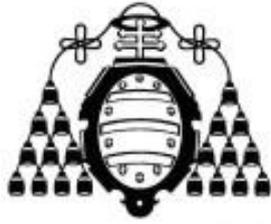


UNIVERSIDAD DE OVIEDO



ESCUELA DE INGENIERÍA INFORMÁTICA

TRABAJO FIN DE MÁSTER

“DISPENSADOR INTELIGENTE DE MEDICAMENTOS
CONECTADO Y SISTEMA DE GESTIÓN”

Vº Bº del Director del
Proyecto

DIRECTOR: Jordán Pascual Espada

CODIRECTOR: Martín González Rodríguez

AUTOR: Jose Antonio Cabañeros Blanco

Agradecimientos

El primer lugar agradecer a mis padres la oportunidad que me han concedido de continuar mi formación y animarme incondicionalmente desde el minuto uno de esta nueva etapa. Estas líneas nunca hubieran sido escritas sin su transmisión del valor del esfuerzo que me han inculcado desde niño.

En segundo lugar, a mi hermano David por haberme ayudado en todo aquello que he necesitado y por haber sido un gran compañero de este viaje. Gracias por ser el primero en aparecer en el momento que me surge un contratiempo y recuerda, algún día llegaremos más alto que los hermanos Chopin ;)

A mis directores del proyecto y en especial a Jordán por sus consejos, conocimientos y entusiasmo que han servido para evolucionar el proyecto hasta lo que es hoy.

A mi cuñada Melodie por aguantarme todo este tiempo y echarme una mano en todo lo que he necesitado. A partir de ahora los lunes serán menos lunes y por supuesto, los viernes empezaré a hacer yo la cena.

A mis abuelos por animarme constantemente a continuar con mi formación y animarme en el día a día. En especial quiero agradecer a mi abuelo Nano la preocupación transmitida por la evolución del proyecto y del máster en general.

A todos mis amigos de Laguna y del máster quienes de uno u otro modo me han ayudado a seguir evolucionando como persona.

También agradecer al resto de familiares y amigos que no he nombrado y que han aportado su granito de arena.

Muchísimas gracias a todos.

Always looking for creative ways to answer questions before they're asked

Oviedo, junio de 2018

Resumen

El proyecto tiene como objetivo crear un sistema conectado desde un enfoque innovador que facilite en la toma de medicación diaria a los pacientes que requieren de tratamientos rutinarios, permitiendo delegar la responsabilidad del control y gestión de estos tratamientos al sistema.

Para ello, se trata de buscar una alternativa más completa y precisa a los pastilleros convencionales usados en la toma de medicamentos. A través del sistema se podrán programar horarios de toma de medicamentos, que serán notificados al paciente para su toma por medio de un dispensador creado específicamente para el sistema y controlar remotamente las tomas.

Uno de los componentes fundamentales del sistema será el *dispensador inteligente de medicamentos conectado*. Este será el elemento hardware del sistema encargado de dispensar los medicamentos y de controlar su toma. Dispondrá de dos contenedores independientes que almacenarán las pastillas de dos tipos diferentes de medicamentos. También servirá como centro de monitorización del entorno y de control de las acciones del usuario para la toma de la medicación.

Por la parte software, la aplicación web de gestión y control será el centro de monitorización del dispensador. A través de esta aplicación se podrán programar horarios de toma de medicación y controlar el estado del dispensador. También permitirá controlar de forma remota la toma de la medicación, el estado de la misma y el entorno del dispensador, mostrando datos ambientales en tiempo real. Además se podrán configurar diversos parámetros, tales como el número de medicamentos del pastillero, como se avisa al paciente o la forma en la que se debe confirmar la toma de la medicación en el dispensador.

La aplicación también dispondrá de un apartado de administración para la gestión de los identificadores de los dispositivos, así como de los usuarios registrados en el sistema.

El sistema habilitará un completo centro de notificaciones que permitirá mantener informados a los usuarios de posibles anomalías en el dispensador, desconexiones y monitorizar las tomas de las medicaciones. Estas notificaciones serán realizadas a través de los correos electrónicos especificados para el dispensador y por medio de mensajes a través de la aplicación Telegram publicados en un canal creado expresamente para el sistema.

Por último, el sistema está orientado en primera instancia a pacientes de avanzada edad que usarían el dispensador y familiares o cuidadores que gestionarían y monitorizarían el dispensador a través de la aplicación. Por ello, el sistema permite no solo programar horarios de medicación y monitorizar su toma, si no que también será capaz de monitorizar el entorno donde vive el paciente, manteniendo informados a los usuarios de anomalías y situaciones de emergencia que el paciente comunique a través del dispensador.

Palabras Clave

IoT, Sistema Conectado, Dispensador de Medicamentos, Programación de Horarios, Gestión, Monitorización Remota, Aplicación Web, Microcontrolador, Notificaciones de Alarmas, Control, Telegram

Índice General

CAPÍTULO 1. MEMORIA DEL PROYECTO	15
1.1 RESUMEN DE LA MOTIVACIÓN, OBJETIVOS Y ALCANCE DEL PROYECTO	16
1.2 RESUMEN DE TODOS LOS ASPECTOS	18
CAPÍTULO 2. INTRODUCCIÓN.....	19
2.1 JUSTIFICACIÓN DEL PROYECTO.....	19
2.2 OBJETIVOS DEL PROYECTO	20
2.3 ESTUDIO DE LA SITUACIÓN ACTUAL.....	21
2.3.1 <i>Sistemas de programación de horarios y alarmas.....</i>	<i>21</i>
2.3.2 <i>Aplicaciones móviles de programación de horarios</i>	<i>24</i>
2.3.3 <i>Conclusiones.....</i>	<i>28</i>
CAPÍTULO 3. ANÁLISIS.....	30
3.1 DEFINICIÓN DEL SISTEMA.....	30
3.1.1 <i>Determinación del Alcance del Sistema</i>	<i>30</i>
3.1.2 <i>Especificación textual de los requisitos del sistema.....</i>	<i>30</i>
3.1.3 <i>Requisitos del Sistema.....</i>	<i>32</i>
3.1.4 <i>Identificación de Actores del Sistema</i>	<i>37</i>
3.1.5 <i>Especificación de Casos de Uso.....</i>	<i>37</i>
3.2 IDENTIFICACIÓN DE LOS SUBSISTEMAS EN LA FASE DE ANÁLISIS	48
3.2.1 <i>Descripción de los Subsistemas.....</i>	<i>48</i>
3.2.2 <i>Descripción de los Interfaces entre Subsistemas</i>	<i>49</i>
3.3 ANÁLISIS DEL DISPOSITIVO DISPENSADOR DE MEDICAMENTOS.....	51
3.4 ANÁLISIS Y ESTUDIO DE ALTERNATIVAS	53
3.4.1 <i>Sistema Distribuido y Base de Datos - Backend.....</i>	<i>53</i>
3.4.2 <i>Dispensador, Microcontrolador y Componentes Electrónicos.....</i>	<i>56</i>
3.4.3 <i>Aplicación Web - Frontend.....</i>	<i>65</i>
3.5 DIAGRAMA DE CLASES PRELIMINAR DEL SISTEMA DISTRIBUIDO	71
3.6 DIAGRAMA PRELIMINAR DE LAS FUNCIONES DEL MICROCONTROLADOR.....	72
3.7 ANÁLISIS DE INTERFACES DE USUARIO DE LA APLICACIÓN WEB	73
3.7.1 <i>Descripción de la Interfaz de la Aplicación web.....</i>	<i>73</i>
3.7.2 <i>Descripción del Comportamiento de la Interfaz de la aplicación</i>	<i>78</i>
3.7.3 <i>Diagrama de Navegabilidad</i>	<i>79</i>
3.8 ANÁLISIS DEL INTERFAZ FÍSICO DEL DISPENSADOR DE PASTILLAS CONECTADO	81
3.9 HIPÓTESIS Y RESTRICCIONES	83
3.10 PRESUPUESTO PARA EL CLIENTE	85
3.11 ESPECIFICACIÓN DEL PLAN DE PRUEBAS	86
CAPÍTULO 4. DISEÑO DEL SISTEMA	87
4.1 ARQUITECTURA DEL SISTEMA.....	87
4.2 DISEÑO DE CASOS DE USO	90
4.2.1 <i>Diagramas de secuencia de los casos de uso del actor Usuario Anónimo.....</i>	<i>91</i>
4.2.2 <i>Diagramas de secuencia de los casos de uso del actor Usuario Cliente</i>	<i>93</i>
4.2.3 <i>Diagramas de secuencia de los casos de uso del actor Usuario Administrador.....</i>	<i>101</i>
4.2.4 <i>Diagramas de secuencia de los casos de uso del actor Usuario del Dispensador</i>	<i>103</i>

4.2.5	<i>Diagramas de secuencia de casos de uso del actor Dispensador</i>	104
4.2.6	<i>Diagramas de secuencia de casos de uso de los actores Servidor de correo y Telegram</i>	106
4.3	ELECCIÓN DE ALTERNATIVAS	107
4.3.1	<i>Sistema Distribuido y Base de Datos - Backend</i>	107
4.3.2	<i>Dispensador, Microcontrolador y Componentes Electrónicos</i>	108
4.3.3	<i>Aplicación Web – Frontend</i>	111
4.3.4	<i>Conclusiones</i>	112
4.4	DISEÑO DE CLASES DEL SISTEMA DISTRIBUIDO	114
4.5	DISEÑO DE LAS FUNCIONES DEL MICROCONTROLADOR	115
4.6	DISEÑO DEL SISTEMA DISTRIBUIDO – BACKEND	117
4.6.1	<i>Aplicación web en el servidor</i>	118
4.6.2	<i>Microcontrolador del dispensador en el servidor</i>	118
4.6.3	<i>Proceso de comprobación de conexión del dispensador</i>	119
4.6.4	<i>Sistema de Alertas y Notificaciones</i>	120
4.7	DISEÑO DE LA BASE DE DATOS – BACKEND	122
4.8	DISEÑO DEL DISPENSADOR DE MEDICAMENTOS CONECTADO – HARDWARE	127
4.8.1	<i>Sistema de control del Dispensador</i>	129
4.9	DISEÑO DE LA INTERFAZ DE LA APLICACIÓN WEB – FRONTEND	131
4.9.1	<i>Diseño y componentes de la interfaz de usuario</i>	131
4.9.2	<i>Diseño responsive</i>	140
4.10	PROTOCOLO DE COMUNICACIÓN ENTRE EL DISPENSADOR Y EL SERVIDOR – API	142
4.10.1	<i>Peticiones GET – Solicitud de horarios</i>	142
4.10.2	<i>Peticiones POST – Evento en el dispensador</i>	144
4.11	ESPECIFICACIÓN TÉCNICA DEL PLAN DE PRUEBAS	146
4.11.1	<i>Pruebas Unitarias y de Integración</i>	146
4.11.2	<i>Pruebas de Usabilidad</i>	151
CAPÍTULO 5. IMPLEMENTACIÓN DEL SISTEMA		153
5.1	ESTÁNDARES Y NORMAS SEGUIDOS	153
5.2	LENGUAJES DE PROGRAMACIÓN	154
5.3	HERRAMIENTAS Y PROGRAMAS USADOS PARA EL DESARROLLO	155
5.3.1	<i>Eclipse junto al plugin NodeEclipse</i>	155
5.3.2	<i>Arduino Software IDE</i>	155
5.3.3	<i>Control de versiones Git</i>	156
5.3.4	<i>GitHub</i>	156
5.3.5	<i>MongoDB en local</i>	156
5.3.6	<i>RoboMongo</i>	157
5.3.7	<i>Postman</i>	157
5.3.8	<i>Heroku</i>	158
5.3.9	<i>mLab</i>	158
5.3.10	<i>Tinkercad</i>	158
5.3.11	<i>Cura</i>	159
5.4	CREACIÓN DEL SISTEMA	160
5.4.1	<i>Dificultades Superadas</i>	160
5.4.2	<i>Paquetes externos utilizados en la implementación del sistema distribuido Node.js</i>	161
5.4.3	<i>Implementación del Microcontrolador</i>	162
5.4.4	<i>Detalles de construcción del dispensador de medicamentos</i>	164
CAPÍTULO 6. DESARROLLO DE LAS PRUEBAS		167
6.1	PRUEBAS UNITARIAS	167

Índice de Figuras

Figura 2.1. Pastillero con alarma	21
Figura 2.2. Pastillero inteligente Pill-Buddy	22
Figura 2.3. Pastillero inalámbrico MedMinder	23
Figura 2.4. Icono aplicación Medisafe	24
Figura 2.5. Pantalla de inicio de Medisafe	25
Figura 2.6. Icono aplicación MyTherapy	26
Figura 2.7. Pantalla de inicio de MyTherapy	26
Figura 2.8. Icono aplicación Hora de la Medicina	27
Figura 2.9. Pantalla de inicio Hora de la Medicina	27
Figura 3.1. Casos de uso - Usuario Anónimo	38
Figura 3.2. Casos de uso 1 - Usuario Cliente	39
Figura 3.3. Casos de uso 2 - Usuario Cliente	41
Figura 3.4. Casos de uso - Usuario Administrador	43
Figura 3.5. Casos de uso - Usuario del Dispensador	44
Figura 3.6. Casos de uso – Dispensador.....	45
Figura 3.7. Casos de uso - Servidor de correo y Telegram.....	47
Figura 3.8. Diagrama conceptual	48
Figura 3.9. Logo framework de desarrollo Spring	53
Figura 3.10. Logo del entorno Node.js	54
Figura 3.11. Logo base de datos relacional MySQL	55
Figura 3.12. Logo de base de datos no relacional MongoDB.....	56
Figura 3.13. Enfoque de pastillero	57
Figura 3.14. Enfoque de dispensador	58
Figura 3.15. Microcontrolador WeMos D1 R2.....	59
Figura 3.16. Microcontrolador MEGA + ESP8266	60
Figura 3.17. Servomotor de rotación continua	61
Figura 3.18. Motor paso a paso 28BYJ-48.....	62
Figura 3.19. Balanza de carga	63
Figura 3.20. Sensor infrarrojos IR.....	63
Figura 3.21. Placa de expansión PCB	64
Figura 3.22. Placa de expansión MEGA	64
Figura 3.23. Logo Angular 5	66
Figura 3.24. Logo React.....	66
Figura 3.25. Logo Vue.js	67
Figura 3.26. Logo Moment.js.....	67
Figura 3.27. Logo jQuery	68
Figura 3.28. Logo Bootstrap	69
Figura 3.29. Logo D3	69
Figura 3.30. Diagrama de clases preliminar	71
Figura 3.31. Interacción entre funciones preliminares del microcontrolador	72
Figura 3.32. Mockup interfaz de la aplicación – Login	73
Figura 3.33. Mockup interfaz de la aplicación – Registro.....	74
Figura 3.34. Mockup interfaz de la aplicación – Horarios	74
Figura 3.35. Mockup interfaz de la aplicación – Dispensador	75
Figura 3.36. Mockup interfaz de la aplicación – Perfil	75

