

## T01.06 – Realizar la validación en servidor y guardado de datos de alta de médicos en la BBDD

<b>Historia de usuario</b>	Como desarrollador, necesito, poder conectar el formulario de alta de personal médico con la base de datos, de esta manera almacenar todos los campos <u>necesarios</u> para obtener la información del personal médico, con la finalidad de tener un control de que usuarios quedan registrados.
<b>Descripción / Aclaraciones</b>	Realizar el PHP nuevo_personal, en el que conectar la base de datos mysql, y realizar la consulta e inserción de los datos introducidos en el formulario HTML a través de objetos JSON.
<b>Persona encargada</b>	ALEJANDRO VENTURA GOMEZ
<b>Tiempo estimado</b>	8
<b>Tiempo empleado</b>	10
<b>Comentarios</b>	Diseño: Diseño de la conexión a la base de datos así como de las consultas para realizar la inserción de los nuevos médicos. Código implementado: insertarPersonal.php Dificultades/Aclaraciones : Dificultades al tratar los objetos JSON, para relacionar el formulario HTML con el PHP insertarPersonal.

#### T01.04 – Diseño de la Base de Datos

<b>Historia de usuario</b>	Como desarrolladora, necesito representar el diseño de la base de datos que utilizará la aplicación, con la finalidad de poder administrar correctamente la clínica.
<b>Descripción / Aclaraciones</b>	Para el diseño de la base de datos, hago uso del modelo entidad relación (MER).
<b>Persona encargada</b>	Carolina Huesca Poveda
<b>Tiempo estimado</b>	4h
<b>Tiempo empleado</b>	2h
<b>Comentarios</b>	<p>Diseño:</p> <ul style="list-style-type: none"><li>- Se encuentra en la imagen “diseño_bddd.png”.</li></ul> <p>Dificultades:</p> <ul style="list-style-type: none"><li>- Abarcar todo tipo de cambio o mejora en la aplicación y tenerlo en cuenta para que afecte lo menos posible al diseño de la base de datos.</li></ul>

### T01.05 – Creación de la Base de Datos

<b>Historia de usuario</b>	Como desarrolladora, necesito crear la base de datos que utilizará la aplicación, con la finalidad de usarla para gestionar todos los datos que genere la clínica.
<b>Descripción / Aclaraciones</b>	Para la creación de la base de datos, me decanto por usar MySQL.
<b>Persona encargada</b>	Carolina Huesca Poveda
<b>Tiempo estimado</b>	4h
<b>Tiempo empleado</b>	2h
<b>Comentarios</b>	<p>Código implementado:</p> <ul style="list-style-type: none"><li>- Se encuentra en el fichero clínica_upm.sql.</li></ul> <p>Dificultades:</p> <ul style="list-style-type: none"><li>- Exportar las relaciones que hay entre las tablas para que las tengan mis compañeros en un .sql.</li></ul>

### T01.09 – Recibir y mostrar la respuesta de la petición AJAX

<b>Historia de usuario</b>	Como desarrolladora, necesito obtener la respuesta del servidor ante una petición AJAX realizada por el usuario, con la finalidad de notificar si la petición se ha realizado correctamente o no.
<b>Descripción / Aclaraciones</b>	Por la parte del servidor, lo único que se ha hecho es, dar formato JSON a la respuesta de si se ha podido o no dar de alta un médico o paciente. Por el lado del cliente, se muestra una ventana emergente con esa respuesta.
<b>Persona encargada</b>	Carolina Huesca Poveda
<b>Tiempo estimado</b>	4h
<b>Tiempo empleado</b>	2h
<b>Comentarios</b>	Código implementado: - insertarPaciente.php, en concreto, el array \$response. - insertarPersonal.php, en concreto, el array \$response.

#### T02.04 – Recibir y mostrar la respuesta de la petición AJAX

<b>Historia de usuario</b>	Como desarrolladora, necesito obtener la respuesta del servidor ante una petición AJAX realizada por el usuario, con la finalidad de notificar si la petición se ha realizado correctamente o no.
<b>Descripción / Aclaraciones</b>	<p>Por la parte del servidor, lo único que se ha hecho es, dar formato JSON a la respuesta de si se ese usuario está registrado o no en el sistema.</p> <p>Por el lado del cliente, se muestra una ventana emergente con esa respuesta.</p>
<b>Persona encargada</b>	Carolina Huesca Poveda
<b>Tiempo estimado</b>	4h
<b>Tiempo empleado</b>	2h
<b>Comentarios</b>	<p>Código implementado:</p> <ul style="list-style-type: none"><li>- login.php, en concreto, el array \$response.</li></ul>

