos, Universitat Politècnica de València., 2017-09-11.
- [4] D. M. Villanueva Castillo, Estudio, análisis e implantación de las medidas Técnicas y Organizativas para el cumplimiento de la Ley Orgánica de Protección de Datos de Carácter Personal (LOPD) 15/1999 en la empresa Cases de Dret 23 S.L, Universitat Politècnica de València, 2010-11-11.
- [5] RGPD: nuevas oportunidades, nuevas obligaciones : lo que las empresas tienen que saber sobre el Reglamento general de protección de datos de la Unión Europea, Luxembourg: Publications Office, 2018.
- [6] B.O.E, Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales., 2018.
- [7] VVAA, Acceso Abierto; Copyleft; Software libre; Licencias libres, Traficantes de Sueños, 2006.
- [8] K. Yank, PHP and MySQL, Collingwood, Victoria, Australia : SitePoint, 2012.
- [9] E. A. Woychowsky, Ajax : Creating Web Pages with Asynchronous JavaScript and XML, Place of publication not identified Prentice Hall, 2007.
- [10] A. Kumar y K. Rugg, Brewers Conjecture and a characterization of the limits, and relationships between Consistency, Availability and Partition Tolerance in a distributed service, 2019.
- [11] Z. Jiménez, Aplicaciones web, Macmillan Iberia, S.A., 2013.
- [12] M. C. Penadés Gramage, Diagrama de clases, Valencia: Universitat Politècnica de València, 2016-07-11.
- [13] M. Gonzalez Gimenez, Diseño Lógico de Bases de Datos, Universitat Politècnica de València, 2011.
- [14] A. H. Martínez, Medicina Intensiva en el Enfermo Crítico, Médica Panamericana, 2019.

[15] I. Sommerville, Ingeniería de Software (9a. Ed.), Pearson Educación, 2011.

[16] L. E. A. Ullman, MySQL, Place of publication not identified Peachpit Press, 2006.

11. Apéndices

Comando SQL para la creación de la base de datos relacional [16].

```
CREATE TABLE `alergias` (  
  
  `idalergias` int(11) NOT NULL,  
  
  `fk_cama` int(11) NOT NULL,  
  
  `fk_paciente` int(11) NOT NULL,  
  
  `contenido` longtext DEFAULT NULL,  
  
  `fecha` timestamp(6) NULL DEFAULT NULL  
  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

--

-- Estructura de tabla para la tabla `antecedentes`

--

```
CREATE TABLE `antecedentes` (  
  
  `idantecedentes` int(11) NOT NULL,  
  
  `fk_cama` int(11) NOT NULL,  
  
  `fk_paciente` int(11) NOT NULL,  
  
  `contenido` longtext DEFAULT NULL,  
  
  `fecha` timestamp(6) NULL DEFAULT NULL  
  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```



Aplicación para controlar el estado de pacientes de cuidados intensivos

```
--  
  
-- Estructura de tabla para la tabla `camas`  
  
--  
  
CREATE TABLE `camas` (  
  `idcamas` int(11) NOT NULL,  
  `libre` bigint(1) DEFAULT 0,  
  `nombre` varchar(45) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;  
  
--  
  
-- Volcado de datos para la tabla `camas`  
  
--  
  
INSERT INTO `camas` (`idcamas`, `libre`, `nombre`) VALUES  
  
(1, 0, NULL),  
  
(2, 0, NULL),  
  
(3, 0, NULL),  
  
(4, 0, NULL),  
  
(5, 0, NULL),  
  
(6, 0, NULL),  
  
(7, 0, NULL),  
  
(8, 0, NULL),  
  
(9, 0, NULL),  
  
(10, 0, NULL),
```




```
(11, 0, NULL),  
  
(12, 0, NULL),  
  
(13, 0, NULL),  
  
(14, 0, NULL),  
  
(15, 0, NULL),  
  
(16, 0, NULL);
```

--

-- Estructura de tabla para la tabla `categorias`

--

```
CREATE TABLE `categorias` (  
  
  `idcategorias` int(11) NOT NULL,  
  
  `nombre` varchar(45) NOT NULL,  
  
  `tipo` int(11) NOT NULL  
  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

--

-- Volcado de datos para la tabla `categorias`

--