Figura 3.37. Mockup interfaz de la aplicación - Ajustes – Tratamiento	76
Figura 3.38. Mockup interfaz de la aplicación – Seriales	76
Figura 3.39. Mockup interfaz de la aplicación – Usuarios.....	77
Figura 3.40. Mockup interfaz móvil de la aplicación – Horarios.....	77
Figura 3.41. Diagrama de navegabilidad entre interfaces de la aplicación web	79
Figura 3.42. Boceto del dispensador de pastillas conectado.....	81
Figura 3.43. Boceto de uno de los pastilleros del dispensador.....	82
Figura 4.1. Arquitectura del sistema	87
Figura 4.2. Diagrama de secuencia "Registrarse en el sistema"	91
Figura 4.3. Diagrama de secuencia "Identificarse como cliente"	92
Figura 4.4. Diagrama de secuencia "Identificarse como administrador"	92
Figura 4.5. Diagrama de secuencia "Consultar estado del dispensador"	93
Figura 4.6. Diagrama de secuencia "Programar horarios de toma de medicación"	94
Figura 4.7. Diagrama de secuencia "Consultar y eliminar horarios"	95
Figura 4.8. Diagrama de secuencia "Visualizar información del dispensador"	95
Figura 4.9. Diagrama de secuencia "Modificar datos personales"	96
Figura 4.10. Diagrama de secuencia "Especificar pastillas introducidas"	97
Figura 4.11. Diagrama de secuencia "Configurar toma de medicación"	97
Figura 4.12. Diagrama de secuencia "Configurar avisos de toma en el dispensador"	98
Figura 4.13. Diagrama de secuencia "Añadir o eliminar emails de notificación"	98
Figura 4.14. Diagrama de secuencia "Suscribirse al canal de Telegram"	99
Figura 4.15. Diagrama de secuencia "Configurar notificaciones"	100
Figura 4.16. Diagrama de secuencia "Añadir nuevo de dispensador"	101
Figura 4.17. Diagrama de secuencia "Consultar y eliminar seriales"	102
Figura 4.18. Diagrama de secuencia "Consultar y eliminar usuarios registrados"	102
Figura 4.19. Diagrama de secuencia "Recoger y tomar medicación"	103
Figura 4.20. Diagrama de secuencia "Confirmar medicación"	103
Figura 4.21. Diagrama de secuencia "Emitir notificación de emergencia"	104
Figura 4.22. Diagrama de secuencia "Rutina del dispensador"	105
Figura 4.23. Diagrama de secuencia "Emitir notificaciones de alarmas"	106
Figura 4.24. Diseño de clases del sistema distribuido	114
Figura 4.25. Interacción entre las funciones de los módulos del microcontrolador.....	116
Figura 4.26. Diseño del sistema distribuido – Backend	117
Figura 4.27. Ejemplo de alarma por correo electrónico.....	120
Figura 4.28. Ejemplo de notificaciones por el bot de Telegram	121
Figura 4.29. Árbol JSON usuario	122
Figura 4.30. Árbol JSON dispensador	124
Figura 4.31. Árbol JSON horario.....	125
Figura 4.32. Árbol JSON dato.....	126
Figura 4.33. Diseño del dispensador de medicamentos conectado.....	127
Figura 4.34. Diseño del dispensador de medicamentos – Parte trasera	128
Figura 4.35. Esquema de conexiones del dispensador	129
Figura 4.36. Pines de conexión de los componentes al microcontrolador	130
Figura 4.37. Diseño de la interfaz – Login.....	131
Figura 4.38. Diseño de la interfaz – Registro.....	132
Figura 4.39. Diseño de la interfaz – Horarios	132
Figura 4.40. Diseño de la interfaz – Dispensador	133
Figura 4.41. Diseño de la interfaz – Registro temperatura	134
Figura 4.42. Diseño de la interfaz – Perfil	135
Figura 4.43. Diseño de la interfaz – Ajustes – Tratamiento.....	135

Figura 4.44. Diseño de la interfaz – Ajustes - Toma de medicación	136
Figura 4.45. Diseño de la interfaz – Ajustes – Dispensador.....	137
Figura 4.46. Diseño de la interfaz – Ajustes – Notificaciones.....	137
Figura 4.47. Diseño de la interfaz – Ajustes – Configurador de notificaciones	138
Figura 4.48. Diseño de la interfaz – Administración – Serials.....	139
Figura 4.49. Diseño de la interfaz – Administración – Usuarios	139
Figura 4.50. Diseño de la interfaz - Horarios - Smartphone vertical	140
Figura 4.51. Diseño de la interfaz - Dispensador - Smartphone vertical	140
Figura 4.52. Diseño de la interfaz - Horarios - Smartphone horizontal	141
Figura 4.53. Diseño de la interfaz - Dispensador - Smartphone horizontal	141
Figura 4.54. Protocolo de comunicación GET - Solicitud de horarios.....	142
Figura 4.55. Protocolo de comunicación POST – Actualización y notificación.....	144
Figura 5.1. Logo Eclipse Oxygen.....	155
Figura 5.2. Logo Arduino	155
Figura 5.3. Logo Git.....	156
Figura 5.4. Logo GitHub.....	156
Figura 5.5. Logo MongoDB	157
Figura 5.6. Logo RoboMongo.....	157
Figura 5.7. Logo Postman	157
Figura 5.8. Logo Heroku	158
Figura 5.9. Logo mLab.....	158
Figura 5.10. Logo Tinkercad.....	159
Figura 5.11. Logo Cura	159
Figura 5.12. Diseño final del dispensador de medicamentos	165
Figura 5.13. Diseño final. Componentes del dispensador	166
Figura 7.1. Instalación del microcontrolador del dispensador – SoftwareSerial.....	179
Figura 7.2. Instalación del microcontrolador del dispensador – Switch del microcontrolador.....	180
Figura 7.3. Instalación del microcontrolador del dispensador – Módulo ESP8266	180
Figura 7.4. Instalación del microcontrolador del dispensador – Módulo ATmega2560	181
Figura 7.5. Instalación del sistema distribuido, aplicación web y base de datos – Iniciar MongoDB .	182
Figura 7.6. Instalación del sistema distribuido, aplicación web y base de datos – Despliegue de la aplicación	182
Figura 7.7. Ejecución del sistema – MongoDB	183
Figura 7.8. Ejecución del sistema – Node.js.....	183
Figura 7.9. Ejecución del sistema – Despliegue en local	184
Figura 7.10. Ejecución del sistema - Serial del dispensador	184
Figura 7.11. Ejecución del sistema – Desconexión del dispensador.....	184
Figura 7.12. Ejecución del sistema – Conexión del dispensador.....	185
Figura 7.13. Manual de la aplicación web – Registro	186
Figura 7.14. Manual de la aplicación web – Inicio de Sesión.....	187
Figura 7.15. Manual de la aplicación web – Gestión de horarios y dispensador	188
Figura 7.16. Manual de la aplicación web – Gestión de horarios y dispensador – Móvil	189
Figura 7.17. Manual de la aplicación web – Dispensador	189
Figura 7.18. Manual de la aplicación web – Dispensador – Temperatura	190
Figura 7.19. Manual de la aplicación web – Datos personales	190
Figura 7.20. Manual de la aplicación web – Ajustes – Tratamiento.....	191
Figura 7.21. Manual de la aplicación web – Ajustes – Toma de medicación	191
Figura 7.22. Manual de la aplicación web – Ajustes – Dispensador.....	192
Figura 7.23. Manual de la aplicación web – Ajustes – Notificaciones	192
Figura 7.24. Manual de la aplicación web – Ajustes - Toma de medicación.....	193

Figura 7.25. Manual de la aplicación web – Seriales 194
Figura 7.26. Manual de la aplicación web – Usuarios..... 194
Figura 7.27. Manual de la aplicación web – Datos personales administrador..... 195
Figura 7.28. Manual del dispensador – Acciones sobre el dispensador 196
Figura 7.29. Estructura del proyecto Node.js..... 199

Capítulo 1. Memoria del Proyecto

El proyecto “*Dispensador Inteligente de Medicamentos Conectado y Sistema de Gestión*” pretende facilitar la toma diaria de los pacientes en tratamiento, permitiendo programar tratamientos completos, además de asegurar la toma de estos a través de varios componentes que monitorizan el proceso.

Sin duda, el método más utilizado hoy día a la hora de tomar la medicación pasa por vincularla con alguna actividad cotidiana como puede ser el desayuno, la comida o acostarse. Algunos usuarios más adentrados en el mundo de la tecnología programan sus medicaciones ayudándose de alguna de las innumerables aplicaciones existentes desarrolladas con este objetivo o incluso mediante una simple aplicación de calendario. Pero estos son una minoría, puesto que la gran mayoría de la población no dispone ni de la tecnología apropiada para realizar este tipo de programaciones ni de los conocimientos tecnológicos para realizarla. Esto cobra aún más importancia en el caso de personas de avanzada edad, que además representan el grueso de la población que más medicación toma diariamente.

Existen ciertos dispositivos que permiten programar medicaciones diarias. Sin embargo, estos dispositivos tienen un gran número de restricciones y limitaciones que no los hacen realmente útiles. Uno de los instrumentos más utilizados para llevar el control de la toma de medicación es la clásica caja de pastillas compuesta por diversos compartimentos en los que se depositan las pastillas correspondientes a una toma y dividida en los días de la semana. Cada día suele estar compuesto por tres huecos, uno por cada toma de medicación.

El objetivo de este proyecto es crear un sistema novedoso y conectado que permita programar tratamientos de toma de medicación además de controlar remotamente las tomas.

El proyecto en cuestión está formado por dos partes bien diferenciadas; por un lado, el **dispensador de medicamentos** que contiene todos los componentes hardware además de ser el “*cerebro*” encargado de controlar todos los componentes físicos. Por otra parte, **el sistema de gestión** que permite, a través de una aplicación web, programar horarios de toma de medicamentos, controlar, monitorizar y configurar cualquier aspecto del dispensador y su entorno.

Además de la programación de tratamientos y su monitorización, el sistema también será capaz de controlar parámetros de su entorno como la temperatura y humedad o la detección de gases perjudiciales en el ambiente.

En los apartados sucesivos se describirán los medios utilizados para la ejecución del proyecto, los componentes, comunicaciones, diseño y construcción para su funcionamiento, así como otros apartados que servirán para comprender el desarrollo del trabajo, la toma de decisiones y cualquier otro aspecto importante.

1.1 Resumen de la Motivación, Objetivos y Alcance del Proyecto

El proyecto está orientado a construir y desarrollar un sistema automático que facilite y ayude en la toma de medicación diaria a la persona o personas que hagan uso del dispensador de medicamentos. La toma de medicación diaria al uso, sin ningún sistema automático, calendario o aplicación que recuerde el momento de su toma tiene un inconveniente de vital importancia: el olvido de una o de varias tomas diarias. Esto es especialmente común durante los primeros días del tratamiento, cuando se cambian las horas de la medicación, o cuando el usuario sale de la rutina diaria. Este problema se agrava en personas mayores más susceptibles de olvidar la toma de la medicación diaria.

Recordar el momento de cada una de las tomas de medicación diaria puede ser complicado. Además, esta situación es aún más preocupante cuando se trata de personas mayores susceptibles de olvidar cuándo deben tomar la medicación, lo cual, sumado a los innumerables medicamentos que deben tomar diariamente en algunos casos, hacen prácticamente inevitable el olvido de la toma de algún medicamento que el médico les ha recetado.

Por ello, se plantea la búsqueda de una solución automática que permita programar los horarios de varios tratamientos a través de un sistema de gestión web, con unas características amplias y funcionales que permitan abstraer el control de la toma de medicamentos al propio sistema en cuestión. Diseñando y desarrollando un sistema con materiales económicos, intuitivo y fácil de usar.

Puesto que la programación de toma de medicamentos será realizada por medio de un sistema de gestión web, no será necesario estar físicamente en el lugar en el que se encuentre el dispensador para programar horarios.

Además, no solo se pretende desarrollar un sistema que permita llevar el control de la toma de medicamentos, sino que también disponga de las funcionalidades necesarias que permitan conocer si la medicación correspondiente a una programación específica ha sido realmente tomada y funcionar como una estación de control ambiental. Esto es, que permita conocer la temperatura ambiental, humedad, detección de gases o si el dispensador ha recibido algún tipo de golpe o caída.

La monitorización ambiental no solo permite comprobar el entorno en el que convive el usuario del sistema, sino que también sirve para revisar el estado de los medicamentos y evitar su deterioro.

Un sistema de este tipo no serviría de mucha ayuda sin un sistema de notificaciones. Por este motivo, también se pretende incorporar un sistema de notificaciones que permita informar a los usuarios de un dispensador específico de las posibles anomalías en el entorno del pastillero o de si una medicación programada ha sido realmente tomada.

En lo referente al alcance del proyecto, se trata de construir un prototipo de dispensador de medicamentos que permita dispensar dos tipos de medicamentos distintos, o lo que es lo mismo, dos tratamientos diferentes. Para ello, el dispensador dispondrá de dos “ruletas” en las que se podrán insertar hasta un máximo de doce pastillas por cada una. Todo lo relativo a cada uno de los dos posibles tratamientos que gestionará el dispensador podrá ser configurado en el sistema de gestión web.

Para este proyecto, el prototipo de dispensador se limita a dos tipos de medicamentos distintos o tratamientos. En un futuro, el dispensador y el sistema en general podrá ser adaptado para convertirlo en un sistema modular en el que se puedan incorporar tantas “ruletas” como medicamentos distintos se necesita tomar. No obstante, el proyecto será desarrollado siguiendo este enfoque modular con el objetivo de que su adaptación futura sea lo mas sencilla posible.

Por último, existe un amplio espectro de usuarios potenciales de esta solución. Por un lado, los usuarios que necesitan tomar un gran número de pastillas diariamente o personas de avanzada edad cuyos encargados de controlar la medicación sean familiares o cuidadores, y que deseen automatizar el dispensado de medicamentos, garantizar su toma y verificar los factores ambientales de los alrededores del dispensador. Por otra parte, residencias y hospitales que deseen automatizar la toma de medicamentos de los pacientes. Para estos últimos, sería necesario escalar el sistema, posible gracias a su diseño modular, para disponer de todos aquellos medicamentos comunes que los usuarios pacientes deben tomar a diario.

1.2 Resumen de Todos los Aspectos

El sistema objeto del proyecto, estará formado por un dispensador de medicamentos capaz de expulsar las pastillas programadas en una fecha y a una hora concretos a través de una aplicación web desarrollada específicamente para este sistema.

El dispensador, además de expulsar pastillas a una fecha y hora programadas, también será capaz de controlar si la medicación fue retirada del dispensador, emitir notificaciones de emergencia iniciadas por el usuario, controlar el entorno ambiental del usuario y de la medicación, detectar caídas y emitir notificaciones luminosas y sonoras a través de ciertos componentes distribuidos por el dispensador.

Por su parte, desde la aplicación web, el usuario podrá programar horarios de toma de medicación en los que serán expulsadas las pastillas, monitorizar el estado del dispensador y pastillas que contiene, visualizar de forma gráfica datos en tiempo real del dispensador y ajustar sus parámetros de configuración.

