Óscar Tarazaga Gálvez

Documentación del proyecto

Proyecto ASIR curso 22/23 IES Triana

Índice

[**Introducción**: 2](#_Toc136186353)

[**1º. ¿De dónde surge esta idea?** 2](#_Toc136186354)

[**2º. Idea general.** 2](#_Toc136186355)

[**3º. Herramientas necesarias para el proyecto.** 5](#_Toc136186356)

[**3.1º. Instalación del software.** 6](#_Toc136186357)

[**4º. Desarrollo del proyecto.** 10](#_Toc136186358)

[**4.1º. Desarrollo del apartado del médico.** 12](#_Toc136186359)

[**4.2º. Desarrollo del apartado del paciente.** 17](#_Toc136186360)

[**4.3º. Desarrollo del apartado del administrador.** 21](#_Toc136186361)

[**4.4º. Diseño de la base de datos.** 26](#_Toc136186362)

[**Problemas encontrados durante el desarrollo:** 27](#_Toc136186363)

[**¿Cómo expandiría esta idea?** 27](#_Toc136186364)

[**Conclusión:** 28](#_Toc136186365)

[**Bibliografía**: 28](#_Toc136186366)

# **Introducción**:

En este proyecto voy a mostrar el proceso y evolución sobre el proyecto, aparte de detallar las herramientas que usaré durante la construcción del mismo.

Este proyecto se basa en una versión simple en un entorno web de clicsalud (Aplicación medica de la Junta de Andalucía), en la cual no solo tendremos la perspectiva del paciente, sino también la del doctor y el administrador.

Este proyecto, lo tomaré como que es a pequeña escala, es decir, como si lo implantase en un solo centro médico y no en todos los de la comunidad autónoma como era la idea original.

# **1º. ¿De dónde surge esta idea?**

Esta idea surge desde hace un tiempo, ya que desde los últimos tiempos de la pandemia ha habido un incremento en el uso de las aplicaciones médicas que usa cada comunidad autónoma.

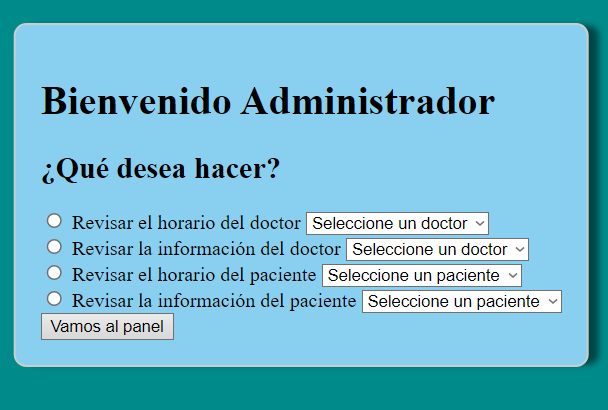
Desde mi punto de vista, este tipo de aplicaciones, tanto la versión web como la versión de móvil, suelen dar fallo, por lo que se me ocurrió el intentar replicar de alguna manera el funcionamiento de este tipo de aplicaciones, ya que me daba bastante curiosidad, por ello, desde que me surgió este pensamiento, estuve dándole vueltas a la idea de ver si yo podía implementar alguna mejora, lo cual voy a tratar de hacer en este proyecto.

# **2º. Idea general.**

Para este proyecto la idea preconcebida que tengo es un entorno web, en el cual, al logarse el usuario, el propio sistema, detecta que tipo de usuario es, en este caso, los tipos solo son 3: Usuario paciente, usuario doctor y usuario administrador.

Cada tipo tendrá su propio panel, es decir, el usuario administrador, por ejemplo, desde el entorno web podrá visualizar los datos tanto de los doctores como de los pacientes, pudiendo hacer algunos cambios, por ejemplo: Uno de los doctores del centro médico, se pre jubila y recibe una reducción de horario, entonces el administrador desde el entorno web podrá modificar la tabla de doctores, diciendo así que el doctor Jiménez cuyo horario era de 08:00 hasta las 18:00 pasa a ser de 08:00 hasta las 12:00.

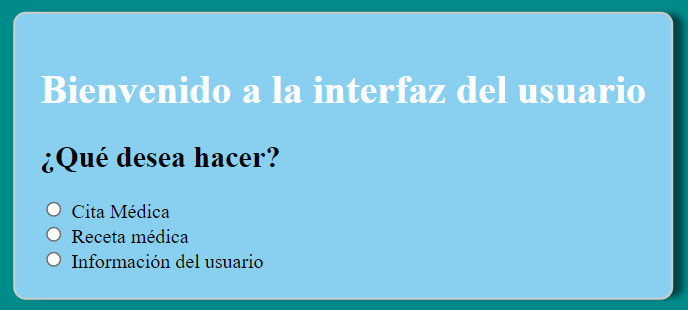
La idea es que cada tipo de usuario tenga varias opciones a realizar, ya sea, siendo administrador, paciente o doctor.



*Imagen de idea preliminar del panel del administrador.*

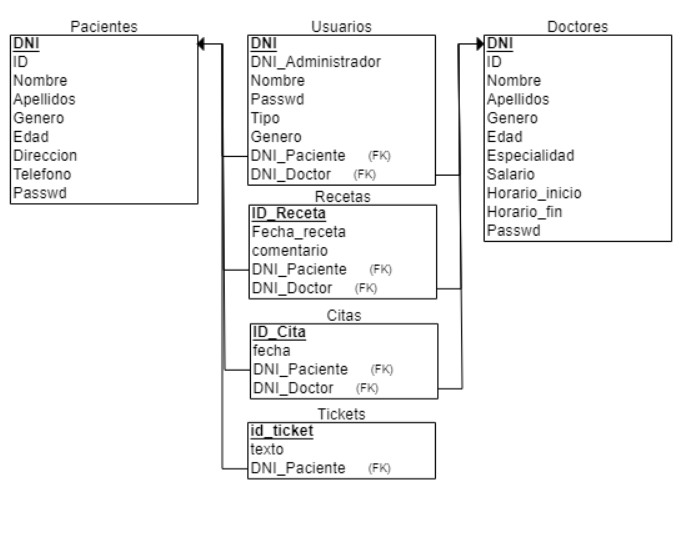


*Imagen de idea preliminar del panel del doctor.*



*Imagen de idea preliminar del panel del paciente.*

Toda esta interfaz web estará conectada con una base de datos, la cual estará documentada tanto en SQL como en su forma de esquema relacional.



*Imagen del esquema relacional de la base de datos*

La base de datos irá en otro archivo en formato SQL.

# **3º. Herramientas necesarias para el proyecto.**

Para este proyecto vamos a usar varias herramientas tanto hardware como software, en mi caso, de hardware la única plataforma que usaré será mi equipo, el cual tiene las siguientes características:

-CPU: Intel Core i5-7300HQ

-RAM: 24 GB

-Almacenamiento: 1 TB en M.2

-Tarjeta gráfica: Nvidia 1050ti

-Placa base: Asus GL553VD

-Sistema operativo: Windows 10 pro

Con esto podemos pasar a hablar del software.

