|  |  |
| --- | --- |
| **T01.03 – Realizar el código encargado de la conexión de los formularios con el servidor** | |
| **Historia de usuario** | Como desarrollador,  necesito enviar los datos datos de los formularios html de alta de paciente y de personal al servidor,  con la finalidad de ser procesados por el mismo. |
| **Descripción / Aclaraciones** | Obtener mediante JQuerry los datos de los formularios nuevo\_paciente y nuevo\_personal y enviarlos mediante un objeto JSON al servidor usando la tecnología AJAX. |
| **Persona encargada** | David Martín Diez de Oñate. |
| **Tiempo estimado** | 16h |
| **Tiempo empleado** | 8h |
| **Comentarios** | 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)).  Código implementado:  Funciones nuevoPaciente(), nuevoPersonal() y enviarJSON(phpUrl, jsonObjet)  del fichero assets/js/main.js  Prueba asociada:  Recibir una petición POST en servidor y devolver el mismo objeto, en cliente imprimir el objeto devuelto,  Dificultades/Aclaraciones :  Cuando se termine la implementación del procesado 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** | |
| **Historia de usuario** | 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. |
| **Descripción / Aclaraciones** | Mediante JQuerry se añadirá un pedazo de html en el contenedor de las especialidades cada vez que se pulse un botón de añadir.  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. |
| **Persona encargada** | David Martin Diez de Oñate. |
| **Tiempo estimado** | 4 |
| **Tiempo empleado** | 2 |
| **Comentarios** | Código implementado (si procede): Función anyadirEspecialidad() en el fichero assets/js/main.js  Prueba asociada:  Se ha comprobado que la recepción por parte del servidor era correcta y si el JSON enviado era correcto.  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** | |
| **Historia de usuario** | Como desarrollador,  obtener los datos del formulario del login y enviarlos al servidor ,  con la finalidad de realizar el login |
| **Descripción / Aclaraciones** | Obtener mediante JQuerry los datos del formulario login y enviarlo mediante un objeto JSON al servidor usando la tecnología AJAX. |
| **Persona encargada** | David Martin Diez de Oñate |
| **Tiempo estimado** | 1 |
| **Tiempo empleado** | 1 |
| **Comentarios** | Diseño:  El diseño de los objetos JSON se detalla en el fichero doc/JSON\_Objets.md  Código implementado : función peticionLogging() en assets/js/main.js  Prueba asociada:  Recibir una petición POST en servidor y devolver el mismo objeto, en cliente imprimir el objeto devuelto, |