```
INSERT INTO `categorias` (`idcategorias`, `nombre`, `tipo`) VALUES  
  
(1, 'Situación actual', 1),  
  
(2, 'Antecedentes personales', 2),
```



Aplicación para controlar el estado de pacientes de cuidados intensivos

```
(3, 'Alergias', 2),

(4, 'Analítica', 3),

(5, 'Gasometría', 3),

(6, 'ECG', 3),

(7, 'Rx tórax', 3),

(8, 'Microbiología', 3),

(9, 'Otros', 3);

-----

--

-- Estructura de tabla para la tabla `categorias_patologias`

--

CREATE TABLE `categorias_patologias` (

  `idcategorias_patologias` int(11) NOT NULL,

  `nombre` varchar(45) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

-- Volcado de datos para la tabla `categorias_patologias`

--

INSERT INTO `categorias_patologias` (`idcategorias_patologias`, `nombre`)

VALUES

(1, 'Coronarios'),

(2, 'Sépticos'),

(3, 'Miscelánea'),

(4, 'Covid19');
```



```

-----
--
-- Estructura de tabla para la tabla `info_sistemas`
--

CREATE TABLE `info_sistemas` (
  `idinfo_sistemas` int(11) NOT NULL,
  `fk_sistema` int(11) NOT NULL,
  `fk_cama` int(11) NOT NULL,
  `fk_paciente` int(11) NOT NULL,
  `contenido` longtext DEFAULT NULL,
  `fecha` timestamp(6) NULL DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
-----

--
-- Estructura de tabla para la tabla `mensaje`
--

CREATE TABLE `mensaje` (
  `idmensaje` int(11) NOT NULL,
  `idusuario` int(11) NOT NULL,
  `idsala` int(11) NOT NULL,
  `mensaje` longtext DEFAULT NULL,
  `fecha` timestamp(6) NOT NULL DEFAULT current_timestamp(6) ON UPDATE
current_timestamp(6)

```



Aplicación para controlar el estado de pacientes de cuidados intensivos

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- -----
--
-- Estructura de tabla para la tabla `pacientes`
--

CREATE TABLE `pacientes` (
  `idpacientes` int(11) NOT NULL,
  `nombre` varchar(45) NOT NULL,
  `edad` int(3) DEFAULT NULL,
  `categoria_patologia` int(11) DEFAULT NULL,
  `patologia` int(11) DEFAULT NULL,
  `entrada` timestamp(6) NULL DEFAULT current_timestamp(6),
  `salida` timestamp(6) NULL DEFAULT NULL,
  `alta` varchar(45) DEFAULT '0',
  `motivo_alta` varchar(45) DEFAULT NULL,
  `cama` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- -----
--
-- Estructura de tabla para la tabla `patologias`
--

CREATE TABLE `patologias` (
```



```

`idpatologias` int(11) NOT NULL,

`nombre` varchar(45) NOT NULL,

`descripcion` varchar(45) DEFAULT NULL,

`categoria` int(11) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-----

--

-- Estructura de tabla para la tabla `pruebas_diagnosticas`

--

CREATE TABLE `pruebas_diagnosticas` (

  `idpruebas_diagnosticas` int(11) NOT NULL,

  `fk_categoria` int(11) NOT NULL,

  `fk_cama` int(11) NOT NULL,

  `fk_paciente` int(11) NOT NULL,

  `titulo` varchar(45) CHARACTER SET utf8mb4 DEFAULT NULL,

  `descripcion` text CHARACTER SET utf8mb4 DEFAULT NULL,

  `imagen` longblob DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-----

--

-- Estructura de tabla para la tabla `sala`

--