Para el desarrollo del código html, css, php y JS usare Visual Studio Code y para la base de datos, MYSQL Workbench, ya que son 2 aplicaciones que conozco bien como para sentirme cómodo para usarlas durante el proceso de desarrollo, además de poderle instalar plugins los cuales me ayudaran durante todo el proceso.

Para la visualización de las diferentes páginas desarrolladas usare XAMPP, del cual solo necesito php, pero al estar trabajando desde Windows es más sencillo usar esta aplicación directamente.

Para la segunda parte del proyecto en la cual trabajaré con contenedores en una máquina virtual de Ubuntu, la cual será manejada desde VMware.

Esta máquina virtual tendrá instalado Docker y Kubernetes, siendo este último el que usaré ya que a partir de los yamls para definir los pods es más cómodo el definir el funcionamiento de cada contenedor.

Como herramientas externas me beneficiaré de la aplicación web Erdplus, la cual permite el diseño de una base de datos desde modelo entidad/relación a SQL pasando por esquema relacional.

Por último, para la gestión de las tareas que tengo que realizar usare trello, y para el control de versiones de la aplicación usaré git hub.

[Enlace de GitHub aquí.](https://github.com/OscarTarazaga/ProyectoFCT)

## **3.1º. Instalación del software.**

En este apartado explicare como se han instalado las herramientas que se usarán durante todo el proceso de desarrollo.

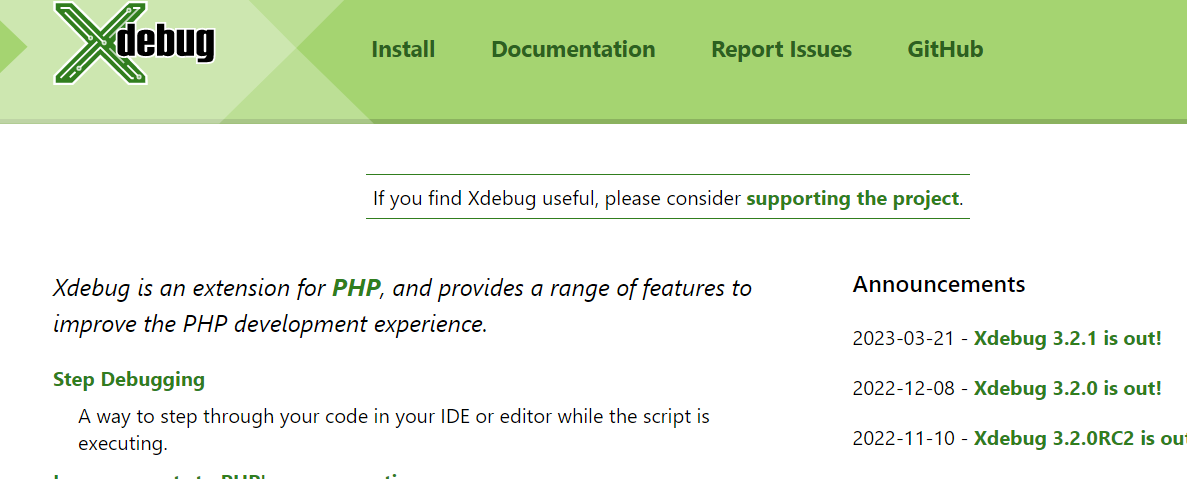
Por ello comenzaremos con MySQL Workbench, el cual es una herramienta visual de diseño y administración de bases de datos.

Este programa es muy sencillo de instalar, ya que tiene los propios ejecutables dispuestos por parte de Microsoft, por lo que solo hay que ejecutar y dar siguiente y siguiente, hasta definir el nombre del usuario que usará esta herramienta, su contraseña y que puerto usará, en este caso el usuario y contraseña son root y el puerto es el 3306, que es el por defecto de mysql.

También se ha instalado Visual Studio Code, el cual es un entorno de desarrollo el cual permite varios tipos de lenguajes, pero para lo que nos atañe, usaré html, css, js y php.

Este programa tiene varios plugins instalados, entre los que se encuentra xdebug, el cual permite de forma eficiente el poder depurar código de manera sencilla.

Este plugin se descarga primero desde la página oficial.



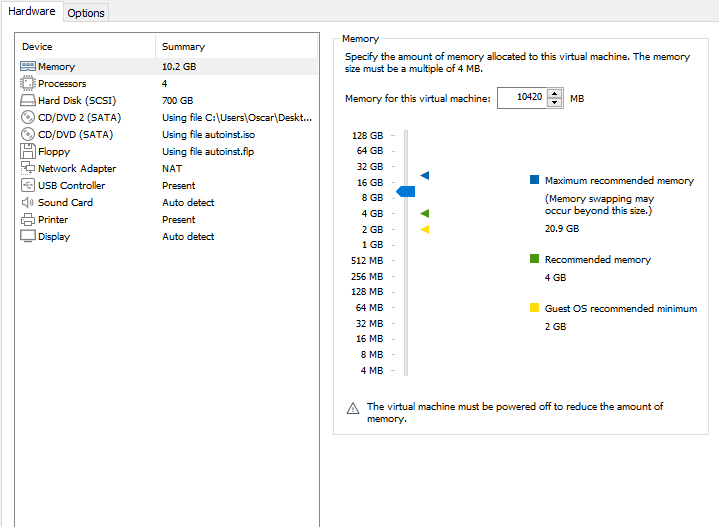
Este fichero con formato .dll se guarda en la ruta en la que este xampp, concretamente dentro de \php\ext y se le renombra como php\_xdebug.dll y se añade la línea Zend\_extension = xdebug, tras esto se reinicia el servicio de apache y con esto estaría instalado, para usarlo desde VScode, seleccionamos primero la opción de depurar, desde la cual se nos abrirá un desplegable para seleccionar el lenguaje que queremos depurar tras esto escribiremos lo siguiente.



En vez de doble contra barra, con una sola bastará y ya estaría el entorno listo.

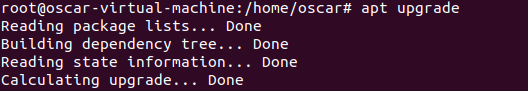
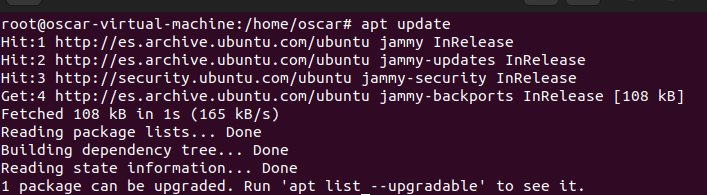
Por último, para la máquina virtual que contendrá los contenedores necesarios para levantar la página web, usaremos VMware Workstation pro, de la cual he adquirido una licencia ya que es una herramienta muy útil para la virtualización.