### T01.01 – Diseñar formulario de alta de médicos

<b>Historia de usuario</b>	Como desarrollador necesito crear un formulario, el cual recoja los datos necesarios del formulario para dar de alta a un médico y sea intuitivo para el cliente.
<b>Descripción / Aclaraciones</b>	Para llevarlo a cabo realizo un documento html mediante una plantilla bootstrap.
<b>Persona encargada</b>	Héctor Blanco González
<b>Tiempo estimado</b>	4h
<b>Tiempo empleado</b>	2h
<b>Comentarios</b>	<p>Diseño: Adaptado a los requisitos necesarios para dar de alta a un médico.</p> <p>Código implementado: Desarrollado mediante htm5 y una plantilla bootstrap. Fichero nuevoPersonal.html</p> <p>Prueba asociada: Rellenar los campos y comprobar que al dar de alta se produce un envío con los datos del formulario.</p> <p>Dificultades/Aclaraciones : Necesidad de crear una nueva tarea para añadir funcionalidad al botón añadir especialidad.</p>

## T01.02 – Diseñar formulario de alta de clientes

<b>Historia de usuario</b>	Como desarrollador necesito crear un formulario, el cual recoja los datos necesarios del formulario para dar de alta a un médico y sea intuitivo para el cliente.
<b>Descripción / Aclaraciones</b>	Para llevarlo a cabo realizo un documento html mediante una plantilla bootstrap.
<b>Persona encargada</b>	Héctor Blanco González
<b>Tiempo estimado</b>	4h
<b>Tiempo empleado</b>	2h
<b>Comentarios</b>	<p>Diseño: Adaptado a los requisitos para dar de alta un paciente.</p> <p>Código implementado: Desarrollado mediante htm5 y una plantilla bootstrap. Fichero nuevoPaciente.html</p> <p>Prueba asociada: Rellenar los campos y comprobar que al dar de alta se produce un envío con los datos del formulario.</p> <p>Dificultades/Aclaraciones : Se reutilizó código del formulario alta de médicos (T01.01).</p>

### T01.07 – Realizar la validación en servidor y guardado de datos de alta de cliente en la BBDD

<b>Historia de usuario</b>	Como desarrollador necesito crear un formulario, el cual recoja los datos necesarios del formulario para dar de alta a un médico y sea intuitivo para el cliente.
<b>Descripción / Aclaraciones</b>	Para llevarlo a cabo realizo un fichero php relacionado con el html del formulario correspondiente.
<b>Persona encargada</b>	Héctor Blanco González
<b>Tiempo estimado</b>	8h
<b>Tiempo empleado</b>	9h
<b>Comentarios</b>	<p>Código implementado: Fichero insertarPaciente.php</p> <p>Prueba asociada: Rellenar el formulario asociado, comprobar que los datos son correctos y verificar que han sido insertados correctamente en la BBDD.</p> <p>Dificultades/Aclaraciones : Dificultad al tratar los datos del objeto <u>json</u> recibido para trabajar sobre php.</p>



## T02.01 – Página de login(Cliente)

<b>Historia de usuario</b>	Equipo de desarrollo. Pantalla de login para la autorización de los usuarios. Gestión del acceso a la aplicación.
<b>Descripción / Aclaraciones</b>	Una página web que envía un formulario con usuario (en este caso DNI) y con su contraseña correspondiente.
<b>Persona encargada</b>	Miguel Arenal
<b>Tiempo estimado</b>	2 horas
<b>Tiempo empleado</b>	3 horas
<b>Comentarios</b>	<b>Diseño (si procede):</b> Se ha integrado con el diseño de la aplicación, el diseño de la parte de formulario ha sido diseñada por separado. <b>Código implementado (si procede):</b> Código html y css para el diseño del formulario. <b>Prueba asociada (si procede):</b> Confirmar que manda un formulario correctamente. <b>Dificultades/Aclaraciones (si procede):</b> Primera vez usando css por lo que llevo más tiempo de lo esperado modelar la página. También se ha llevado acabo algunas modificaciones posteriores para ajustar el formulario para que funcionase bien con JavaScript