```



Aplicación para controlar el estado de pacientes de cuidados intensivos

```
CREATE TABLE `sala` (  
  
  `idsala` int(11) NOT NULL,  
  
  `idtipo` int(11) NOT NULL,  
  
  `nombre` varchar(45) NOT NULL,  
  
  `fecha` datetime(6) NOT NULL  
  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;  
  
-----  
  
--  
  
-- Estructura de tabla para la tabla `sala_usuario`  
  
--  
  
CREATE TABLE `sala_usuario` (  
  
  `idsala_usuario` int(11) NOT NULL,  
  
  `idsala` int(11) NOT NULL,  
  
  `idusuario` int(11) NOT NULL  
  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;  
  
-----  
  
--  
  
-- Estructura de tabla para la tabla `sistemas`  
  
--  
  
CREATE TABLE `sistemas` (  
  
  `idsistemas` int(11) NOT NULL,  
  
  `nombre` varchar(45) NOT NULL,
```

```

        `tipo` int(45) DEFAULT 0

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

-- Volcado de datos para la tabla `sistemas`

--

INSERT INTO `sistemas` (`idsistemas`, `nombre`, `tipo`) VALUES

(1, 'Neurológico', 1),

(2, 'Hemodinámico', 1),

(3, 'Respiratorio', 1),

(4, 'Renal', 1),

(5, 'Infeccioso', 1),

(6, 'Abdomen/Metabólico', 1),

(7, 'Otros', 1);

-----

--

-- Estructura de tabla para la tabla `tipo_mensaje`

--

CREATE TABLE `tipo_mensaje` (

  `idtipo_mensaje` int(11) NOT NULL,

  `nombre` varchar(45) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

```



Aplicación para controlar el estado de pacientes de cuidados intensivos

```
-- Volcado de datos para la tabla `tipo_mensaje`  
  
--  
  
INSERT INTO `tipo_mensaje` (`idtipo_mensaje`, `nombre`) VALUES  
  
(1, 'mensaje'),  
  
(2, 'video'),  
  
(3, 'audio'),  
  
(4, 'imagen');  
  
-----  
  
--  
  
-- Estructura de tabla para la tabla `tipo_sala`  
  
--  
  
CREATE TABLE `tipo_sala` (  
  
  `idtipo_sala` int(11) NOT NULL,  
  
  `nombre` varchar(45) NOT NULL  
  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;  
  
--  
  
-- Volcado de datos para la tabla `tipo_sala`  
  
--  
  
INSERT INTO `tipo_sala` (`idtipo_sala`, `nombre`) VALUES  
  
(1, 'privada'),  
  
(2, 'grupo');
```




```

-----
--
-- Estructura de tabla para la tabla `users`
--

CREATE TABLE `users` (
  `idusers` int(11) NOT NULL,
  `username` varchar(45) NOT NULL,
  `password` varchar(60) NOT NULL,
  `name` varchar(45) DEFAULT NULL,
  `surname` varchar(45) DEFAULT NULL,
  `age` int(3) DEFAULT NULL,
  `role` varchar(45) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

-- Índices para tablas volcadas
--

--

-- Indices de la tabla `alergias`
--

ALTER TABLE `alergias`

  ADD PRIMARY KEY (`idalergias`),

```



Aplicación para controlar el estado de pacientes de cuidados intensivos

```
ADD KEY `fk_cama` (`fk_cama`),

ADD KEY `fk_paciente` (`fk_paciente`);

--

-- Indices de la tabla `antecedentes`

--

ALTER TABLE `antecedentes`

ADD PRIMARY KEY (`idantecedentes`),

ADD KEY `fk_cama` (`fk_cama`),

ADD KEY `fk_paciente` (`fk_paciente`);

--

-- Indices de la tabla `camas`

--

ALTER TABLE `camas`

ADD PRIMARY KEY (`idcamas`);

--

-- Indices de la tabla `categorias`

--

ALTER TABLE `categorias`

ADD PRIMARY KEY (`idcategorias`);

--

-- Indices de la tabla `categorias_patologias`
```



```
--  
  
ALTER TABLE `categorias_patologias`  
  
    ADD PRIMARY KEY (`idcategorias_patologias`);
```

```
--  
  
-- Indices de la tabla `info_sistemas`  
  
--
```

```
ALTER TABLE `info_sistemas`  
  
    ADD PRIMARY KEY (`idinfo_sistemas`),  
  
    ADD KEY `fk_sistema` (`fk_sistema`),  
  
    ADD KEY `fk_cama` (`fk_cama`),  
  
    ADD KEY `fk_paciente` (`fk_paciente`);
```

```
--  
  
-- Indices de la tabla `mensaje`  
  
--
```

```
ALTER TABLE `mensaje`  
  
    ADD PRIMARY KEY (`idmensaje`),  
  
    ADD KEY `idusuario` (`idusuario`),  
  
    ADD KEY `idsala` (`idsala`);
```

```
--  
  
-- Indices de la tabla `pacientes`  
  
--
```

```
ALTER TABLE `pacientes`
```