Primero crearemos una máquina virtual de Ubuntu con estas características:



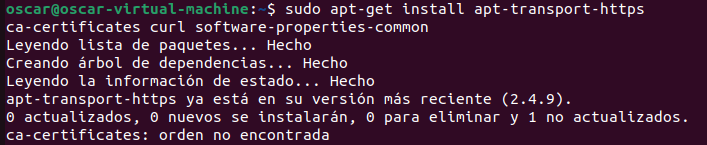
Esta máquina virtual dispondrá de Docker y Kubernetes, ya que son los gestores de contenedores que necesitamos, Docker es muy sencillo de instalar ya que son pocos pasos.

1º. Primero hacemos un update y un upgrade de la máquina.



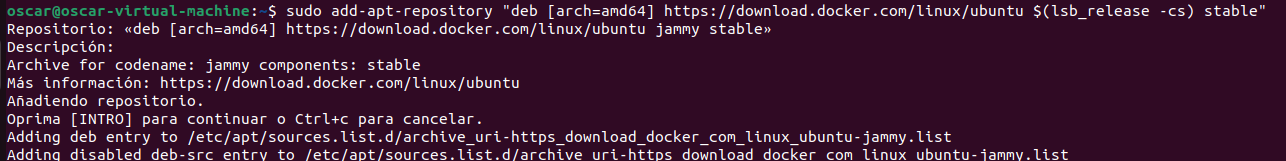
2º. Luego hacemos este comado.

sudo apt-get install \  
ca-certificates \  
curl \  
gnupg \  
lsb-release

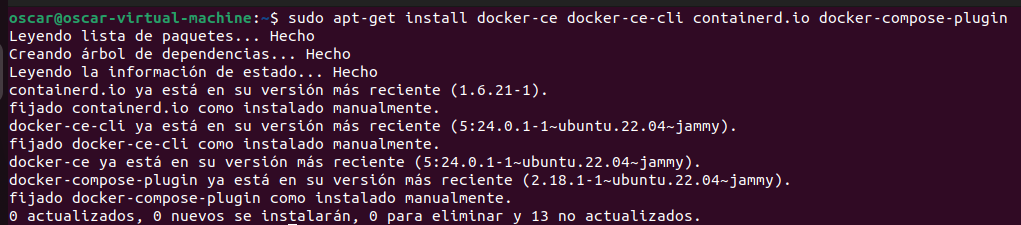


3º. Tras esto hacemos un curl con la siguiente ruta:  


4º. Nos bajamos el repositorio de Docker.



5º. Instalamos Docker-ce y algunos paquetes más de Docker.

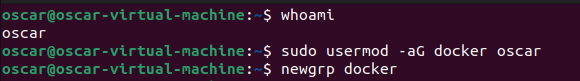


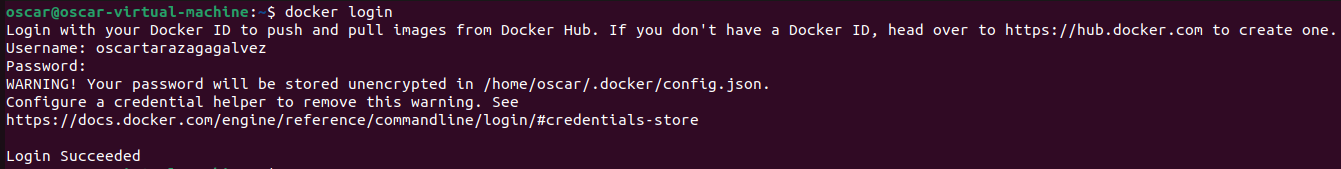
6º. Miramos la versión instalada

7º. Habilitamos Docker.



8º. Hacemos esto para que no haga falta poner sudo cada vez que ejecutamos alguna orden de Docker.





Con esto hecho, ya podemos instalar kubernetes.

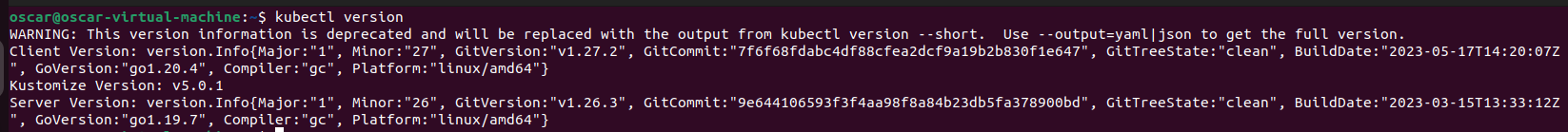
1º. Primero haremos un curl.



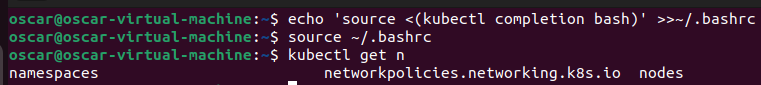
2º. Luego haremos un install del kubectl.



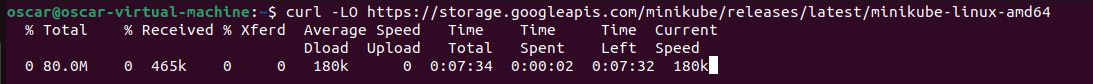
3º. Con esto ya podemos ver la versión instalada.

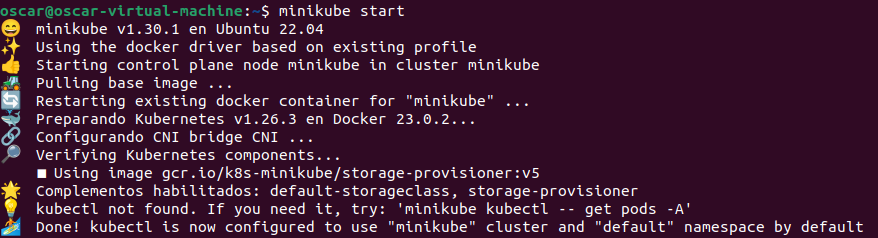


4º. Y por último añadimos kubectl al autocomplete.



5º. Tras esto instalamos minikube y lo iniciamos, esto lo hace automáticamente con docker.





# **4º. Desarrollo del proyecto.**

En la primera parte del desarrollo, empecé con un FrontEnd básico hecho con html y css, el cual está representado de la siguiente manera.