El sistema también será capaz de mantener informados a los usuarios del dispensador sobre anomalías originadas en el mismo a través de correos electrónicos y mensajes instantáneos.

Capítulo 2. Introducción

2.1 Justificación del Proyecto

El sistema conectado que se pretende crear en este proyecto tiene como principal objetivo automatizar la toma de la medicación diaria y ayudar a las personas en este proceso, sobre todo a personas de avanzada edad.

Para ello, el sistema estará compuesto por un dispensador físico que será el encargado de dispensar las pastillas y de avisar mediante luces y sonidos de la toma de medicación y una aplicación web de gestión desde la que se programarán horarios y se monitorizará el dispensador y su entorno.

Además de la programación de horarios y su correspondiente dispensación, el sistema dispondrá de otras funcionalidades orientadas a su propia monitorización, accesible a través de la aplicación. La aplicación también permitirá realizar ajustes en el comportamiento del dispensador.

Otra funcionalidad de gran valor para el sistema es la emisión de notificaciones. El sistema conectado permitirá enviar notificaciones de anomalías en el dispensador a través de correos electrónicos y mensajes instantáneos a los usuarios del dispensador.

El principal público hacia el que irá dirigido el sistema es al de las personas mayores, debido a que son las más susceptibles y las que más dificultades presentan a la hora de tomar la medicación diaria principalmente por el olvido de su toma. No obstante, el sistema podría ser utilizado por cualquier otra persona que desee automatizar la toma de sus medicamentos o traspasar la responsabilidad del control de la medicación al sistema conectado.

En el mercado existen multitud de dispositivos que pretenden dar solución al problema que supone el olvido de la toma de medicación, pero la gran mayoría presentan un gran número de limitaciones y restricciones que los hacen poco útiles. Lo que se pretende con este proyecto es crear un sistema conectado novedoso que complete y extienda las funcionalidades de los actuales dispositivos existentes en el mercado y proporcione un dispositivo realmente útil, fácil de usar y, sobre todo, que contribuya a no olvidar la toma de medicación a sus usuarios.

2.2 Objetivos del Proyecto

Los principales objetivos del sistema conectado que será desarrollado en este proyecto y que servirán de base para la construcción del sistema son los siguientes:

- 1. Ayudar en la toma de medicación diaria y tratamientos.**
El sistema ayudará a no olvidar la toma de medicación diaria en tratamientos, además de enfocarlo hacia su uso por personas de avanzada edad.
- 2. Automatizar la toma de medicamentos y tratamientos.**
El sistema permitirá ayudar a automatizar la toma de medicamentos diaria por medio de la programación previa de tratamientos.
- 3. Diseñar y construir un prototipo de dispensador desde un enfoque interaccional.**
El dispensador de medicamentos estará conectado a Internet y será el encargado de dispensar las pastillas de los horarios programados además de perseguir un enfoque interaccional.
- 4. Desarrollar una aplicación web de gestión y programación de horarios.**
Otro de los principales componentes será la aplicación web de gestión desde la que se podrán programar horarios y gestionar el dispensador.
- 5. Monitorizar el entorno y el estado del dispensador.**
Desde la aplicación web se monitorizarán los datos del entorno del dispensador y su estado en relación a los medicamentos contenidos.
- 6. Monitorizar en tiempo real el estado de toma de medicación.**
Desde la aplicación web y a través de notificaciones, se podrá conocer en tiempo real el estado de toma de medicación de horarios programados, esto es, si las pastillas dispensadas correspondientes a una programación han sido tomadas o no.
- 7. Asegurar la recogida de las pastillas del dispensador.**
El dispensador asegurará que la medicación ha sido retirada del mismo cuando las pastillas hayan sido dispensadas.
- 8. Desarrollar un sistema completo de notificaciones sobre anomalías mediante correos electrónicos y mensajes instantáneos.**
Otro aporte de valor al sistema serán las notificaciones mediante correos electrónicos y mensajes instantáneos que permitirán estar al corriente de anomalías y eventos surgidos en el dispensador.

2.3 Estudio de la Situación Actual

2.3.1 Sistemas de programación de horarios y alarmas

Yendo un paso más allá del proceso y recordatorios clásicos a la hora de tomar la medicación diaria, existen numerosos sistemas que permiten insertar las pastillas correspondientes a cada toma en cada uno de los compartimentos del pastillero. En ellos, con la medicación insertada, el dispositivo permite programar alarmas que suenen a la hora programada y el usuario tome la medicación correspondiente.

Los inconvenientes de estos sistemas son numerosos. El usuario del pastillero es el encargado de coger la medicación del compartimento correspondiente, existiendo la posibilidad de que el usuario se confunda de compartimento y, por ende, de la medicación que toma. Otro de los principales inconvenientes de estos sistemas es que no permiten hacer un seguimiento en tiempo real por parte de terceras personas sobre si la medicación se está tomando o no.

Algunos de estos pastilleros se programan a través del propio dispositivo o por medio de una aplicación móvil, siendo necesario estar presente en el lugar donde se encuentra el pastillero para programar los horarios, además de necesitar un smartphone que será el encargado de notificar las alarmas correspondientes a los horarios programados. También cabe destacar que es estos sistemas no disponen de ningún tipo de mecanismo que garantice la toma de la medicación en los horarios programados.

2.3.1.1 Pastillero con alarma

La caja de pastillas con alarma unifica en un único dispositivo el almacenamiento de las pastillas para su toma, la programación de horarios y el centro de alarmas que notifica la toma de las pastillas contenidas en alguno de los compartimentos del dispositivo.



Figura 2.1. Pastillero con alarma

Al igual que los pastilleros convencionales, cada una de las columnas del pastillero hace referencia a un día de la semana. Cada una de las columnas contiene cuatro compartimentos asociados a distintos momentos del día y que suelen coincidir con el momento de toma de medicación.

Este dispositivo es de los más sencillos en cuanto a la programación de horarios de toma de pastillas. Permite programar fechas y horas de alarmas en el propio pastillero. La caja emite un sonido a la hora programada como recordatorio de toma de la medicación.

El principal inconveniente que presenta este pastillero es la independencia entre el programador y los compartimentos del dispositivo. Cuando la alarma suena, el usuario debe saber el día de la semana en el que se encuentra y el momento del día para acceder al compartimento correspondiente que contiene las pastillas. Este inconveniente se agrava si se trata de personas mayores quienes hacen uso de este pastillero.

Además, disponer de cuatro compartimentos por día de la semana también puede suponer un problema. Hay muchos usuarios que solo precisan realizar tres tomas por día o incluso menos. Con este pastillero, el usuario puede confundir el compartimento con el momento del día en el que se ha programado la alarma y tomar aquellas pastillas que no lo corresponden o acceder a un compartimento vacío pensando que ya tomo la medicación o que la tomo erróneamente con anterioridad.

2.3.1.2 Pastillero inteligente Pill-Buddy

Uno de los sistemas de pastillero inteligente para la toma de medicamentos es el *Pastillero inteligente Pill-Buddy*.



Figura 2.2. Pastillero inteligente Pill-Buddy

Este dispositivo dispone de seis compartimentos en los que almacenar los medicamentos, que junto con una aplicación móvil disponible para Android¹ e iOS², permite programar horarios de toma de medicación y controlar el contenido del mismo.

Además, se trata de un pastillero realmente económico, con un precio cercano a 10 EUR. Sin embargo dispone de un gran número de limitaciones.

El dispositivo no deja de ser un pastillero convencional, dejando toda la gestión de horarios, inventario y alarmas para la propia aplicación requiriendo su uso junto a un Smartphone. Otro inconveniente de este sistema es que el usuario debe conocer las pastillas que debe tomar en un momento concreto, o bien, consultarlas en la aplicación. Dejando abierta la posibilidad de que el usuario se equivoque con la medicación que debe tomar.

Por último, puesto que el pastillero no dispone de ningún sistema tecnológico ni microcontrolador, no permite conocer si la medicación ha sido realmente tomada.

2.3.1.3 Pastillero inalámbrico MedMinder

Otro sistema para la programación de horarios de toma de medicamentos compuesto por el pastillero y un sistema de gestión web es el pastillero inalámbrico *MedMinder*.



Figura 2.3. Pastillero inalámbrico MedMinder

Se trata de un sistema notablemente más sofisticado que los anteriores que permite programar horarios desde la web, sin necesidad de estar presente junto al pastillero.

El precio de este sistema no es precisamente económico, partiendo de 39.99 USD por mes de suscripción en su versión más económica. Además, únicamente está disponible para su contratación en los EE.UU.

¹ Android es un sistema operativo basado en el núcleo Linux. Fue diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes, tabletas y también para relojes inteligentes, televisores y automóviles.

² iOS es un sistema operativo móvil de la multinacional Apple Inc. Originalmente desarrollado para el iPhone (iPhone OS). No permite la instalación de iOS en hardware de terceros.

A pesar del alto precio de su suscripción mensual, presenta muchas limitaciones. Una de las más importantes es, de nuevo, la posible confusión a la hora de tomar la medicación. Dispone de numerosos compartimentos en los que se depositan las pastillas. El compartimento en el que se encuentran las pastillas que el usuario debe tomar se ilumina, aun así, deja la posibilidad que el usuario abra cualquiera de los compartimentos para tomar la medicación que no le corresponde.

A pesar de disponer de conexión a internet y permitir generar reportes estadísticos del pastillero, no permite monitorizar en tiempo real si la persona se está tomando las pastillas o garantizar que las pastillas se han recogido de su posición.

Un punto a favor de este dispositivo es que dispone de un sistema de notificaciones por mensajes de texto y correos electrónicos de anomalías en el pastillero. Aun así, no dispone de ningún tipo de mecanismo que permita garantizar si la medicación ha sido tomada o al menos, retirada del compartimento correspondiente.

2.3.2 Aplicaciones móviles de programación de horarios

Existen multitud de aplicaciones móviles orientadas a la programación de horarios de toma de medicamentos. Estas apps no son más que programadores de alarmas que notifican la toma de medicación previamente programada.

Este tipo de aplicaciones han sido diseñadas para funcionar independientemente del tipo de pastillero que se utilice para su organización, no precisando de ningún tipo de pastillero o dispensador para su uso.

2.3.2.1 Aplicación móvil *Medisafe*

Un ejemplo de este tipo de aplicaciones es *Medisafe*, aplicación con mayor número de descargas en el ámbito de la programación de horarios de toma de medicamentos para el sistema operativo Android.

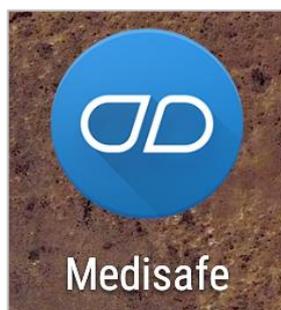


Figura 2.4. Icono aplicación Medisafe

Permite programar horarios de toma de pastillas, administrar las dosis de cada toma y añadir un gran número de medicamentos según su nombre comercial disponibles en su base

de datos, además de monitorizar ciertas medidas del usuario como la presión arterial. Para esta monitorización es necesario que el usuario introduzca los valores obtenidos para la medición que desea añadir.

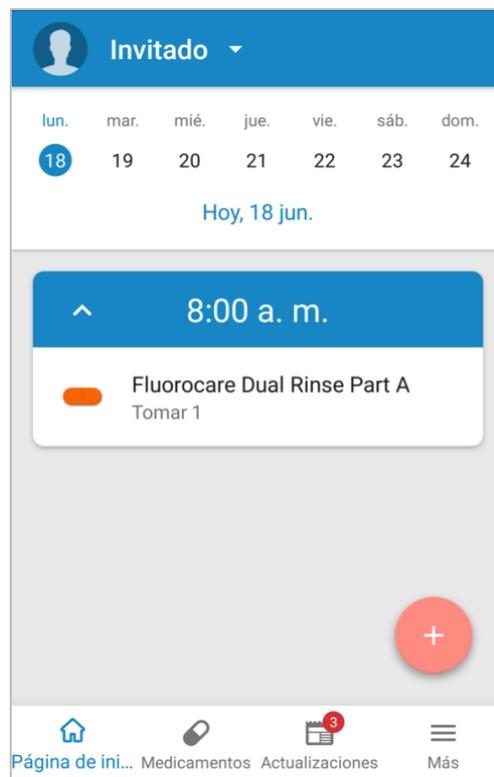


Figura 2.5. Pantalla de inicio de Medisafe

A pesar de que esta aplicación permite la programación de horarios en los que el usuario debe tomar medicación y llevar un control exhaustivo de distintas mediciones, no deja de ser una simple aplicación de alarmas y notificaciones. Además, debido a que la aplicación no ha sido desarrollada junto a un pastillero o dispensador, esta no hace obligatorio su uso, pero si un smartphone donde instalar la aplicación y que servirá como centro de alarmas.

El smartphone sería el encargado de avisar al usuario a la fecha y hora programadas para que tome la medicación, contabilizar el número de pastillas tomadas y cuando se tomaron. Además, se trata de una aplicación sencilla e intuitiva que permite escoger hasta la forma y color de las pastillas al añadir un nuevo medicamento.

2.3.2.2 Aplicación móvil MyTherapy

MyTherapy es otra de las aplicaciones móviles de ayuda en la toma de pastillas. Esta aplicación permite recibir notificaciones, administrar los medicamentos debidamente y controlar los síntomas del usuario.



Figura 2.6. Icono aplicación MyTherapy

Se trata de una aplicación muy completa que no funciona únicamente como programador de horarios, sino que también permite registrar medicamentos por su nombre o código de barras, monitorizar mediciones comunes como el colesterol o los triglicéridos o el control de síntomas del paciente.

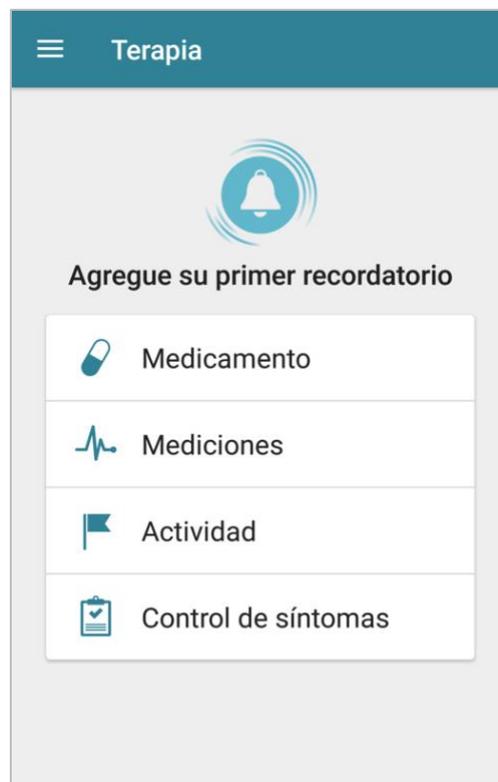


Figura 2.7. Pantalla de inicio de MyTherapy

A pesar de ser una aplicación menos completa que la anterior, permite de igual manera añadir medicamentos y programar su toma. La aplicación envía notificaciones de recordatorio de toma de medicación al smartphone el día y hora programados.