Aplicación para controlar el estado de pacientes de cuidados intensivos

```
ADD PRIMARY KEY (`idpacientes`),

ADD KEY `cama` (`cama`),

ADD KEY `patologia` (`patologia`),

ADD KEY `categoria_patologia` (`categoria_patologia`);

--

-- Indices de la tabla `patologias`

--

ALTER TABLE `patologias`

    ADD PRIMARY KEY (`idpatologias`),

    ADD KEY `categoria` (`categoria`);

--

-- Indices de la tabla `pruebas_diagnosticas`

--

ALTER TABLE `pruebas_diagnosticas`

    ADD PRIMARY KEY (`idpruebas_diagnosticas`),

    ADD KEY `fk_categoria` (`fk_categoria`),

    ADD KEY `fk_cama` (`fk_cama`),

    ADD KEY `fk_paciente` (`fk_paciente`);

--

-- Indices de la tabla `sala`

--

ALTER TABLE `sala`
```



```

ADD PRIMARY KEY (`idsala`),

ADD KEY `idtipo` (`idtipo`);

--

-- Indices de la tabla `sala_usuario`

--

ALTER TABLE `sala_usuario`

ADD PRIMARY KEY (`idsala_usuario`),

ADD KEY `idsala` (`idsala`),

ADD KEY `idusuario` (`idusuario`);

--

-- Indices de la tabla `sistemas`

--

ALTER TABLE `sistemas`

ADD PRIMARY KEY (`idsistemas`);

--

-- Indices de la tabla `tipo_mensaje`

--

ALTER TABLE `tipo_mensaje`

ADD PRIMARY KEY (`idtipo_mensaje`);

--

-- Indices de la tabla `tipo_sala`

```



Aplicación para controlar el estado de pacientes de cuidados intensivos

```
--  
  
ALTER TABLE `tipo_sala`  
  
    ADD PRIMARY KEY (`idtipo_sala`);  
  
--  
  
-- Indices de la tabla `users`  
  
--  
  
ALTER TABLE `users`  
  
    ADD PRIMARY KEY (`idusers`);  
  
--  
  
-- AUTO_INCREMENT de las tablas volcadas  
  
--  
  
--  
  
-- AUTO_INCREMENT de la tabla `alergias`  
  
--  
  
ALTER TABLE `alergias`  
  
    MODIFY `idalergias` int(11) NOT NULL AUTO_INCREMENT;  
  
--  
  
-- AUTO_INCREMENT de la tabla `antecedentes`  
  
--  
  
ALTER TABLE `antecedentes`
```



```

MODIFY `idantecedentes` int(11) NOT NULL AUTO_INCREMENT;

--

-- AUTO_INCREMENT de la tabla `camas`

--

ALTER TABLE `camas`

MODIFY `idcamas` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=17;

--

-- AUTO_INCREMENT de la tabla `categorias`

--

ALTER TABLE `categorias`

MODIFY `idcategorias` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=10;

--

-- AUTO_INCREMENT de la tabla `categorias_patologias`

--

ALTER TABLE `categorias_patologias`

MODIFY `idcategorias_patologias` int(11) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=5;

--

-- AUTO_INCREMENT de la tabla `info_sistemas`

--

ALTER TABLE `info_sistemas`