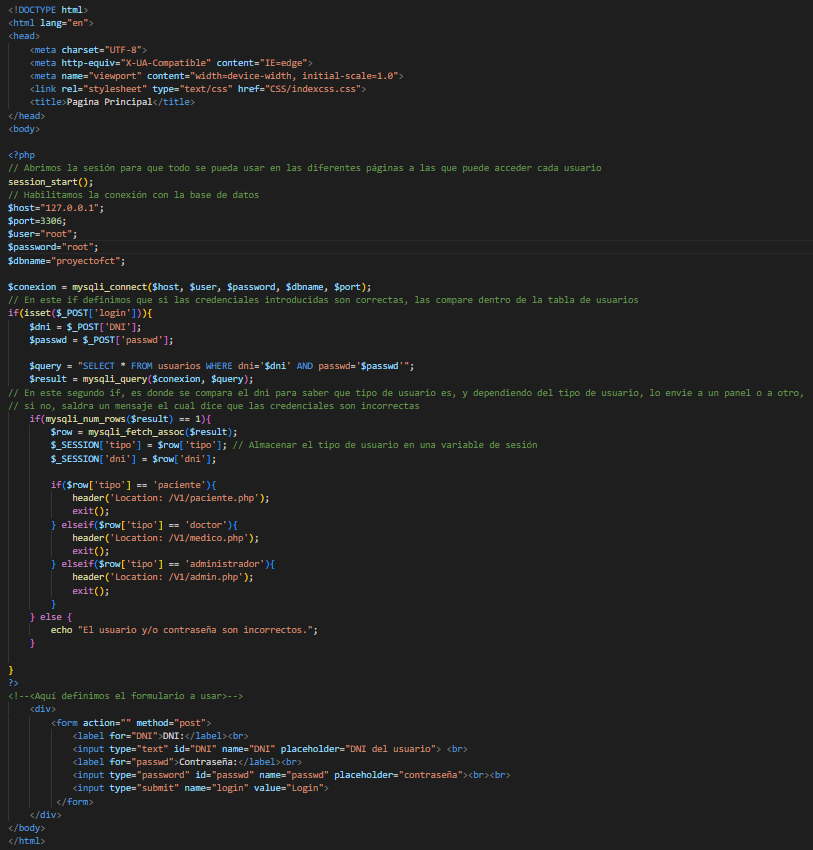
*Imagen de idea preliminar del LogIn.*

En esta primera versión solo tenemos un div el cual contiene el formulario para el sistema de LogIn el cual detecta que tipo de usuario es mediante el DNI.

Este código en php, mostrado a continuación, es el encargado de comparar el DNI introducido, y dependiendo de que usuario sea el DNI le llevará a un panel u otro, también si el usuario se equivoca con la contraseña o el DNI introducido saltará un mensaje de error.

Todo el código desarrollado estará comentado de manera clara dentro del propio fichero con extensión “.php”.

Como podemos ver, hacemos uso de “*IFs*” cuya finalidad es la comparación del dni y la comparación del tipo de usuario que intenta acceder, también podemos observar el uso de headers que son para la redirección a cada página dependiendo del usuario y podemos distinguir el formulario hecho en html y la conexión con la base de datos.

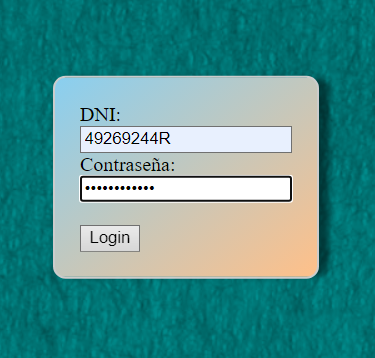


*Imagen del código del LogIn.*

A este código, también se le añadirán trazas de lenguaje JavaScript, esto para la finalidad de que el navegador muestre como ventana emergente el error de que se han introducido erróneamente las credenciales.

## **4.1º. Desarrollo del apartado del médico.**

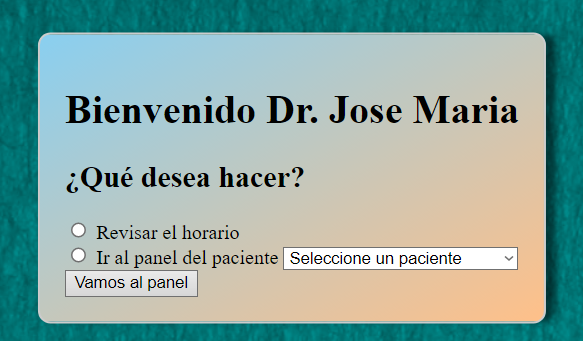
En este apartado explicare el proceso de desarrollo de los paneles/páginas que conforman las opciones que puede realizar el médico.



*Imagen del LogIn.*

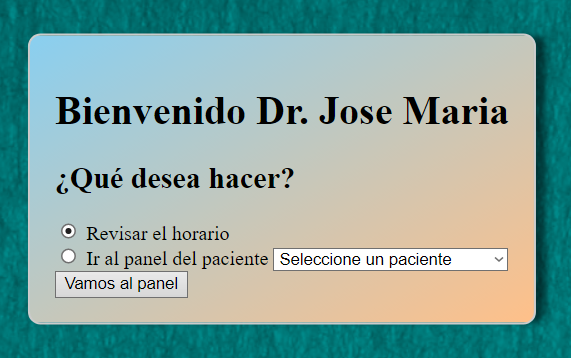
Para empezar al loguearnos como doctor, accederemos al siguiente panel, el cual tiene 2 opciones que son, revisar el horario del doctor para ver que citas tiene asignadas cada día durante la semana actual, y la selección del paciente, desde el cual podrá ver el motivo de la cita, la información del paciente y podrá hacer una receta.

Como podemos ver, al entrar la página usando el dni introducido anteriormente, nos saca el nombre del doctor y si es doctor o doctora.



*Imagen del panel del doctor.*

Si seleccionamos la opción de revisar el horario, nos llevara hasta la página donde seleccionando el campo de horario\_inicio y horario\_fin de la tabla de doctores, nos saca una tabla la cual se genera de lunes a viernes con el horario del doctor, mostrando en el futuro en que horas hay citas asignadas.



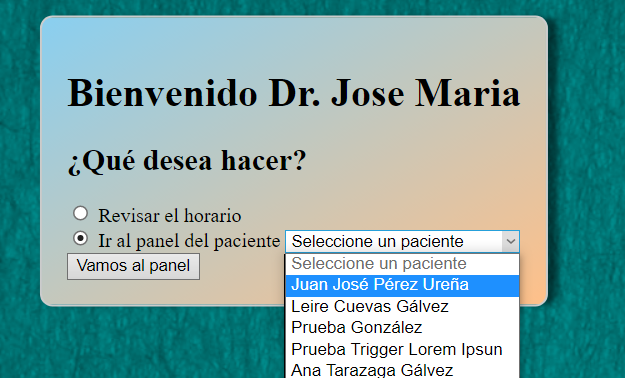
*Imagen de la selección del horario.*

En esta imagen vemos que el horario es desde las 8 de la mañana a las 14 de la tarde de lunes a viernes.



*Imagen del horario.*

Con la siguiente captura observaremos que tenemos un desplegable, el cual muestra a los pacientes registrados en la tabla pacientes, esto lo hace mediante un “SELECT dni, nombre, apellidos FROM pacientes” el cual es introducido en el selector del formulario, al seleccionar el usuario, guarda el dni del paciente en una variable llamada dni\_paciente para usarla en la siguiente página.