Esta aplicación también permite llevar el control de ciertas mediciones comunes en términos de salud y bienestar, el control de síntomas o el registro de actividades físicas para su monitorización.

Un aspecto novedoso de esta aplicación es que permite añadir medicamentos introduciendo el número del código de barras de la caja de medicamento o escanear directamente el código.

2.3.2.3 Aplicación móvil *Hora de la Medicina*

Otra de las aplicaciones más populares en el ámbito de la programación de horarios de toma de medicación para el sistema operativo Android es *Hora de la Medicina*.



Figura 2.8. Icono aplicación *Hora de la Medicina*

Al igual que las dos aplicaciones descritas anteriormente, permite añadir nuevos medicamentos y programar su toma. El usuario recibe notificaciones en el día y hora de la programación con información de las pastillas que deben tomarse.

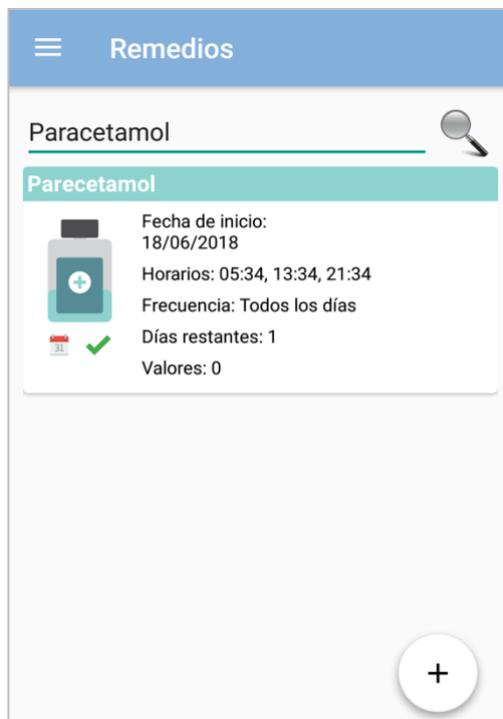


Figura 2.9. Pantalla de inicio *Hora de la Medicina*

Se trata de una aplicación menos intuitiva y mas difícil de usar debido principalmente al gran número de campos obligatorios que deben ser completados para programar una toma

de medicación. A esto se suma la apariencia obsoleta de la interfaz debido a que dispone de funciones, organización e iconos de versiones antiguas en Android.

Aun así, puede utilizarse como centro de alarmas, monitorizar la toma de medicamentos y contabilizar el número de pastillas restantes. Debido al gran número de parámetros y funciones de las que dispone, a pesar de dificultar su utilización y aprendizaje, proporciona otras muchas funcionalidades de las que otras aplicaciones de este ámbito carecen, tales como la programación de eventos (cita con el doctor) o asignar un medicamento a una patología concreta.

2.3.3 Conclusiones

Lo que se pretende con este proyecto es desligar el uso de un dispositivo electrónico “complejo”, como puede ser un *smartphone* o un ordenador, del dispensador encargado de gestionar los medicamentos y que será el centro de alarmas.

A pesar de que el objetivo de este proyecto es crear una aplicación web desde la que programar horarios de toma de medicamentos y un dispensador que reciba estas órdenes, el usuario que haga uso del dispensador no necesitará un dispositivo electrónico con acceso a internet para su uso diario. De esta manera, un supuesto de caso uso puede ser un familiar o cuidador de una persona de avanzada edad. El familiar o cuidador se encargaría de programar horarios de toma de medicamentos desde la aplicación web (sin requerir estar presente junto al dispensador) y la persona de avanzada edad únicamente requeriría recoger y tomar los medicamentos del dispensador cuando este se lo notifique (mediante elementos sonoros y visuales).

En lo referente a los sistemas inteligentes de toma de medicamentos vistos anteriormente, estos siguen un enfoque de pastillero con alarma. Existiendo un problema en este tipo de enfoques; a pesar de no ser necesario estar presente en el lugar del pastillero para programar horarios de toma de medicamentos (como en el caso del *Pastillero inalámbrico MedMinder*) ya que esta acción se realiza por medio de un sistema de gestión web, no resulta ser realmente útil. Antes de programar los horarios es necesario posicionar cada una de las pastillas en su compartimento correspondiente de día y toma, condicionando así su toma.

Además, dada la gran cantidad de compartimentos de los que disponen estos tipos de pastilleros, tanto el encargado de colocar las pastillas como el usuario que las toma pueden confundirse fácilmente.

Otro de los inconvenientes de este tipo de sistemas es que no permiten monitorizar las acciones del usuario con el pastillero. Y por ello, tampoco permiten conocer en tiempo real si la medicación ha sido tomada ni enviar notificaciones de medicación no tomada a los familiares o cuidadores.

El sistema que se pretende crear en este proyecto sigue un enfoque de dispensador; cada compartimento o “ruleta” del dispensador contendrá únicamente las pastillas de un tipo de medicamento y desde la aplicación web se programarán el número de pastillas de un tipo de medicamento (o de varios) que se desean dispensar en un día y hora concretos.

En este enfoque de dispensador de medicamentos tiene una mayor utilidad disponer de un sistema de gestión web que no requiera estar presente junto al dispositivo para la programación de horarios. Esto es posible gracias a que se programan horarios con tipos de medicamentos que serán dispensados y no con posiciones del pastillero que ya disponen de antemano de las pastillas que serán tomadas.

Otro aspecto importante que tendrá el sistema conectado y con el que no cuentan los pastilleros analizados, es la monitorización en tiempo real de las acciones del paciente sobre el dispensador por parte de los familiares o cuidadores a través de la aplicación web. El sistema será capaz de detectar si la medicación dispensada ha sido tomada y, en caso contrario, enviar alarmas a los familiares o cuidadores notificando la no toma de la medicación.

En resumen, el dispensador permite crear *“a la carta”* la toma de la medicación de una programación concreta de forma dinámica. Mientras que un pastillero únicamente permite programar un horario asignado a un compartimento que ya dispone de las pastillas antes de ser programado y que, por lo general, será rellenado a principio de semana.

También se extenderá la funcionalidad proporcionada por los pastilleros inteligentes vistos anteriormente, puesto que el sistema conectado permitirá monitorizar en tiempo real las acciones del usuario sobre el dispensador, su estado y entorno. El sistema también será capaz de detectar si una medicación programada ha sido tomada y en caso negativo, notificar a los familiares o cuidadores a través de mensajes instantáneos y correos electrónicos.

Capítulo 3. Análisis

Este capítulo contendrá toda la especificación de requisitos y funcionalidades junto a la documentación del análisis del sistema conectado y a partir de la cual se elaborará posteriormente el diseño del sistema.

3.1 Definición del Sistema

3.1.1 Determinación del Alcance del Sistema

En este proyecto se pretende diseñar y construir un prototipo de dispensador de medicamentos que permita expulsar pastillas según los horarios programados. También será capaz de monitorizar la toma de las medicaciones, controlar datos ambientales y el estado del dispensador e interactuar con él a través de botones. Las notificaciones del dispensador para la toma de medicamentos se realizarán por medio alarmas sonoras y luminosas.

Todos los horarios de programación, la monitorización en tiempo real de la toma de medicamentos, datos ambientales y estado del dispensador y otras configuraciones referentes al dispensador y a las notificaciones se realizará por medio de una aplicación web que también será creada para el proyecto. La aplicación permitirá el registro y la identificación de usuarios para el control, monitorización y programación de todos los aspectos del dispensador.

La identificación de anomalías en el dispensador y la no toma de medicamentos serán notificadas por medio de correos electrónicos y mensajes a los usuarios suscritos en la aplicación web.

Para este proyecto, el dispensador se limita a expulsar dos tipos de medicamentos distintos o llevar el control de dos tipos de tratamientos. En un futuro, tanto el dispensador como el sistema conectado podrá ser adaptado para su conversión a un sistema conectado modular y escalable, en el que puedan incorporarse componentes adicionales para dispensar tantos medicamentos distintos como se deban tomar. Por ello, el proyecto será desarrollado siguiendo este enfoque modular con el objetivo de que su futura adaptación sea lo más sencilla posible.

Además, el dispensador podrá contener un número máximo de pastillas predefinido para cada tratamiento. Cada pastilla tendrá su posición en dispensador según sea de uno u otro tratamiento.

3.1.2 Especificación textual de los requisitos del sistema

El proyecto consistirá en la construcción de un sistema conectado capaz de dispensar medicamentos de forma programada a través de una aplicación web. Para ello, se diseñará y

construirá un prototipo físico de dispensador de medicamentos en el que un microcontrolador será capaz de actuar sobre elementos electromecánicos para expulsar pastillas. Este prototipo será diseñado e impreso en 3D y alojará en su interior tanto el microcontrolador, capaz de dirigir todas las acciones del dispensador, como el resto de los componentes electrónicos y sensores utilizados para completar las funcionalidades de este.

El dispensador será capaz de controlar si la medición fuere retirada del mismo. También será capaz de notificar la toma de la medicación a través de avisos luminosos y sonoros. Dispondrá de otros componentes tales como botones con los que confirmar la toma de una medición o comunicar una situación de emergencia, detectar gases peligrosos o la caída del dispensador y también podrá medir la temperatura y humedad del ambiente.

Para que todas las acciones y datos recogidos del dispensador sean funcionales, se desarrollará un sistema distribuido capaz de enviar acciones a ejecutar por el dispensador y de recoger los datos recibidos de éste. Todos los datos del sistema serán almacenados en una base de datos cuyo acceso se restringe al servidor.

El envío de notificaciones a través de correos electrónicos y mensajes de Telegram³ de anomalías en el dispensador, desconexión o la no toma de medicamentos, es otra de las funcionalidades indispensables de este sistema. El sistema distribuido será el encargado de analizar las posibles anomalías de los datos recibidos del dispensador haciendo uso de la información almacenada en la base de datos.

Para completar el sistema, se diseñará e implementará una aplicación web de gestión del dispensador administrada por el servidor. Esta aplicación será la puerta de entrada a todas las acciones disponibles en el dispensador y en el propio sistema. A través de la aplicación, el usuario podrá programar horarios de toma de medicación y consultar el estado y horarios programados de cada uno de los pastilleros del dispensador. También podrá informarse de datos en tiempo real del dispensador y distribuciones horarias de estos datos a través de gráficos y visualización y modificar datos personales.

La aplicación también dispondrá de un completo sistema de configuración que permitirá al usuario modificar el tratamiento de cada pastillero, ajustar cuando se considera tomada la medicación en el dispensador, cómo se realizan las notificaciones de toma de medicamentos, correos electrónicos a los que se realizarán las notificaciones de pertinentes y otros ajustes de estas notificaciones que permitirán al usuario personalizar las alarmas en función a eventos surgidos en el dispensador.

Además, la aplicación permitirá el registro de nuevos usuarios, para lo que será necesario el número de serie del dispensador en cuestión. También contará con un panel de administración. Iniciando sesión como usuario administrador del sistema se podrán añadir nuevos dispensadores a través de su número de serie, eliminarlos y gestionar los usuarios registrados en el sistema.

³ Telegram Messenger es una aplicación de mensajería desarrollada desde el año 2013 por los hermanos Nikolái y Pável Dúrov. El servicio está enfocado en el envío y recepción de mensajes de texto y multimedia y es administrada por una organización sin ánimo de lucro cuya sede opera en Dubái, Emiratos Árabes.

Y por supuesto, la aplicación será desarrollada siguiendo la filosofía de desarrollo *mobile first*⁴, permitiendo su utilización desde cualquier dispositivo sin importar el tamaño de la pantalla del mismo.

3.1.3 Requisitos del Sistema

En la siguiente sección se describirán detalladamente todos los requisitos funcionales que especificarán las funcionalidades del sistema conectado, junto con la especificación de los requisitos no funcionales que detallarán las características de funcionamiento del sistema.

3.1.3.1 Requisitos Funcionales

Código	Nombre Requisito	Descripción del Requisito
RF-01	Registrar usuarios en el sistema	Se podrán registrar nuevos usuarios en el sistema. Para el registro, el usuario deberá introducir el serial de su dispensador que lo identifique inequívocamente y que se encuentra registrado en el sistema.
RF-02	Identificar usuarios cliente en el sistema	Se podrán identificar usuarios registrados en el sistema proporcionando el email de registro y la contraseña.
RF-03	Identificar usuarios administradores en el sistema	Se podrán identificar usuarios administradores en el sistema proporcionando el email y contraseña del usuario con privilegios de administración.
RF-04	Consultar estado del dispensador	El usuario cliente iniciado en sesión podrá consultar el estado del dispensador con datos de las pastillas tomadas, programadas y restantes.
RF-05	Programar días y horas a las que deben expulsarse las pastillas	El usuario cliente iniciado en sesión podrá programar horarios de toma de medicación a los que se expulsarán las pastillas correspondientes.
RF-06	Consultar horarios programados	El usuario cliente iniciado en sesión podrá consultar horarios programados para el dispensador.
RF-07	Eliminar horarios programados	El usuario cliente iniciado en sesión podrá eliminar horarios programados para el dispensador.
RF-08	Mostrar gráficamente el tiempo de vida del dispensador	El usuario cliente iniciado en sesión podrá consultar la distribución horaria de las conexiones del dispensador de las últimas 24 horas.
RF-09	Mostrar gráficamente el registro de temperaturas y humedades	El usuario cliente iniciado en sesión podrá consultar la distribución horaria de la temperatura y humedad del entorno del dispensador de las últimas 24 horas.
RF-10	Mostrar datos en tiempo real del dispensador	El usuario cliente iniciado en sesión podrá consultar datos en tiempo real de la temperatura, humedad, estado de conexión, detección de gas y caídas del dispensador.
RF-11	Modificar datos personales de registro	El usuario iniciado en sesión (cliente o administrador) podrá modificar sus datos

⁴ Modo de diseñar que tenga en cuenta, en primera instancia, un dispositivo móvil.