```



Aplicación para controlar el estado de pacientes de cuidados intensivos

```
MODIFY `idinfo_sistemas` int(11) NOT NULL AUTO_INCREMENT;

--

-- AUTO_INCREMENT de la tabla `mensaje`

--

ALTER TABLE `mensaje`

MODIFY `idmensaje` int(11) NOT NULL AUTO_INCREMENT;

--

-- AUTO_INCREMENT de la tabla `pacientes`

--

ALTER TABLE `pacientes`

MODIFY `idpacientes` int(11) NOT NULL AUTO_INCREMENT;

--

-- AUTO_INCREMENT de la tabla `patologias`

--

ALTER TABLE `patologias`

MODIFY `idpatologias` int(11) NOT NULL AUTO_INCREMENT;

--

-- AUTO_INCREMENT de la tabla `pruebas_diagnosticas`

--

ALTER TABLE `pruebas_diagnosticas`
```




```

MODIFY `idpruebas_diagnosticas` int(11) NOT NULL AUTO_INCREMENT;

--

-- AUTO_INCREMENT de la tabla `sala`

--

ALTER TABLE `sala`

MODIFY `idsala` int(11) NOT NULL AUTO_INCREMENT;

--

-- AUTO_INCREMENT de la tabla `sala_usuario`

--

ALTER TABLE `sala_usuario`

MODIFY `idsala_usuario` int(11) NOT NULL AUTO_INCREMENT;

--

-- AUTO_INCREMENT de la tabla `sistemas`

--

ALTER TABLE `sistemas`

MODIFY `idsistemas` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=8;

--

-- AUTO_INCREMENT de la tabla `tipo_mensaje`

--

ALTER TABLE `tipo_mensaje`

MODIFY `idtipo_mensaje` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=5;

```



Aplicación para controlar el estado de pacientes de cuidados intensivos

```
--  
  
-- AUTO_INCREMENT de la tabla `users`  
  
--  
  
ALTER TABLE `users`  
  
    MODIFY `idusers` int(11) NOT NULL AUTO_INCREMENT;  
  
--  
  
-- Restricciones para tablas volcadas  
  
--  
  
--  
  
-- Filtros para la tabla `alergias`  
  
--  
  
ALTER TABLE `alergias`  
  
    ADD CONSTRAINT `alergias_ibfk_1` FOREIGN KEY (`fk_paciente`) REFERENCES  
`pacientes` (`idpacientes`);  
  
--  
  
-- Filtros para la tabla `antecedentes`  
  
--  
  
ALTER TABLE `antecedentes`  
  
    ADD CONSTRAINT `antecedentes_ibfk_1` FOREIGN KEY (`fk_paciente`) REFERENCES  
`pacientes` (`idpacientes`),  
  
    ADD CONSTRAINT `antecedentes_ibfk_2` FOREIGN KEY (`fk_cama`) REFERENCES  
`pacientes` (`cama`) ON DELETE CASCADE ON UPDATE CASCADE;
```



```

--

-- Filtros para la tabla `info_sistemas`

--

ALTER TABLE `info_sistemas`

    ADD CONSTRAINT `info_sistemas_ibfk_1` FOREIGN KEY (`fk_paciente`) REFERENCES
`pacientes` (`idpacientes`),

    ADD CONSTRAINT `info_sistemas_ibfk_2` FOREIGN KEY (`fk_cama`) REFERENCES
`pacientes` (`cama`) ON DELETE CASCADE ON UPDATE CASCADE,

    ADD CONSTRAINT `info_sistemas_ibfk_3` FOREIGN KEY (`fk_sistema`) REFERENCES
`sistemas` (`idsistemas`) ON DELETE CASCADE ON UPDATE CASCADE;

--

-- Filtros para la tabla `mensaje`

--

ALTER TABLE `mensaje`

    ADD CONSTRAINT `mensaje_ibfk_1` FOREIGN KEY (`idusuario`) REFERENCES `users`
(`idusers`),

    ADD CONSTRAINT `mensaje_ibfk_2` FOREIGN KEY (`idsala`) REFERENCES `sala`
(`idsala`) ON DELETE CASCADE ON UPDATE CASCADE;

--

-- Filtros para la tabla `pacientes`

--

ALTER TABLE `pacientes`

    ADD CONSTRAINT `pacientes_ibfk_1` FOREIGN KEY (`cama`) REFERENCES `camas`
(`idcamas`),