*Imagen de la selección del paciente*

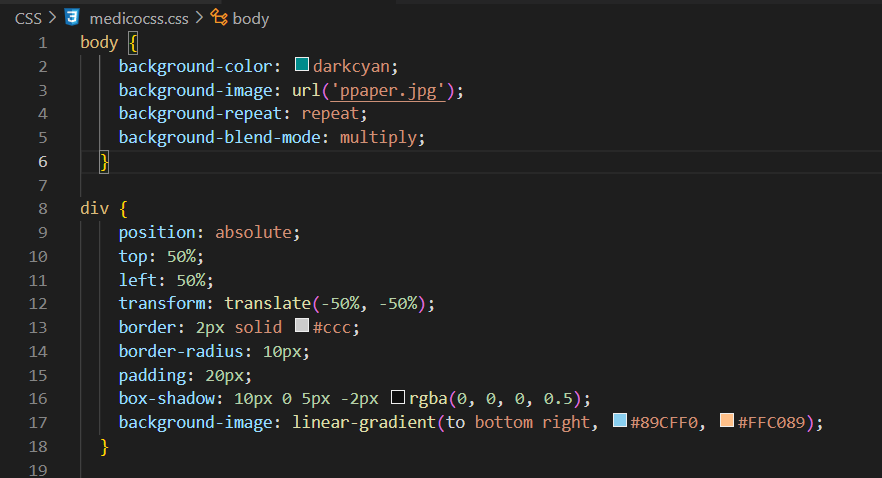
En esta otra captura vemos que tenemos 3 divs, en los cuales tenemos la información del paciente, otro en el que se mostrará el motivo de la cita y otro en el que se escribirá la receta que se almacenará en su tabla correspondiente.



*Imagen del panel del paciente*

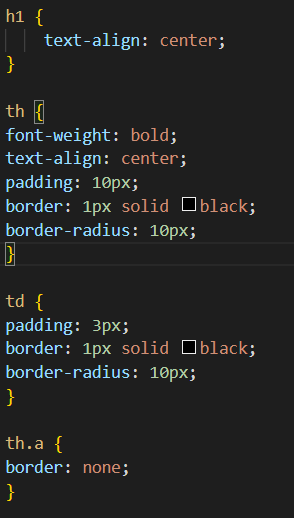
Por otra parte, tenemos el css el cual ha causado algunos problemas, ya que hay uno general para todas las páginas, pero luego algunos ajustes se han hecho en el propio fichero php.

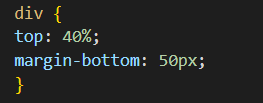
En el primer caso es un css sencillo con el color del body y el formato del div.



*Imagen del css general*

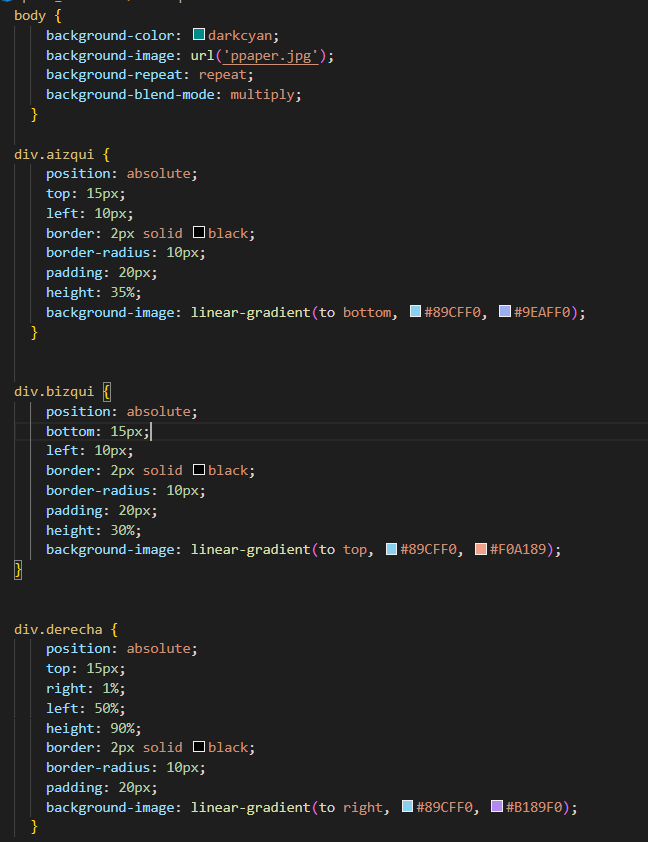
En este en cambio tenemos el formato para el titulo y para los bordes de la tabla, también vemos el del div corregido para esta página y el del botón para volver.



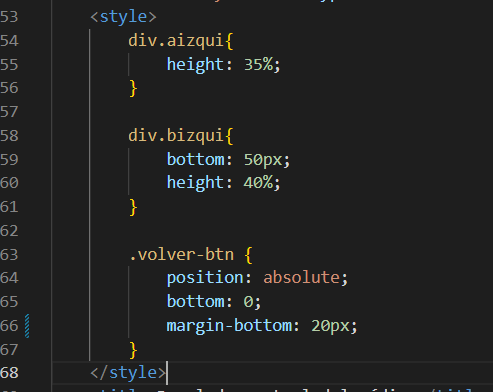


*Imagen del css de medico\_horario.php*

En este último css vemos que los ajustes son respecto al tamaño y la posición de los divs con respecto al css general y en este caso también al de panel\_css.css.



*Imagen del css general de medico\_panel.php*

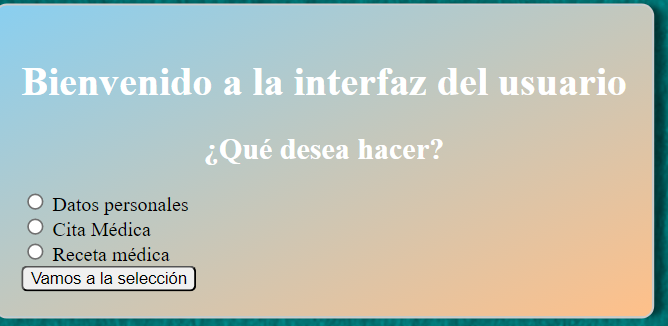


*Imagen del css de medico\_panel.php*

## **4.2º. Desarrollo del apartado del paciente.**

En este apartado explicare el proceso de desarrollo de los paneles/páginas que conforman las opciones que puede realizar el paciente.

Para comenzar, tenemos el panel con las 3 opciones que puede realizar el paciente.



*Imagen del panel del paciente.*

Como podemos ver, tenemos la opción de visualizar los datos personales, la selección de cita médica y la de ver las recetas médicas que tenga el paciente, a modo de histórico.

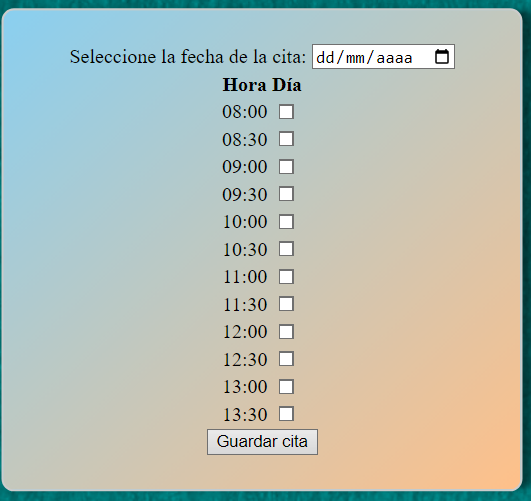
Para comenzar hablaremos de la opción de ver los datos personales, la cual únicamente tiene los datos del paciente y en el futuro una opción para que, si hay algún apartado erróneo, se le pueda enviar un ticket al administrador para que cambie ese dato y asi restringir también para que el paciente no pueda cambiar asi como así sus datos.