		personales salvo el identificador del dispensador.
RF-12	Especificar qué pastillas han sido introducidas en el dispensador	El usuario cliente iniciado en sesión podrá especificar el nombre y número de pastillas introducidas en el dispensador.
RF-13	Configurar el tiempo de espera que determina que no se ha tomado la medicación	El usuario cliente iniciado en sesión podrá configurar el tiempo de espera en minutos que determina si la medicación programada ha sido tomada.
RF-14	Configurar el modo en el que se considera tomada la medicación	El usuario cliente iniciado en sesión podrá configurar las acciones que deben realizarse en el dispensador para considerar tomada una medicación programada. Estas son; retirada de pastillas del dispensador y/o pulsación del botón de confirmación.
RF-15	Configurar la emisión de notificaciones de toma de medicación en el dispensador	El usuario cliente iniciado en sesión podrá configurar el modo en el que se emiten las notificaciones en el dispensador de toma de medicación mediante avisos sonoros y/o luminosos. También podrá configurarse el tiempo de sonido de la notificación.
RF-16	Añadir o eliminar correos electrónicos de notificación de alarmas	El usuario cliente iniciado en sesión podrá añadir y eliminar correos electrónicos a los que se notificarán anomalías en el dispensador y medicaciones programadas no tomadas.
RF-17	Suscribirse al canal de Telegram de notificaciones	El usuario cliente podrá suscribirse al canal de Telegram al que se enviarán mensajes de anomalías en el dispensador o medicaciones programadas no tomadas.
RF-18	Configurar notificaciones de alarmas a los emails y canal de Telegram	El usuario cliente iniciado en sesión podrá configurar las notificaciones de anomalías del dispensador que se envían.
RF-18.1	Activar o desactivar notificación de pérdida de conexión	El usuario cliente iniciado en sesión podrá activar o desactivar la notificación de pérdida de conexión del dispensador y el número de minutos que debe permanecer desconectado para que se considere pérdida de conexión.
RF-18.2	Activar o desactivar notificación de pocas pastillas restantes	El usuario cliente iniciado en sesión podrá activar o desactivar la notificación pocas pastillas restantes en el dispensador y el número de pastillas a partir del cual se realizará la notificación.
RF-18.3	Activar o desactivar notificación de anomalía en la temperatura	El usuario cliente iniciado en sesión podrá activar o desactivar la notificación de anomalía en la temperatura y establecer un mínimo y un máximo a partir de los cuales se considerará anomalía y se enviará la notificación.
RF-18.4	Activar o desactivar notificación de anomalía en la humedad	El usuario cliente iniciado en sesión podrá activar o desactivar la notificación de anomalía en la humedad y establecer un mínimo y un máximo a partir de los cuales se considerará anomalía y se enviará la notificación.
RF-18.5	Activar o desactivar	El usuario cliente iniciado en sesión podrá activar o

	notificación de detección de gas	desactivar la notificación de detección de gas en el entorno del dispensador.
RF-18.6	Activar o desactivar notificación de golpe del dispensador	El usuario cliente iniciado en sesión podrá activar o desactivar la notificación de detección de caída o golpe del dispensador.
RF-19	Añadir nuevos seriales al sistema	El usuario administrador iniciado en sesión podrá añadir nuevos seriales de dispensadores al sistema.
RF-20	Consultar seriales registrados en el sistema	El usuario administrador iniciado en sesión podrá consultar los seriales registrados en el sistema y la fecha y hora de la última conexión.
RF-21	Eliminar seriales del sistema	El usuario administrador iniciado en sesión podrá eliminar seriales del sistema siempre y cuando no disponga de usuario registrados.
RF-22	Consultar usuarios registrados en el sistema	El usuario administrador iniciado en sesión podrá consultar los usuarios registrados en el sistema junto al serial de registro.
RF-23	Eliminar usuarios del sistema	El usuario administrador iniciado en sesión podrá eliminar usuarios registrados en el sistema.
RF-24	Recoger y tomar medicación	El usuario del dispensador recogerá y tomará la medicación expulsada por el dispensador.
RF-25	Confirmar medicación tomada	El usuario dispensador confirmará que ha tomado la medición a través de un componente accionable en el dispensador.
RF-26	Emitir notificación de emergencia	El usuario del dispensador podrá emitir una notificación de emergencia a través de un componente accionable en el dispensador.
RF-27	Expulsar pastillas	El dispensador expulsará pastillas a las fechas y horas programadas por el usuario cliente.
RF-28	Registrar estado de dispensador	El sistema registrará el nuevo estado del pastillero cada vez que se dispensen pastillas programadas.
RF-29	Registrar que el dispensador está conectado a internet	Se registrará que el dispensador se encuentra conectado a internet cada minuto.
RF-30	Almacenar datos ambientales de temperatura y humedad	Se almacenarán datos ambientales de temperatura y humedad del dispensador cada minuto.
RF-31	Almacenar detección de gas o golpe del dispensador	Se almacenará el estado de detección de gas y caída o golpe del dispensador cada minuto.
RF-32	Enviar notificación por email y canal de Telegram cuando no se tome la medicación	El sistema enviará una notificación de alarma cuando no se tome una medicación programada a los emails de los usuarios registrado con el serial del dispensador y al canal de Telegram.
RF-33	Enviar notificación por email y canal de Telegram cuando se produzca una anomalía en el dispensador	El sistema enviará una notificación de alarma a los emails de los usuarios registrados con el serial del dispensador y al canal de Telegram cuando se produzca una anomalía recogida en el <i>RF-18</i> .
RF-34	Enviar notificación por email y canal de Telegram cuando se pulse el botón	El sistema enviará una notificación de alarma a los emails de los usuarios registrado con el serial del dispensador y al canal de Telegram cuando se pulse

	de emergencia del dispensador	el botón de emergencia del dispensador.
RF-35	Gestionar horarios de toma de medicación	El dispensador controlará el tiempo establecido para tomar la medicación dispensada.
RF-36	Emitir notificaciones sonoras y luminosas	El dispensador emitirá notificaciones sonoras y luminosas correspondientes a la toma de medicación programada. Notificaciones configuradas en el <i>RF-15</i> .
RF-37	Configurar dispensador	El sistema enviará datos de configuración al dispensador establecidos por el usuario cliente.

3.1.3.2 Requisitos No Funcionales

Requisitos de Usuario		
Código	Nombre del Requisito	Descripción del Requisito
RNF-1.1	Conocimientos previos	No se precisa de ningún conocimiento de específico previo para el uso del sistema conectado. Si bien es necesario ser capaz de hacer uso de un navegador, de una cuenta de correo electrónico y opcionalmente, del servicio de mensajería Telegram.

Requisitos Tecnológicos		
Código	Nombre del Requisito	Descripción del Requisito
RNF-2.1	Conexión a Internet del dispensador	Será necesario disponer de un router wifi con internet que de acceso a Internet al dispensador.
RNF-2.2	Navegador y conexión a internet del dispositivo	Será necesario disponer de un dispositivo móvil u ordenador con conexión a internet y un navegador instalado con el que acceder a la aplicación web.

Requisitos de Usabilidad y Otros		
Código	Nombre del Requisito	Descripción del Requisito
RNF-3.1	Diseño sencillo y visualmente atractivo	La aplicación web deberá tener un diseño austero, sencillo y bien formado en el que prevalezcan los componentes visuales.
RNF-3.2	Diseño responsivo	La aplicación web se adaptará a cualquier dispositivo y tamaño de pantalla, en los que se incluyen ordenadores, dispositivos móviles y tabletas.
RNF-3.3	Notificaciones	En la aplicación web deberá informar de todas y cada una de las acciones ejecutadas por el usuario.
RNF-3.4	Uso de colores seguros	En la aplicación web se deberá hacer uso de colores seguros con el objetivo de garantizar el uso correcto de la aplicación por parte de personas con daltonismo.
RNF-3.5	Impresión del prototipo en 3D	El prototipo debe ser impreso en su totalidad en 3D con el objetivo de crear un dispensador compacto y de poco peso.

RNF-3.5.1	Color del material de impresión	El color del material plástico utilizado para la impresión debe contrastar con el típico color blanco de los medicamentos.
RNF-3.6	Las dimensiones no superarán los 20cm	Las medidas del dispensador no superarán los 20cm de alto x 20cm de ancho y 20cm de largo. El objetivo de estas dimensiones es crear un dispositivo usable, con un tamaño aceptable.
RNF-3.7	Uso de pastillas de hasta 3cm	En el interior de los dos pastilleros del dispensador deberán poder alojarse pastillas de hasta 3cm. El objetivo es que el prototipo pueda dispensar cualquier pastilla por muy grande o pequeña que sea.
RNF-3.8	Las pastillas siempre serán dispensadas en la misma posición	El dispensador siempre depositará las pastillas en la misma posición del dispensador con el objetivo de acostumar al usuario del lugar de recogida de los medicamentos.

Requisitos de Seguridad		
Código	Nombre del Requisito	Descripción del Requisito
RNF-4.1	Control de errores	El sistema será capaz de manejar errores en las comunicaciones.
RNF-4.2	No considerar como dispensada la medicación en caso de desconexión del dispensador	Cuando el dispensador no permanezca conectado y disponga de horarios programados, estos serán ignorados y marcados como medicación no dispensada.
RNF-4.3	El material para la impresión del dispensador debe soportar el calor generado por los componentes electrónicos	El material plástico usado para la impresión del dispensador debe soportar el calor generado por los componentes electrónicos y especialmente por microcontrolador.
RNF-4.4	Solo permitir comunicaciones por HTTPS	Todos los componentes del sistema se comunicarán a través de internet por medio del protocolo HTTPS.
RNF-4.5	El sistema únicamente dispondrá de aquellos módulos requeridos para su uso	El sistema únicamente tendrá instalados aquellos módulos y paquetes externos que resulten indispensables para el correcto funcionamiento.
RNF-4.6	Almacenamiento de contraseñas en formato encriptado	Todas las contraseñas de los usuarios del sistema serán almacenadas en formato encriptado.
RNF-4.7	Cierre de sesión por inactividad	Las sesiones de los usuarios iniciados en el sistema serán cerradas después de 10 minutos de inactividad.

Requisitos de Eficiencia y Rendimiento		
Código	Nombre del Requisito	Descripción del Requisito

RNF-5.1	Frecuencia de solicitud de horarios	El dispensador solicitará horarios de toma de medicación cada minuto.
RNF-5.2	Frecuencia de refresco de la aplicación web	La aplicación se actualizará automáticamente cada minuto.
RNF-5.3	Disponibilidad 24x7 del sistema conectado	El sistema conectado debe estar disponible las 24 horas del día, los 7 días de la semana.

3.1.4 Identificación de Actores del Sistema

Los actores que interaccionan con el sistema conectado pueden ser bien actores primarios encargados de interactuar con el sistema o secundarios, sobre los que el sistema actúa. Los diferentes actores que intervienen en el sistema son:

- **Usuario anónimo:** es el usuario no registrado en la aplicación y que únicamente puede registrarse en el sistema o iniciar sesión.
- **Usuario cliente:** es el usuario identificado en el sistema como cliente y que puede programar, monitorizar, configurar y modificar todo lo relativo al dispensador correspondiente al número de serie con el que se registró en el sistema.
- **Usuario administrador:** es el usuario identificado en el sistema como administrador y que puede añadir, consultar, modificar y eliminar seriales de dispensadores y usuarios registrados.
- **Usuario del dispensador:** es el usuario que hace uso del dispensador de medicamentos y el encargado de tomar la medicación dispensada. También puede emitir eventos desde los ciertos componentes del dispensador.
- **Dispensador:** es el componente hardware encargado de dispensar los medicamentos, además de procesar, ejecutar, enviar y recibir un gran número de datos y parámetros del sistema.
- **Servidor de correo:** es el sistema encargado de enviar las notificaciones de alarmas a los correos electrónicos registrados para el serial del dispensador.
- **Mensajería Telegram:** es el sistema encargado de enviar mensajes de alarmas al canal de Telegram del sistema conectado.

3.1.5 Especificación de Casos de Uso

En esta sección se especifican los casos de uso del sistema y su interacción por parte de los actores vistos en la sección anterior. Los casos de uso están directamente relacionados con los requisitos del sistema y se detallarán por medio de diagramas y descripciones textuales.

Los casos de uso se organizan según el actor que toma parte en los mismos o sobre el que el sistema actúa en caso el caso de tratarse de actores secundarios.

3.1.5.1 Casos de uso del actor Usuario Anónimo

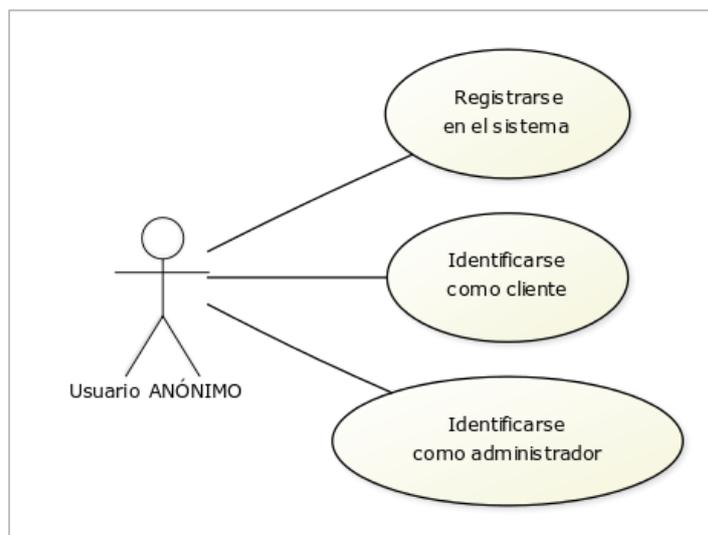


Figura 3.1. Casos de uso - Usuario Anónimo

Nombre del Caso de Uso	
Registrarse en el sistema	
Descripción	
El usuario anónimo introduce el serial de su dispensador, nombre, apellido, número de teléfono, email y contraseña. Con todos los campos validados (en especial, el serial del dispensador que debe estar registrado en el sistema) se registra el usuario en el sistema y se redirecciona al usuario al inicio de sesión.	

Nombre del Caso de Uso	
Identificarse como cliente	
Descripción	
El usuario anónimo y registrado introduce su email y contraseña, proporcionados cuando se registró en el sistema, para iniciar sesión y acceder al panel de gestión de su dispensador.	

Nombre del Caso de Uso	
Identificarse como administrador	
Descripción	
El usuario anónimo se identifica con un email y contraseña almacenados en el sistema como usuario con privilegios de administración y se redirige al usuario al panel de administración del sistema.	