```



Aplicación para controlar el estado de pacientes de cuidados intensivos

```
ADD CONSTRAINT `pacientes_ibfk_2` FOREIGN KEY (`patologia`) REFERENCES
`patologias` (`idpatologias`) ON DELETE CASCADE ON UPDATE CASCADE,
```

```
ADD CONSTRAINT `pacientes_ibfk_3` FOREIGN KEY (`categoria_patologia`)
REFERENCES `patologias` (`categoria`) ON DELETE CASCADE ON UPDATE CASCADE;
```

```
--
```

```
-- Filtros para la tabla `patologias`
```

```
--
```

```
ALTER TABLE `patologias`
```

```
ADD CONSTRAINT `patologias_ibfk_1` FOREIGN KEY (`categoria`) REFERENCES
`categorias_patologias` (`idcategorias_patologias`);
```

```
--
```

```
-- Filtros para la tabla `pruebas_diagnosticas`
```

```
--
```

```
ALTER TABLE `pruebas_diagnosticas`
```

```
ADD CONSTRAINT `pruebas_diagnosticas_ibfk_1` FOREIGN KEY (`fk_cama`)
REFERENCES `pacientes` (`cama`) ON DELETE CASCADE ON UPDATE CASCADE,
```

```
ADD CONSTRAINT `pruebas_diagnosticas_ibfk_2` FOREIGN KEY (`fk_paciente`)
REFERENCES `pacientes` (`idpacientes`) ON DELETE CASCADE ON UPDATE CASCADE,
```

```
ADD CONSTRAINT `pruebas_diagnosticas_ibfk_3` FOREIGN KEY (`fk_categoria`)
REFERENCES `categorias` (`idcategorias`) ON DELETE CASCADE ON UPDATE CASCADE;
```

```
--
```

```
-- Filtros para la tabla `sala`
```

```
--
```

```
ALTER TABLE `sala`
```



```
ADD CONSTRAINT `sala_ibfk_1` FOREIGN KEY (`idtipo`) REFERENCES `tipo_sala`
(`idtipo_sala`) ON DELETE CASCADE ON UPDATE CASCADE;

--

-- Filtros para la tabla `sala_usuario`

--

ALTER TABLE `sala_usuario`

ADD CONSTRAINT `sala_usuario_ibfk_1` FOREIGN KEY (`idusuario`) REFERENCES
`users` (`idusers`) ON DELETE CASCADE ON UPDATE CASCADE,

ADD CONSTRAINT `sala_usuario_ibfk_2` FOREIGN KEY (`idsala`) REFERENCES `sala`
(`idsala`) ON DELETE CASCADE ON UPDATE CASCADE;

COMMIT;
```



12. Glosario

- **UCI:** Unidades de Cuidados Intensivos.
- **LOPD:** Ley Orgánica de Protección de Datos Personales y garantía de los derechos digitales.
- **RGPD:** Reglamento Europeo General de Protección de Datos.
- **HTML:** del inglés HyperText Markup Language, es un lenguaje de marcado que se utiliza para el desarrollo de páginas de Internet.
- **CSS:** del inglés Cascading Style Sheets, es un lenguaje para la composición y estructuración de páginas web.
- **PHP:** del inglés Hypertext Pre-Processor, es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.
- **MySQL:** del inglés My Structured Query Language o Lenguaje de Consulta Estructurado.
- **AJAX:** del inglés Asynchronous JavaScript and XML, es una técnica de desarrollo web para crear aplicaciones web asíncronas.
- **Framework:** es un marco o esquema de trabajo utilizado para realizar el desarrollo de software.
- **Wireframing:** guía visual que representa la estructura esquelética de un sitio web.
- **XMLHttpRequest:** del inglés Extensible Markup Language / Hypertext Transfer Protocol, es una interfaz empleada para realizar peticiones HTTP y HTTPS a servidores Web.



Aplicación para controlar el estado de pacientes de cuidados intensivos

- **Éxito:** es un término latino significa «salida» y se emplea en medicina como simplificación de la expresión más correcta exitus letalis, que literalmente significa «salida mortal» o más médicamente «proceso hacia la muerte».