*Imagen del panel con la información del paciente.*

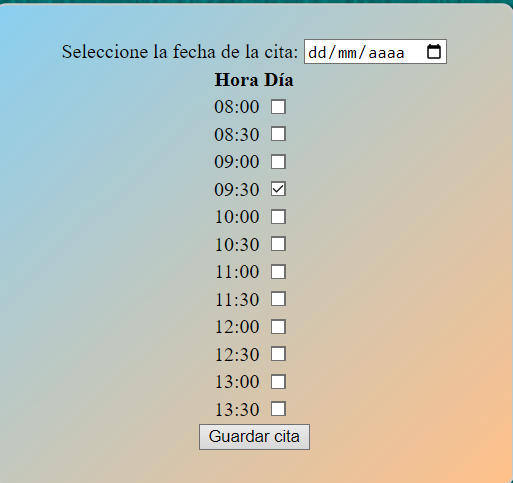
En este caso, podemos ver que este contiene 3 elementos, el elemento date, para seleccionar la fecha, que restringe la selección de fines de semana, la tabla con las horas y el botón de guardar la cita.

A continuación, pasaremos a hablar de la selección de la cita.



*Imagen del panel del horario 1.*

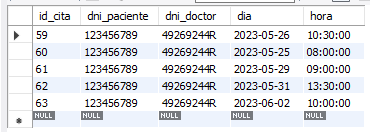
En el caso del panel de la selección de la hora de la cita, este tiene un poco más de miga, pues para que solo se pueda seleccionar un checkbox, he usado JavaScript para que detecte cuando se pulsa una caja y desactive esa columna.



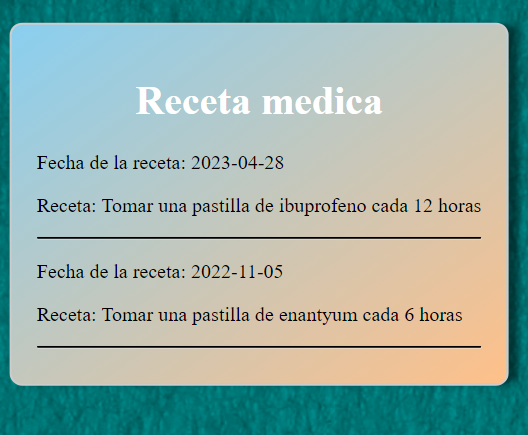
*Imagen del panel del horario 2.*

En este caso, dentro del div que agrupa esta tabla, tenemos el botón de guardar cita, mientras que fuera, tenemos el de volver a la selección, este es, ya que, si el paciente se ha equivocado al seleccionar la opción, que no tenga que cerrar la página y volverla a abrir para seleccionar la otra opción.

Al guardar la cita, la hora y el día se guardarán en la tabla cita de la base de datos usada durante todo el proyecto.



Por último, explicare cómo funciona el histórico de recetas.

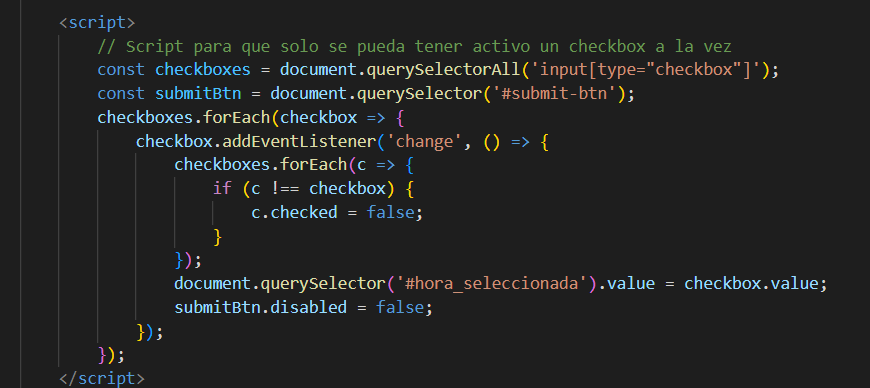


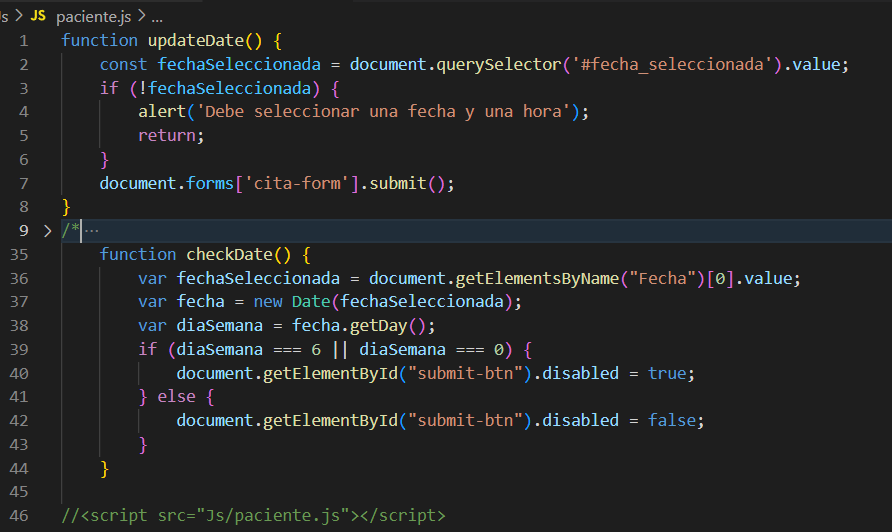
*Imagen del panel del histórico de recetas.*

Como podemos ver en la captura anterior, este paciente consta ahora mismo con 2 recetas, una de 2023 y otra de 2022, como vemos están separadas entre si mediante la etiqueta de HTML llamada “hr”, la intención ya que este es una mera prueba, es que se impriman solo las 3 ultimas recetas, para así no ocupar toda la pantalla con todas las recetas del paciente.

Para el css en este caso es muy similar, por no decir el mismo, que se usa en el resto de entornos.

A continuación, mostrare el código más interesante que tiene esta parte que es para mí el JavaScript, ya que, es la parte que más trabajo me ha costado.



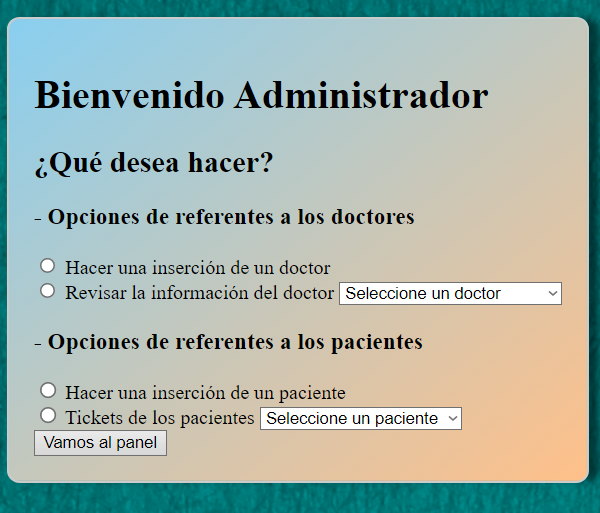


*Imagen del panel del código JavaScript.*

## **4.3º. Desarrollo del apartado del administrador.**

Dentro de este apartado explicare el desarrollo de los paneles del administrador, los cuales son conformados por la inserción de un doctor, la revisión de la información del doctor, la inserción del paciente y la actualización y visión de tickets de un paciente.