3.1.5.2 Casos de uso del actor Usuario Cliente

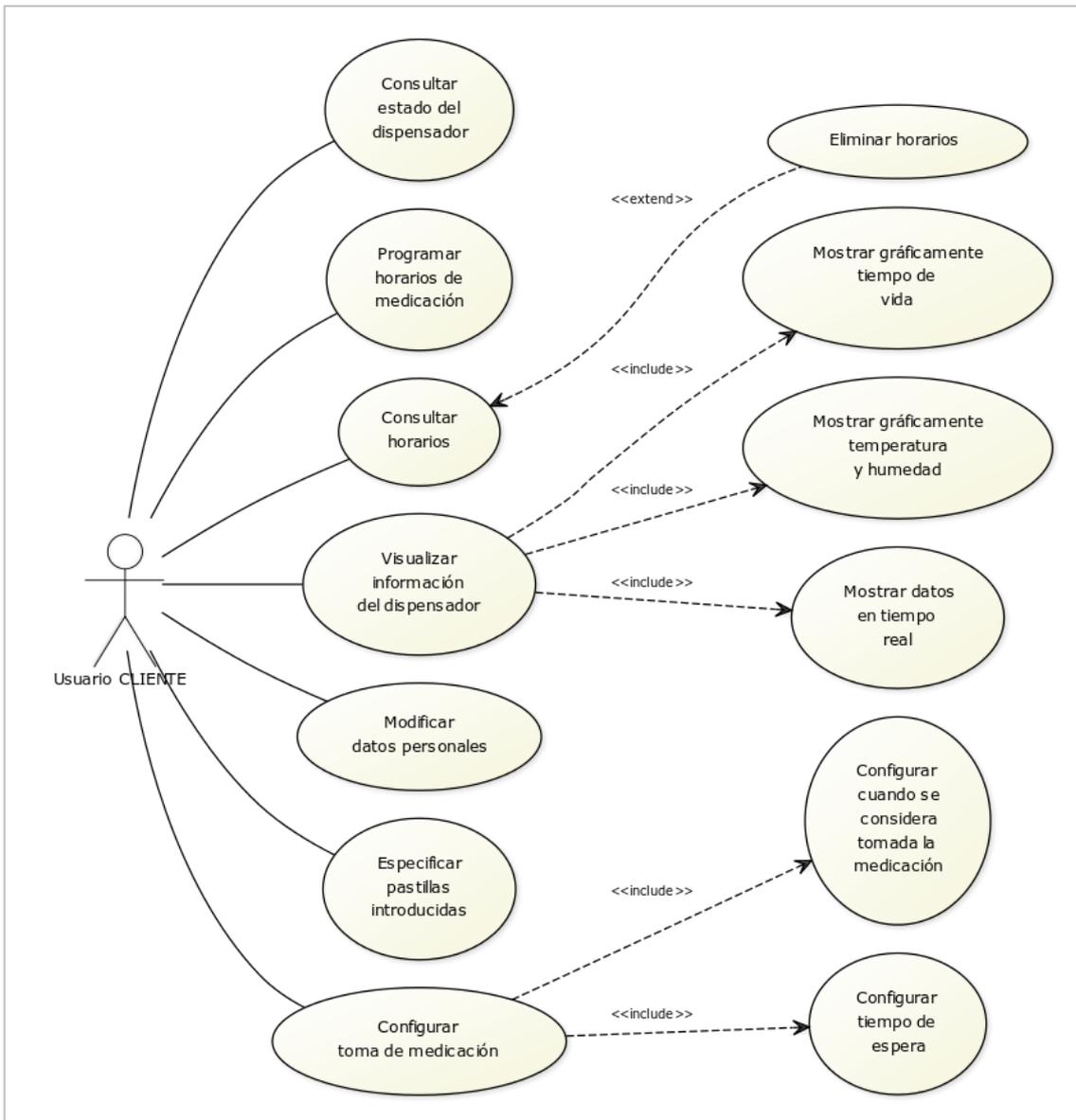


Figura 3.2. Casos de uso 1 - Usuario Cliente

Nombre del Caso de Uso	Consultar estado del dispensador
Descripción	El usuario cliente iniciado en sesión consulta los datos referentes al estado del dispensador. Entre estos datos se encuentra, el número de pastillas tomas/dispensadas, el número de pastillas programadas (que serán expulsadas próximamente) y el número de pastillas restantes que no han sido programadas. Todos estos números de pastillas van en relación al número de pastillas especificadas en el tratamiento.

Nombre del Caso de Uso	Programar horarios de toma de medicación
Descripción	

El usuario cliente iniciado en sesión selecciona el pastillero del dispensador que será programado, la fecha y hora a la que se dispensarán las pastillas y el número de pastillas que serán dispensadas. Opcionalmente podrá programar más de un horario a la vez añadiendo el número de horas que deben transcurrirse entre medicaciones. Y finalmente valida la programación contra el sistema.

Nombre del Caso de Uso

Consultar horarios

Descripción

El usuario cliente iniciado en sesión consulta los horarios programados de toma de medicación. Estos horarios muestran la fecha y hora a la que se dispensarán los medicamentos y el número de pastillas que serán dispensadas. Cada horario programado dispone de la opción de ser **eliminado**. Para ello el sistema emite una confirmación de seguridad que el usuario debe aceptar para finalmente eliminarlo.

Nombre del Caso de Uso

Visualizar información del dispensador

Descripción

El usuario cliente iniciado en sesión visualiza datos referentes al estado del dispensador e información ambiental. Los datos que visualiza son:

- **Datos en tiempo real** de la temperatura, humedad, estado de conexión y estado de detección de gas y golpes del dispensador.
- **Visualización** del número de conexiones del dispensador o **tiempo de vida** de las últimas 24 horas.
- **Gráficas** con la distribución horaria de la **temperatura** y **humedad** de las últimas 24 horas.

Nombre del Caso de Uso

Modificar datos personales

Descripción

El usuario cliente iniciado en sesión modifica sus datos personales introducidos durante el registro y entre los que se incluyen el nombre, apellido, teléfono, email y cambio de contraseña. El único campo que no puede modificarse es el serial del dispensador.

Nombre del Caso de Uso

Especificar pastillas introducidas

Descripción

El usuario cliente iniciado en sesión especifica el nombre y número de pastillas del tratamiento para cada uno de los dos pastilleros del dispensador. El número de pastillas del tratamiento solo puede ser modificado si no dispone de horarios programados y no ha dispensado ninguna pastilla (el pastillero se encuentra en la posición de partida).

Nombre del Caso de Uso

Configurar toma de medicación

Descripción

El usuario cliente iniciado en sesión configura los parámetros de toma de medicación que serán utilizados como valores de referencia para considerar tomada una medicación o no. Los valores que configura el usuario son:

- **Tiempo de espera** desde que la medicación ha sido dispensada hasta que se considera como no tomada la medicación.
- **Acciones** que debe realizar el usuario del dispensador **para considerar tomada la medicación**. Estas acciones son; la detección de la retirada de las pastillas y la confirmación mediante un componente accionable del dispensador.

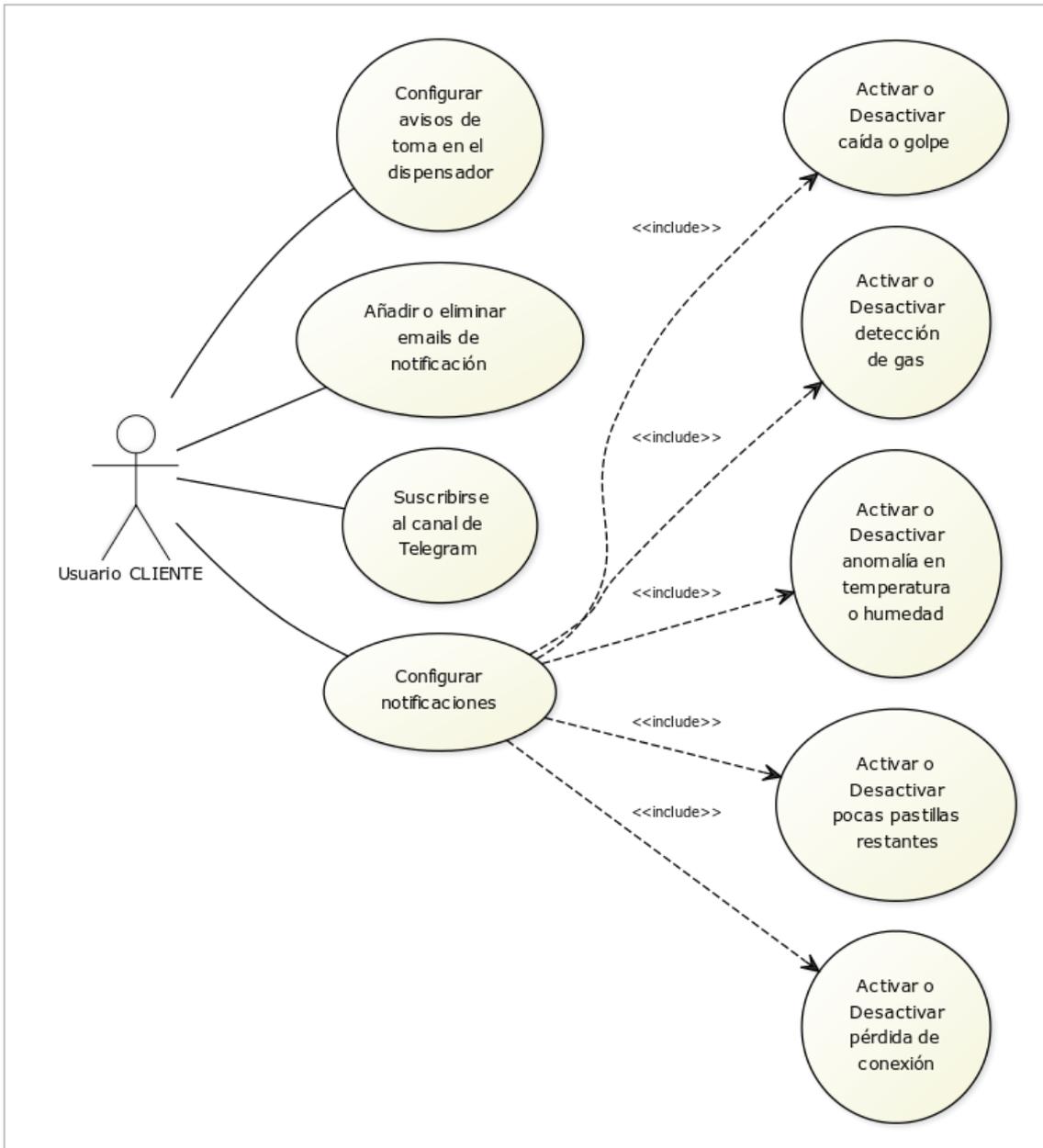


Figura 3.3. Casos de uso 2 - Usuario Cliente

Nombre del Caso de Uso
Configurar avisos de toma en el dispensador

Descripción
El usuario cliente iniciado en sesión configura la manera en la que se emiten notificaciones en el dispensador para la toma de la medicación. Estas son; alarmas sonoras y el tiempo en el que permanece sonando y alarmas luminosas.

Nombre del Caso de Uso
Añadir o eliminar emails de notificación
Descripción
El usuario cliente iniciado en sesión añade y elimina correos electrónicos extra a los que se realizan notificaciones de anomalías en el dispensador, la no toma de la medicación y emisión de alertas de emergencia por parte del usuario del dispensador.

Nombre del Caso de Uso
Suscribirse al canal de Telegram
Descripción
El usuario cliente iniciado en sesión se suscribe a las notificaciones de alarmas originadas en su dispensador a través de la aplicación de Telegram proporcionando el serial del dispensador. El usuario sigue los pasos que indica el canal de Telegram para su suscripción.

Nombre del Caso de Uso
Configurar notificaciones
Descripción
<p>El usuario cliente iniciado en sesión configura cuando se realizan notificaciones de alarmas del dispensador. O lo que es lo mismo, cuando el usuario considera anómalo un estado del dispensador para que se realicen notificaciones a los emails y canal de Telegram. Las configuraciones que puede realizar el usuario son:</p> <ul style="list-style-type: none"> • Activar o desactivar la notificación por golpe o caída del dispensador. • Activar o desactivar la notificación por detección de gas en el entorno del dispensador. • Activar o desactivar la notificación de anomalía en la temperatura o humedad. En el caso de activación, el usuario fija unos valores máximo y mínimo a partir de los cuales se considera anomalía y el sistema procede a enviar las notificaciones pertinentes. • Activar o desactivar la notificación de cantidad mínima de pastillas restantes en alguno de los pastilleros del dispensador. En caso de activación, el usuario especifica el número de pastillas a partir del cual se enviarán las notificaciones. • Activar o desactivar la notificación de pérdida de conexión del dispensador. En caso de activación, el usuario especifica el número de minutos que debe permanecer desconectado el dispensador para que se envíen las notificaciones pertinentes.

3.1.5.3 Casos de uso del actor Usuario Administrador

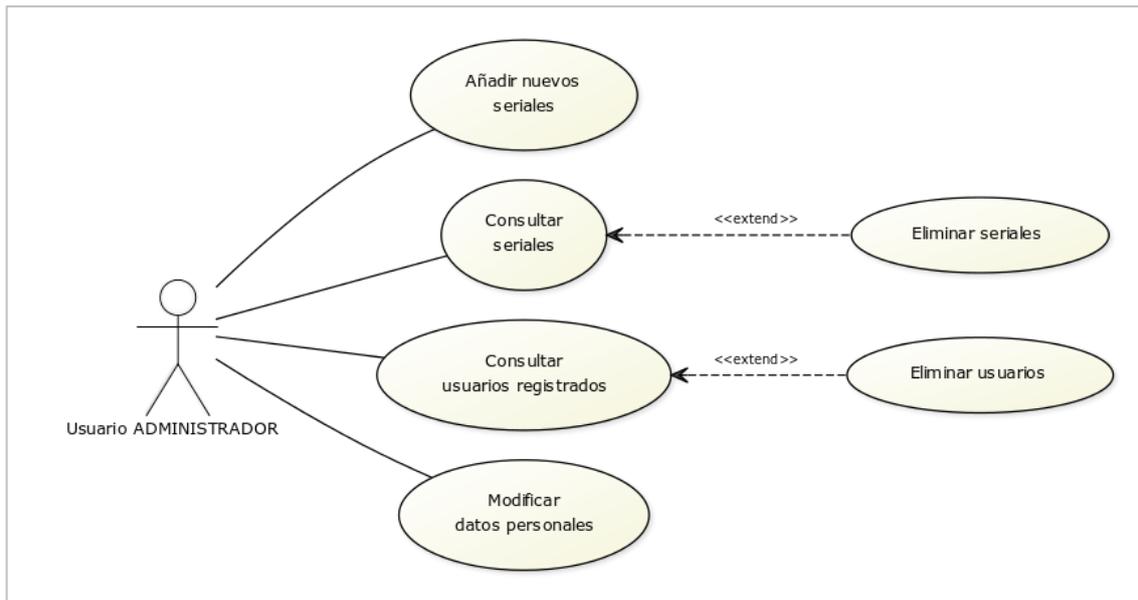


Figura 3.4. Casos de uso - Usuario Administrador

Nombre del Caso de Uso
Añadir nuevos seriales de dispensadores
Descripción
El usuario administrador iniciado en sesión añade nuevos seriales de dispensadores construidos para permitir el registro de nuevos usuarios cliente.

Nombre del Caso de Uso
Consultar seriales
Descripción
El usuario administrador iniciado en sesión consulta los seriales registrados, el número de usuarios registrados para cada serial y la fecha y hora de la última conexión de cada dispensador. Cada serial dispone de la de la opción de ser eliminado . Para ello, el sistema emite una confirmación de seguridad para su eliminación y comprueba que no dispone de usuario registrados, en caso contrario el administrador no podrá eliminar el serial.

Nombre del Caso de Uso
Consultar usuarios registrados
Descripción
El usuario administrador iniciado en sesión consulta los usuarios registrados; el email de cada usuario y el serial asociado al usuario. Cada usuario dispone de la opción de ser eliminado . Para ello, el sistema emite una confirmación de seguridad para su eliminación.

Nombre del Caso de Uso
Modificar datos personales
Descripción
El usuario administrador iniciado en sesión modifica sus datos personales entre los que se

incluyen el nombre, apellido, teléfono, email y cambio de contraseña. El usuario administrador no dispone de serial asociado a su cuenta.