T02.02 – PHP Login	
<b>Historia de usuario</b>	Equipo de desarrollo Comprobación de las credenciales Autenticación del usuario
<b>Descripción / Aclaraciones</b>	Código que busca y compara en la base de datos las credenciales introducidas en la pantalla de login para su autenticación e identificación del tipo de usuario(médico o gestor)
<b>Persona encargada</b>	Miguel Arenal
<b>Tiempo estimado</b>	8 horas
<b>Tiempo empleado</b>	6 horas
<b>Comentarios</b>	<p>Código implementado (si procede): PHP</p> <p>Prueba asociada (si procede): Pruebas de funcionamiento con la base de datos:</p> <ul style="list-style-type: none"> <li>• Medico: DNI y contraseña verificados</li> <li>• Gestor: DNI y contraseña verificados</li> <li>• DNI y contraseña incorrectos</li> </ul> <p>Dificultades/Aclaraciones (si procede): Algunos problemas de conexion</p>

### T01.03 – Realizar el código encargado de la conexión de los formularios con el servidor

<b>Historia de usuario</b>	Como desarrollador, necesito enviar los datos de los formularios html de alta de paciente y de personal al servidor, con la finalidad de ser procesados por el mismo.
<b>Descripción / Aclaraciones</b>	Obtener mediante JQuery los datos de los formularios nuevo_paciente y nuevo_personal y enviarlos mediante un objeto JSON al servidor usando la tecnología AJAX.
<b>Persona encargada</b>	David Martín Díez de Oñate.
<b>Tiempo estimado</b>	16h
<b>Tiempo empleado</b>	8h
<b>Comentarios</b>	<p>Diseño (si procede): El diseño de los objetos JSON se detalla en el fichero doc/JSON_Objets.md Y se ha creado una función genérica para mandar datos al servidor (enviarJSON/phpUrl, jsonObjet)).</p> <p>Código implementado: Funciones nuevoPaciente(), nuevoPersonal() y enviarJSON/phpUrl, jsonObjet) del fichero assets/js/main.js</p> <p>Prueba asociada: Recibir una petición POST en servidor y devolver el mismo objeto, en cliente</p>

imprimir el objeto devuelto,

Dificultades/Aclaraciones :

Cuando se termine la implementación del procesamiento de los datos en servidor, se ha de pensar en las posibles respuestas en base a errores o si todo sale bien.

## T01.08 – Realizar el código encargado de añadir más de una especialidad en el formulario de alta de personal

<b>Historia de usuario</b>	Como desarrollador, necesito crear una función javascript para añadir otro campo a especialidad, con la finalidad de poder añadir mas de una especialidad a la hora de crear un nuevo Personal.
<b>Descripción / Aclaraciones</b>	<p>Mediante JQuery se añadirá un pedazo de html en el contenedor de las especialidades cada vez que se pulse un botón de añadir.</p> <p>Debido a que el formato del objeto JSON debe admitir un array en el apartado de especialidad, se ha de actualizar la función nuevoPersonal() para que obtenga correctamente los datos y los añada a un array dentro del objeto JSON.</p>
<b>Persona encargada</b>	David Martin Diez de Oñate.
<b>Tiempo estimado</b>	4
<b>Tiempo empleado</b>	2
<b>Comentarios</b>	<p>Código implementado (si procede): Función anyadirEspecialidad() en el fichero assets/js/main.js</p> <p>Prueba asociada: Se ha comprobado que la recepción por parte del servidor era correcta y si el JSON enviado era correcto.</p>

Dificultades/Aclaraciones:

Se ha Actualizado el objeto JSON del form  
html nuevo\_personal y su correspondiente  
función de obtención de datos  
(nuevoPersonal() en assets/js/main.js)  
para que admita un array de  
especialidades.

### T02.03 – Realizar el código encargado de la conexión del formulario de login con el servidor

<b>Historia de usuario</b>	Como desarrollador, obtener los datos del formulario del login y enviarlos al servidor , con la finalidad de realizar el login
----------------------------	--

<b>Descripción / Aclaraciones</b>	Obtener mediante JQuery los datos del formulario login y enviarlo mediante un objeto JSON al servidor usando la tecnología AJAX.
---------------------------------------	--

<b>Persona encargada</b>
--------------------------

David Martin Diez de Oñate
----------------------------

<b>Tiempo estimado</b>	1
------------------------	---

<b>Tiempo empleado</b>	1
------------------------	---

<b>Comentarios</b>	Diseño: El diseño de los objetos JSON se detalla en el fichero doc/JSON_Objets.md
--------------------	--

Código implementado : función peticiónLogging() en assets/js/main.js
--

Prueba asociada: Recibir una petición POST en servidor y devolver el mismo objeto, en cliente imprimir el objeto devuelto,
---