Para empezar, tenemos el panel del administrador el cual tiene varias opciones que al ser seleccionadas nos redirigen al panel deseado.



*Imagen del panel del administrador.*

Como podemos observar, el panel está dividido en 2 partes, en las opciones para los doctores y otras para las opciones de los pacientes.

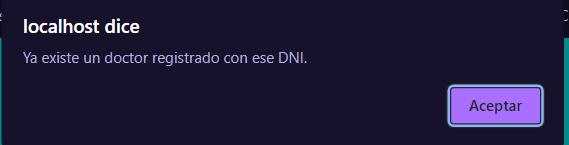
Si nos vamos a la primera opción veremos que tenemos un formulario, el cual al ser rellenado nos insertará en la base de datos un usuario doctor.



*Imagen del panel de inserción de un doctor.*

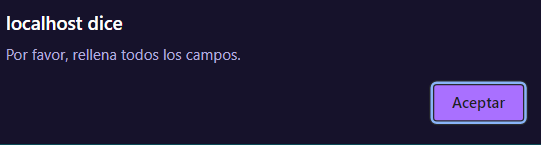
Este formulario al ser rellenado nos devuelve de nuevo al panel principal del administrador.

También si intentamos insertar un usuario cuyo dni ya existe, este dará error y nos pedirá que rellenemos todos los campos y que no repitamos el dni.



*Imagen de error 1.*

Cuando intentemos enviar algún campo vacío nos aparecerá este mensaje de error.

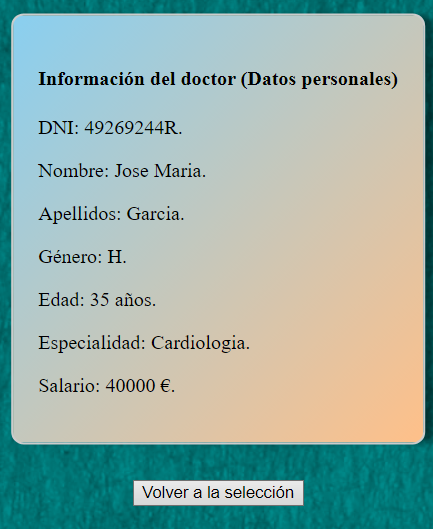


*Imagen de error 2.*

Con la siguiente opción podremos verificar la información de los doctores.



*Imagen de la selección del doctor.*



*Imagen de la información del doctor.*

Pasando ya a la siguiente opción, es lo mismo que la primera solo que para los pacientes.



*Imagen de la inserción de un paciente.*

Cuando los datos son rellenados correctamente son insertados en la base de datos, mientras que si se deja un campo vacío o ya existe un paciente con el mismo dni saldrán errores parecidos a los que salen en el panel de insertar un doctor.

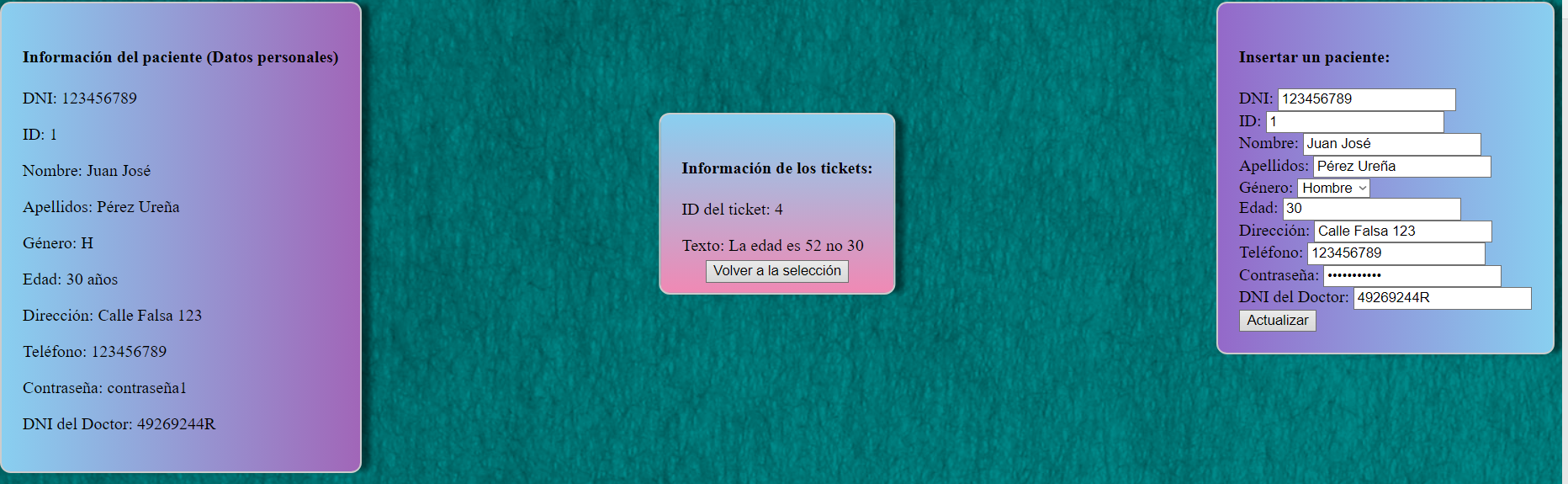
En el siguiente panel tenemos el visor de tickets, por si algún dato de un paciente está mal, que este pueda enviar un aviso al administrador y que este lo cambie.



*Imagen de la selección del ticket del paciente.*

A continuación, en el panel de los tickets tenemos la información actual del paciente, el mensaje donde pide la corrección y el formulario que pide los datos para la actualización.