3.1.5.4 Casos de uso del actor Usuario del Dispensador

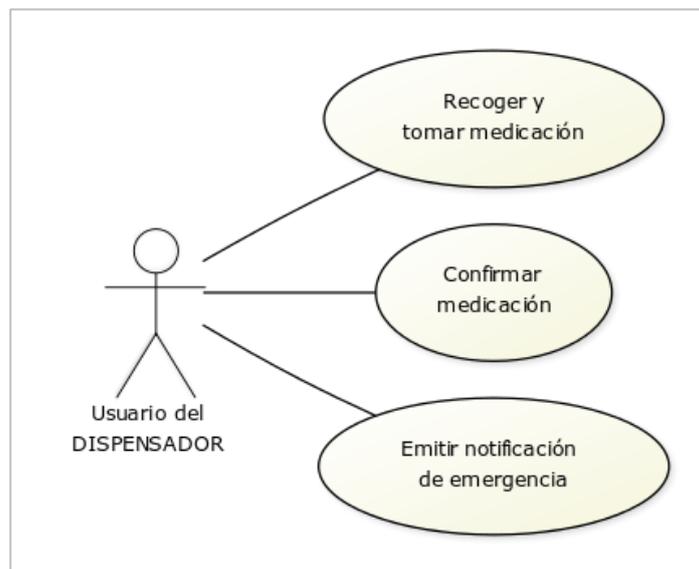


Figura 3.5. Casos de uso - Usuario del Dispensador

Nombre del Caso de Uso	
Recoger y tomar medicación	
Descripción	
El usuario o paciente que hace uso del dispensador retira las pastillas expulsadas por el dispensador para su toma. El dispensador detecta la retirada de las pastillas y realiza las acciones pertinentes.	

Nombre del Caso de Uso	
Confirmar toma de la medicación	
Descripción	
El usuario o paciente que hace uso del dispensador confirma al sistema la toma de la medicación a través un componente accionable disponible en el dispensador.	

Nombre del Caso de Uso	
Emitir notificación de emergencia	
Descripción	
El usuario o paciente que hace uso del dispensador emite una notificación de emergencia al sistema por medio de un componente accionable dispuesto en el dispensador.	

3.1.5.5 Casos de uso del actor *Dispensador* – Actor secundario

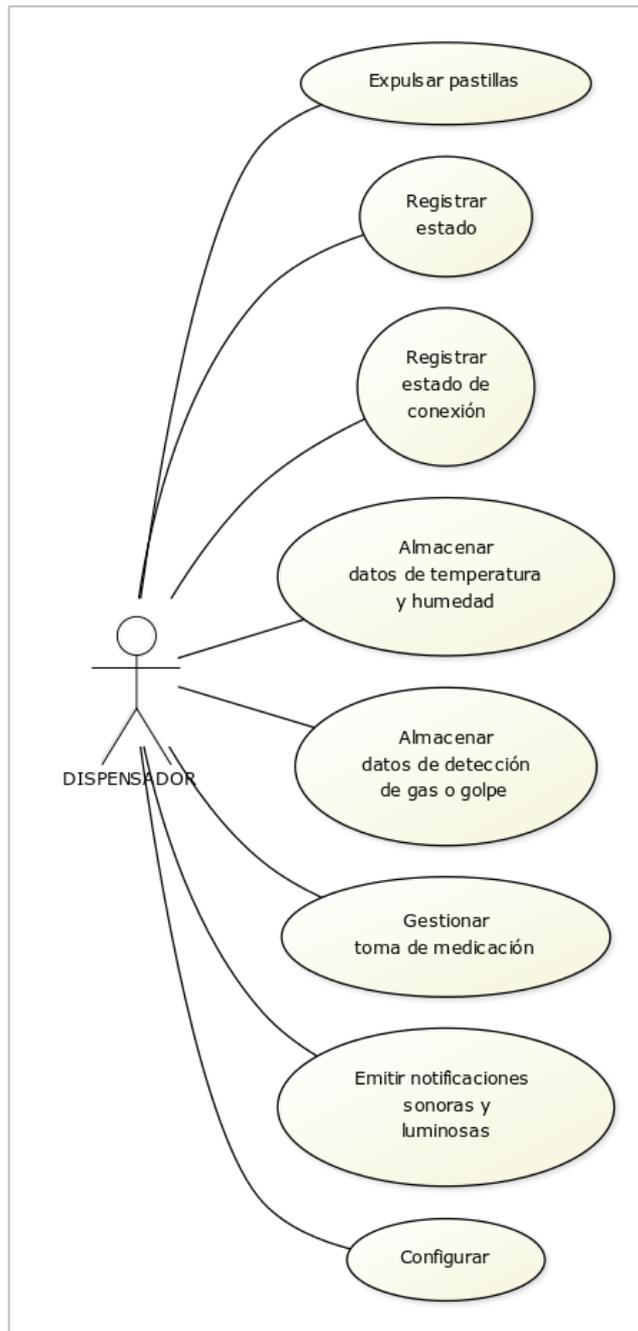


Figura 3.6. Casos de uso – Dispensador

Nombre del Caso de Uso	
Expulsar pastillas	
Descripción	
El dispensador expulsa las pastillas a la fecha y hora correspondientes a un horario programado por el usuario cliente.	

Nombre del Caso de Uso	
Registrar estado	

Descripción
El dispensador envía al sistema datos del nuevo estado de los pastilleros del dispensador para su registro cada vez que se expulsan pastillas de horarios programados.

Nombre del Caso de Uso
Registrar estado de conexión
Descripción
El dispensador realiza cada minuto peticiones al sistema y este registra dichas peticiones para controlar el estado de conexión del dispensador.

Nombre del Caso de Uso
Almacenar datos de temperatura y humedad
Descripción
El dispensador realiza peticiones cada minuto al sistema donde envía datos actuales de temperatura y humedad que el sistema almacena.

Nombre del Caso de Uso
Almacenar datos de detección de gas o golpe
Descripción
El dispensador realiza peticiones cada minuto al sistema donde envía datos del estado de detección de gases y golpes del dispensador que el sistema procesa y almacena.

Nombre del Caso de Uso
Gestionar toma de medicación
Descripción
El dispensador gestiona los tiempos de toma de medicaciones programadas para notificar al sistema en caso de superar el tiempo establecido sin haber tomado la medicación. El dispensador es el encargado de valorar si la medicación ha sido tomada o no.

Nombre del Caso de Uso
Emitir notificaciones sonoras y luminosas
Descripción
El dispensador emite alarmas sonoras y/o luminosas que notifican el requerimiento de toma de medicación para un horario programado.

Nombre del Caso de Uso
Configurar
Descripción
El dispensador recibe datos de configuración del sistema que establece en el dispensador. Estos datos son los configurados por el usuario cliente.

3.1.5.6 Casos de uso de los actores Servidor de correo y Telegram – Actores secundarios

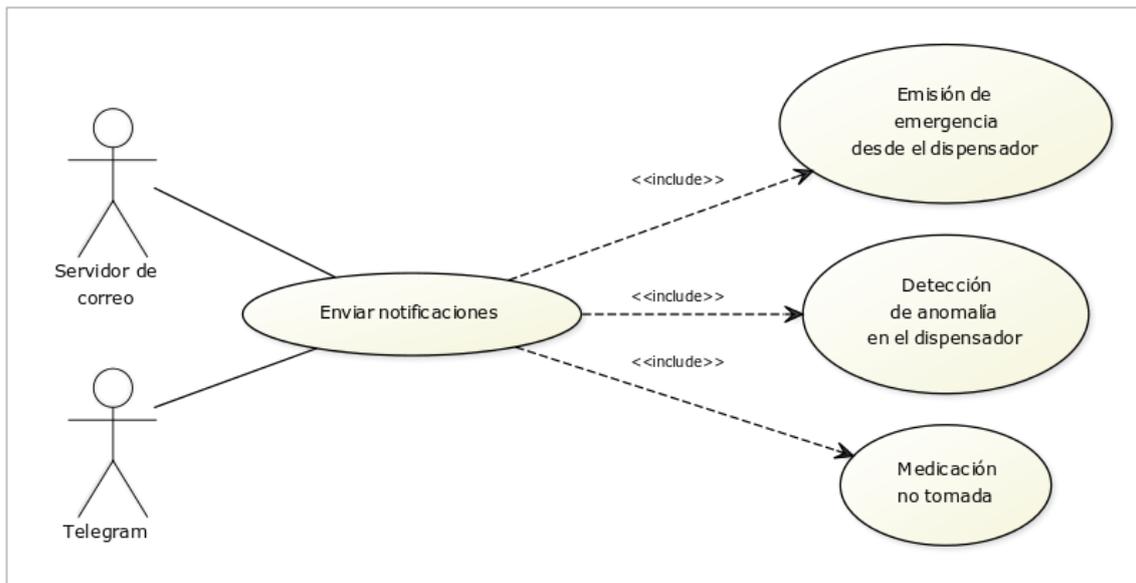


Figura 3.7. Casos de uso - Servidor de correo y Telegram

Nombre del Caso de Uso
Enviar notificaciones
Descripción
<p>El sistema, a través del servidor de correo y el servicio de mensajería Telegram, envía notificaciones de eventos surgidos en el dispensador a través de correos electrónicos y mensajes. Estas notificaciones se realizan por los siguientes eventos:</p> <ul style="list-style-type: none"> • Emisión de emergencia desde el dispensador e iniciada por parte del usuario del dispensador. • Detección de anomalía en el dispensador. Esta anomalía puede ser por pérdida de conexión, número mínimo de pastillas restantes, temperatura, humedad, detección de gas o golpe del dispensador. • Medicación no tomada por el usuario del dispensador. Superado el tiempo de espera sin recoger las pastillas del dispensador y/o confirmar la toma.

3.2 Identificación de los Subsistemas en la Fase de Análisis

En esta sección se analizan y describen los subsistemas en los que se descompone el sistema conectado y las necesidades de comunicación entre subsistemas para abordar todas las funcionalidades descritas en el catálogo de requisitos.

3.2.1 Descripción de los Subsistemas

Con los requisitos del sistema definidos y los casos de uso de especificados, es necesario analizar los subsistemas de los que se compondrá el sistema conectado para abordar todas las funcionalidades requeridas para el proyecto.



Figura 3.8. Diagrama conceptual

La programación de horarios, la monitorización del dispensador y de la toma de medicamentos junto a un completo sistema de notificaciones, son algunas de las funcionalidades principales que abarca este proyecto.

En la imagen anterior se ilustra el sistema completo dividido en los principales subsistemas de los que se compone. Los subsistemas y su funcionalidad dentro del sistema conectado son los siguientes:

- **Subsistema 1: Aplicación web de control y gestión**

La aplicación web de control y gestión será puerta de entrada a todas las acciones disponibles en el sistema y monitorización del dispensador. A través de ella, el usuario podrá programar horarios de toma de medicación, monitorizar el estado del dispensador y parámetros ambientales y configurar un gran número de ajustes del dispensador, entre otras acciones. A través de la aplicación también se podrán programar notificaciones realizadas por medio de correos electrónicos y de un canal de Telegram creado específicamente para este fin. El objetivo de estas notificaciones es informar a los usuarios de anomalías en el dispensador.

La aplicación también contará con una sección de administración desde la que el usuario con privilegios podrá gestionar tanto los seriales de dispensadores registrados

en el sistema como los usuarios asociados a cada uno. Además, podrá añadir nuevos números de serie de dispensadores.

- **Subsistema 2: Dispensador de pastillas conectado**

El dispensador de pastillas conectado será el instrumento hardware que será diseñado y construido para el proyecto. En él se encontrará alojado un microcontrolador capaz de conectarse a Internet e interactuar con el sistema. Además del microcontrolador, el interior del dispensador albergará distintos componentes electrónicos y sensores cuyo objetivo es dispensar los medicamentos, el control ambiental del dispositivo, detectar anomalías, gestionar notificaciones de toma de medicamentos o ser capaz de garantizar la toma de pastillas dispensadas entre otros.

El microcontrolador representará el núcleo del dispensador y será el encargado de orquestrar todos los componentes electrónicos, mecánicos y sensores conectados a él, además de realizar peticiones al servidor del sistema.

- **Subsistema 3: Base de datos**

En la base de datos se almacenarán todos los datos recabados tanto del dispensador de pastillas conectado como de la aplicación web de control de forma persistente. Entre estos datos se encuentra la información de usuarios, programaciones de horarios, datos del dispensador o configuraciones. La base de datos únicamente será accesible desde el servidor del sistema.

Todos los subsistemas serán los responsables de trabajar en conjunto para proporcionar la funcionalidad descrita en el catálogo de requisitos del sistema.

3.2.2 Descripción de los Interfaces entre Subsistemas

Para que todos los subsistemas puedan trabajar en conjunto necesitan comunicarse. Todos los subsistemas identificados se encontrarán físicamente en distintas localizaciones, por lo que la única forma de comunicación posible es a través de Internet.

El servidor será el elemento encargado procesar las solicitudes de las peticiones realizadas tanto por el navegador del cliente donde se renderiza la aplicación web de control como por el microcontrolador del dispensador de pastillas conectado. También será el encargado de comunicarse con la base de datos para obtener o almacenar información.

El sistema conectado puede tratarse como un **sistema distribuido** debido a que la aplicación web se encontrará desplegada en una máquina diferente a la que se encuentra la base de datos y la comunicación entre ambos se realiza por internet.

A la vista de la imagen de la sección anterior, la comunicación entre los subsistemas que componen el sistema completo puede resumirse en dos interfaces:

- **La comunicación entre la base de datos y el servidor en el que se encuentra desplegada la aplicación web.** Como ya se mencionó anteriormente, la aplicación web

se encontrará desplegada en una máquina distinta a la que se encuentra la base de datos. Por ello, la comunicación entre el subsistema de aplicación web y la base de datos se realizará por red. La aplicación será conocedora de la dirección y credenciales de acceso para realizar peticiones a la base de datos.

- La **comunicación entre el dispensador de pastillas conectado y el servidor en el que se encuentra desplegada la aplicación web**. El microcontrolador del dispensador será el encargado de realizar peticiones al servidor. Para que esta comunicación sea efectiva, el servidor dispondrá de una API REST de acceso restringido al dispensador de pastillas conectado. Así, el dispensador realizará peticiones a la API REST del servidor y éste será el encargado de responder al dispensador con la información solicitada y el resultado del procesamiento de la petición.

3.3 Análisis del Dispositivo dispensador de medicamentos

El dispensador de medicamentos conectado será impreso en 3D y contará con **dos ruletas o pastilleros** en los que se alojarán las pastillas. La idea es que cada uno de los dos pastilleros disponga únicamente de pastillas de un tipo de medicación. De este modo, el dispensador podrá expulsar dos tipos diferentes de pastillas.

Las pastillas serán dispensadas siempre en la misma posición del dispensador. El objetivo de dispensar siempre las pastillas en la misma posición del dispensador es acostumbrar al usuario del lugar de recogida de la medicación.