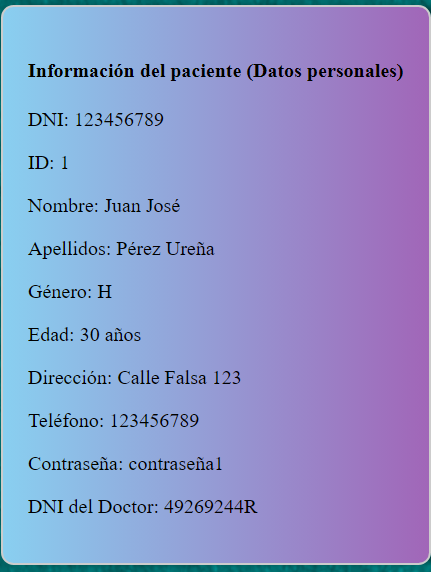
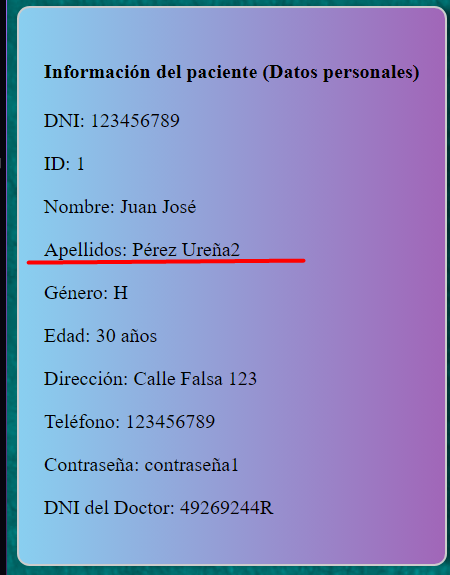
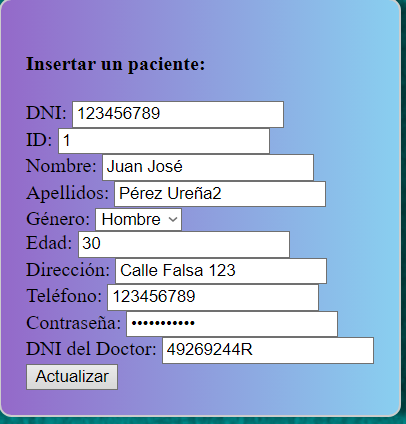
Como podemos ver en la imagen, en el div que actualiza los datos del paciente, mete automáticamente los datos tal y como están en la tabla, para que no quede ningún campo vacio.



*Imagen del panel de los tickets.*

Al hacer alguna actualización, los datos se actualizan tanto en la tabla de pacientes como en la tabla de usuarios.

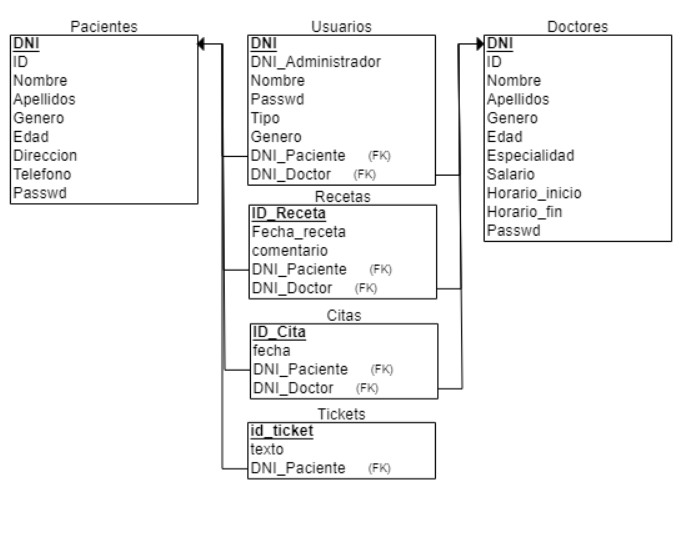
A continuación, mostrare como se actualizan los datos de los pacientes.

*Conjunto de imágenes de como se actualizan los datos del paciente.*

## **4.4º. Diseño de la base de datos.**

Este proyecto usa una base de datos en MYSQL, esta contiene las tablas de paciente, doctores, usuarios, recetas, horarios, citas y tickets.



En esta imagen podemos ver el modelo relacional de la base de datos, viendo cada conexión de manera clara y concisa.

Para la funcionalidad del login, usamos la tabla usuarios, la cual reúne a los 3 tipos de usuarios para poder verificar el tipo de usuario que intenta entrar.

Para la tabla recetas, almacena el id de la receta, que es auto incremental, tenemos la fecha de la receta, para poder ordenarla en el panel del paciente, también almacena la receta dentro del campo comentario y los dni del paciente y del doctor.

La tabla citas es usada para la asignación de una cita por parte del paciente y la revisión de esta por parte del doctor.

Por último, la tabla tickets almacena la incidencia que tiene el paciente con sus datos, para que más adelante el administrador pueda cambiar los datos erróneos del paciente.

# **Problemas encontrados durante el desarrollo:**

1º. A la hora de imprimir por pantalla los datos del paciente en la página medico\_panel.php, no conseguía sacar los datos del paciente y esto era debido a que no se estaba guardando el dni del doctor desde medico\_php y por lo tanto al pasarlo no se detectaba, esto fue corregido creando la variable $dni\_paciente e indicando que su valor será el dni seleccionado en el selector del formulario.

2º. A la hora de implementar la tabla con el horario para que pidan cita los pacientes, no se insertaban los datos, el problema era que el formulario estaba fuera de la tabla, por lo que no se enviaban los horarios seleccionados, concretamente la hora seleccionada mediante el checkbox, para ello moví todo el formulario dentro del div y le tuve que agregar el atributo name a los checkboxes para que se pudieran enviar correctamente y con todo esto ya funcionó.

3º. A la hora de implementar la función de insertar la receta en la tabla he estado teniendo errores, puesto que no hacia correctamente la inserción.

El problema era la sentencia del statement que he estado usando, ya que estaba definiendo el campo fecha como int en vez de como string, también había otro error que no conseguí solucionar que al recargar la página los datos del paciente desaparecían, al final también tuve algunos errores que no supe identificar y la solución fue crear desde cero el panel, el cual ha quedado mejor optimizado y con una funcionalidad completa.

# **¿Cómo expandiría esta idea?**

Para expandir este proyecto o como lo haría de otra manera, para empezar, seguramente montaría todo usando Python y Django, los cuales son un lenguaje de programación de alto nivel, y Django que es un framework de desarrollo web de alto nivel escrito en Python.

Esto ayudaría y optimizaría la velocidad de las consultas a la base de datos y una mejora del login bastante contundente.

# **Conclusión:**

# **Bibliografía**:

-Enlaces seguidos para la redirección a las diferentes páginas:

<https://code.tutsplus.com/es/tutorials/how-to-redirect-with-php--cms-34680>

<https://help.dreamhost.com/hc/es/articles/214395288-Crear-un-redireccionamiento-de-PHP>

<https://www.factoriacreativabarcelona.es/blog/redireccion-php/>

-Enlaces seguidos para instalar Docker y Kubernetes:  
 -Docker:

<https://sergioss.net/instalar-docker-ce-en-ubuntu-22-04-lts-jammy/>

-Kubernetes:  
 <http://localhost:51235/unidad2_2> (Manuales dados por el profesor

Nicolás José del Pozo Madroñal durante el segundo trimestre)

-Enlaces seguidos para el balanceo de carga con Kubernetes:

<https://www.youtube.com/watch?v=0iMEcrcfG5A&ab_channel=PeladoNerd>

<https://www.youtube.com/watch?v=2i5t6bfgWvE&ab_channel=PeladoNerd>

<https://kubernetes.io/es/docs/concepts/services-networking/service/>

<https://www.ovhcloud.com/es-es/public-cloud/kubernetes/kubernetes-load-balancer/>