En cuanto a los componentes electrónicos, mecánicos y sensores que deben ser utilizados en el dispensador para abordar la funcionalidad del dispositivo definida en el catálogo de requisitos son los siguientes:

- **Microcontrolador con WiFi:** será el encargado de establecer conexión a internet y de orquestar todos los componentes electrónicos, electromecánicos y sensores del dispensador.
- **Dos motores que permitan dispensar las pastillas:** cada una de las *ruletas* o pastilleros dispondrá de un motor que permitirá rotar el componente para dispensar pastillas de forma controlada.
- **Un LED de notificación de toma de pastillas:** este led se encenderá cuando el usuario del dispensador deba tomar medicación. Servirá como elemento notificador luminoso de toma de pastillas.
- **Un emisor de sonido de notificación de toma de pastillas:** este componente emitirá sonido cuando el usuario del dispensador deba tomar medicación. Servirá como elemento notificador sonoro de toma de pastillas.
- **Un LED de notificación de conexión a Internet:** este led permanecerá encendido siempre y cuando el microcontrolador del dispensador disponga de conexión a internet.
- **Un sensor que permita conocer si la medicación ha sido retirada del dispensador:** será necesario un sensor o componente electrónico que permita garantizar que la medicación dispensada de un horario de programación ha sido retirada del dispensador.
- **Un elemento accionable de confirmación de toma de medicación:** el dispensador tendrá un elemento accionable (probablemente un botón) que permitirá al usuario del dispensador confirmar la toma de la medicación.
- **Un elemento accionable de emisión de emergencia:** el dispensador tendrá un elemento accionable (probablemente un botón) que permitirá al usuario del dispensador emitir una notificación de emergencia.
- **Un sensor que detecte un golpe en el dispensador:** se requería un sensor que sea capaz de detectar un golpe o caída del dispositivo dispensador.

- **Un sensor que permita conocer la temperatura y humedad ambiental:** será necesario un sensor que permita al microcontrolador medir la temperatura y humedad ambiental del dispensador.
- **Un sensor que permita detectar gases en el ambiente:** el dispensador tendrá un sensor que permitirá detectar gases en el ambiente que puedan ser perjudiciales para el usuario del dispensador.
- **Un transformador para alimentar el microcontrolador:** para alimentar el dispensador será necesario un transformador que proporcione corriente de entrada acorde a los requisitos del microcontrolador y los elementos conectados al mismo.

Además, dado el gran número de componentes electrónicos, electromecánicos y sensores requeridos para completar la funcionalidad del dispensador, será necesario algún tipo de placa adaptable al microcontrolador que permita organizar y establecer todas las conexiones entre los componentes y el microcontrolador.

Con todos los componentes electrónicos y sensores descritos anteriormente, se completan todas las funcionalidades descritas en el catálogo de requisitos del sistema para el dispositivo dispensador.

3.4 Análisis y estudio de Alternativas

En este capítulo se presenta un estudio de las alternativas propuestas para cada uno de los subsistemas y elementos que componen el sistema conectado completo objeto del proyecto.

Para cada alternativa propuesta, además de una descripción, se valorarán ventajas e inconvenientes de su uso en este proyecto y cualquier otro aspecto importante que sea necesario valorar o influya en el mismo.

3.4.1 Sistema Distribuido y Base de Datos - Backend

El entorno de ejecución usado para el desarrollo de la capa del servidor y la base de datos donde se almacena toda la información relativa a la aplicación desarrollada, son dos de los componentes principales en la creación de cualquier aplicación o sistema web.

En este apartado se valorarán algunos de los entornos de ejecución de la capa de servidor mas utilizados y bases de datos que podrían resultar ser una alternativa como subsistema de desarrollo de la capa del servidor y de la base de datos.

3.4.1.1 Alternativas de desarrollo de la capa del Servidor

3.4.1.1.1 Java y Spring como Framework

Spring es un framework para el desarrollo de aplicaciones web multiplataforma, de código abierto para el lenguaje de programación Java. Se trata de un framework lanzado hace mas de 14 años, lo que conlleva que ya se encuentre más que asentado en el mercado y con una gran comunidad de desarrolladores que lo respalda.



Figura 3.9. Logo framework de desarrollo Spring

Algunos de las ventajas de usar Spring son; el framework hace uso el patrón MVC (modelo-vistas-controlador), al estar basado en el lenguaje de programación Java dispone de un fuerte tipado y además de tratase de un framework muy potente.

No obstante, los inconvenientes que podrían resultar de usar Spring en este proyecto son numerosos; este framework no esta orientado a la creación de APIs (la comunicación entre el microcontrolador del dispensador y el servidor se realizará mediante APIs REST⁵). A pesar de que la lógica se organiza jerárquicamente de forma perfecta, añadir nuevas funcionalidades y extender el proyecto resulta ser complejo y requerir demasiado tiempo. Además, es necesario escribir su propio ecosistema de despliegue.

Este framework se puede caracterizar por la obligación de generar una estructura de desarrollo muy bien jerarquizada, lo que requiere emplear mucho tiempo de desarrollo. Además, Spring no esta pensado para la creación de APIs (requisito fundamental para este proyecto).

3.4.1.1.2 Node.js

Node.js es un entorno de tiempo de ejecución multiplataforma de código abierto para la capa del servidor basado en el lenguaje de programación ECMAScript⁶, con una arquitectura orientada a eventos y basada en el motor V8 de Google.



Figura 3.10. Logo del entorno Node.js

Se trata de un entorno de alto rendimiento, asíncrono, simple y altamente escalable. Hace uso del gestor de paquetes NPM⁷ para su funcionamiento y dispone de un gran número de paquetes y librerías para ampliar sus funcionalidades.

Esta orientado a la creación de sistemas en red modulares y pensado para el desarrollo y soporte de APIs, además de disponer de una enorme comunidad de desarrolladores. También cabe destacar la facilidad con la que se puede iniciar y ejecutar un proyecto desarrollado en este entorno.

⁵ La transferencia de estado representacional (en inglés representational state transfer) o REST es un estilo de arquitectura software para sistemas hipermedia distribuidos como la World Wide Web.

⁶ ECMAScript es una especificación de lenguaje de programación publicada por ECMA International.

⁷ NPM es el gestor de paquetes por defecto para Node.js, un entorno de ejecución para JavaScript.

Como único inconveniente podríamos mencionar que la organización jerárquica del desarrollo de un proyecto en Node.js no resulta ser del todo sencilla. Esto puede conllevar la mala práctica de no organizar correctamente los ficheros de desarrollo del proyecto dando lugar a un sistema difícil de mantener.

3.4.1.2 Alternativas de Base de Datos

Actualmente, las bases de datos se pueden categorizar en dos grandes tipos según su estructura; relacionales y no relacionales. A pesar de la tendencia actual por la utilización de bases de datos no relacionales, el uso de un tipo de base de datos u otro está condicionado plenamente al tipo de proyecto en el que se quieran utilizar.

En este apartado, se estudiarán las ventajas e inconvenientes que ofrece usar cada uno de estos tipos de bases de datos.

3.4.1.2.1 Base de datos relacional – MySQL

Las bases de datos relacionales o SQL utilizan un modelo relacional para su organización. Debido a que este tipo de bases de datos llevan un largo tiempo en el mercado, disponen de un gran soporte y *add-ons* para su gestión. Se pueden configurar para el control de la atomicidad⁸ de las operaciones y los datos deben cumplir los requisitos de integridad impuestos por la estructura de la base de datos.

Uno de los sistemas de gestión de bases de datos relacionales más utilizadas en desarrollo web es MySQL.



Figura 3.11. Logo base de datos relacional MySQL

MySQL es la base de datos de código abierto más popular del mundo desarrollada bajo la licencia de *Oracle Corporation*.

Las bases de datos relacionales también presentan una serie de desventajas, como el hecho de que la atomicidad de las operaciones juegue un papel fundamental en su

⁸ La atomicidad es la propiedad que asegura que una operación se ha realizado o no, y por lo tanto ante un fallo del sistema no puede quedar a medias.

rendimiento. Además, otro de los mayores inconvenientes de este tipo de bases de datos es su escalabilidad, a pesar de ser usadas en muchos entornos productivos.

Es necesario tener definida la estructura de la base de datos y sus relaciones antes de comenzar el desarrollo del proyecto, ya que modificar su estructura y/o sus relaciones suele ser un proceso tedioso, largo y complicado.

3.4.1.2.2 Bases de datos no relacionales – MongoDB

Las bases de datos no relaciones o NoSQL, no utilizan un modelo relacional para su organización. Este tipo de bases de datos presentan muchas ventajas, siendo algunas de las importantes su facilidad de escalabilidad debido a su carácter descentralizado, son bases de datos mucho más abiertas y flexibles y permiten hacer cambios de los esquemas sin tener que parar las bases de datos y permiten la escalabilidad horizontal.

El sistema de gestión de bases de datos NoSQL más utilizado en el mundo es sin duda MongoDB.



Figura 3.12. Logo de base de datos no relacional MongoDB

MongoDB es un sistema de bases de datos no relacionales orientado a documentos y desarrollado en código abierto. Guarda estructuras de datos en documentos similares a JSON⁹ con un esquema dinámico, permitiendo que la integración de los datos con las aplicaciones sea más fácil y sencilla.

Las bases de datos no relacionales también presentan una serie de desventajas, que dependiendo el tipo de proyecto donde se pretendan integrar, pueden ser mas o menos críticas. No todas las bases de datos NoSQL contemplan la atomicidad y la integridad de los datos, presentan problemas de compatibilidad entre instrucciones SQL y presentan una falta de estandarización.

3.4.2 Dispensador, Microcontrolador y Componentes Electrónicos

La construcción del prototipo de dispensador requiere de una cantidad considerable de componentes electrónicos que deben trabajar en conjunto con el microcontrolador para un correcto funcionamiento y poder alcanzar el objetivo para el que se será construido.

⁹ JSON, acrónimo de JavaScript Object Notation, es un formato de texto ligero para el intercambio de datos.

En las siguientes secciones se evaluarán distintos enfoques de diseño del prototipo, alternativas de microcontroladores y componentes electrónicos y mecánicos necesarios para su construcción.

3.4.2.1 Alternativas de Dispensador

El prototipo de dispensador puede ser diseñado y desarrollado siguiendo uno de los dos enfoques evaluados a continuación. Cada uno de estos enfoques tiene una serie de ventajas e inconvenientes.

3.4.2.1.1 Enfoque de pastillero

El enfoque de pastillero persigue la construcción de un prototipo siguiendo el diseño de un pastillero tradicional. En cada uno de los huecos se alojarían las pastillas correspondientes a una toma de un día y hora concretos.

El microcontrolador se encargaría de notificar mediante sonidos y avisos luminosos de la toma de medicación. También controlaría que las pastillas del hueco correspondiente fueron retiradas y otros muchos aspectos objeto de este proyecto.



Figura 3.13. Enfoque de pastillero

Este enfoque presenta una serie de limitaciones que dificultan su adaptación en este proyecto. El objetivo de este trabajo es crear un sistema inteligente que permita programar horarios de toma de medicamentos desde una aplicación web y sin necesidad de estar presente junto al prototipo. Como se ha mencionado, cada uno de los huecos del pastillero contendría las pastillas de una toma de un momento del tiempo.

Dado que el pastillero dispondría de todos los medicamentos organizados secuencialmente para toda la semana, no serviría de mucha utilidad la opción de poder programar los horarios desde una aplicación web.

El hecho de rellenar el pastillero con las pastillas correspondientes a cada una de las tomas diarias de la semana condiciona su toma y su programación, siendo prácticamente innecesario programar los horarios desde una aplicación web pudiendo programarlos desde una aplicación conectada al supuesto microcontrolador del pastillero (siendo necesario estar presente junto al pastillero) el día que se rellene.

Siguiendo este enfoque, no dejaría de ser un pastillero con alarma precondicionado a la posición que ocupa cada uno de los huecos y a las pastillas que contiene.

La única ventaja de este enfoque sería poder alojar en él varios medicamentos distintos para su toma.

3.4.2.1.2 Enfoque de dispensador

El enfoque de dispensador persigue la construcción de un prototipo formado por *dispensadores*. Cada uno de estos *dispensadores* contendría un tipo de medicamento y podría dispensar tantas pastillas de un mismo tipo como se le solicitara.

El microcontrolador sería el encargado de dispensar estos medicamentos mediante algún componente mecánico conectado a él y también se encargaría de controlar que la medicación fue retirada, notificaciones sonoras y luminosas y otros muchos aspectos objeto de este proyecto.



Figura 3.14. Enfoque de dispensador

Este enfoque permitiría crear una toma de medicación *a la carta* sin precondiciones de tener posicionados sus medicamentos de alguna forma lógica. Cada uno de los *dispensadores* tendría un único tipo de medicación y el prototipo tendría tantos *dispensadores* como medicamentos distintos necesite tomar el usuario.

Con este enfoque cobra mucho más sentido la idea de poder programar horarios de toma de medicamentos desde una aplicación web que no requiera estar presente junto al prototipo de dispensador para su programación.

Como único inconveniente, mencionar la dificultad de crear un dispensador que se adapte al inmenso número de pastillas con dimensiones distintas que existen en el mercado. La dificultad recaería en crear algún tipo de adaptación para cada tipo de pastilla que permita al dispensador controlar la liberación de medicamentos uno a uno.

Otra posible solución sería la construcción de un diseño generalizado que permita posicionar cada una de las pastillas de un tipo de medicamento en una posición dentro del dispensador. De esta manera, se dispensaría la pastilla de esa única posición permitiendo alojar en su interior cualquier tipo de medicamento, además de garantizar la liberación de pastillas una a una.

3.4.2.2 Alternativas de Microcontrolador

El microcontrolador del dispensador será el encargado de orquestar todos los componentes electrónicos distribuidos por el prototipo y conectados a él. Además de ser el encargado de las comunicaciones a través de Internet con el servidor.

En esta sección se evaluarán distintos microcontroladores candidatos a ser usados en este proyecto. Todos ellos disponen del módulo ESP8266 que permite su conexión a una red wifi.

3.4.2.2.1 WeMos D1 ESP8266 WIFI

El microcontrolador D1 es una placa que usa la arquitectura de Arduino¹⁰. Dispone de un módulo ESP8266 con el que conectarse a una red wifi y disponer de internet. Además, este tipo de placas pueden programarse desde el propio IDE de Arduino requiriendo para ello una librería adicional.



Figura 3.15. Microcontrolador WeMos D1 R2

¹⁰ Arduino es una compañía open source y open hardware, así como un proyecto y comunidad internacional que diseña y manufactura placas de desarrollo de hardware para construir dispositivos digitales y dispositivos interactivos que puedan controlar objetos del mundo real.

Uno de los aspectos más interesantes de este tipo de placas es su facilidad de programación. No resulta ser mucho más compleja que la programación de una placa Arduino convencional, añadiendo la posibilidad de disponer de una conexión a Internet con las enormes posibilidades que esto supone.

El principal inconveniente de la placa WeMos D1 para su uso en este proyecto es el escaso número de pines digitales de los que dispone. Presenta 11 entradas que no resultarían suficientes para controlar todos los componentes electrónicos que debería conectarse para su uso en el dispensador.

3.4.2.2 Arduino MEGA + ESP8266 WIFI

El microcontrolador Arduino MEGA + ESP8266 es una placa híbrida desarrollada por RobotDyn¹¹ que integra el controlador MEGA 2560 y el módulo ESP8266 en una única placa. Esto permite unificar en una placa lo mejor de los dos mundos; la capacidad y potencia de controlador MEGA y la disponibilidad de conexión a Internet del módulo ESP8266.

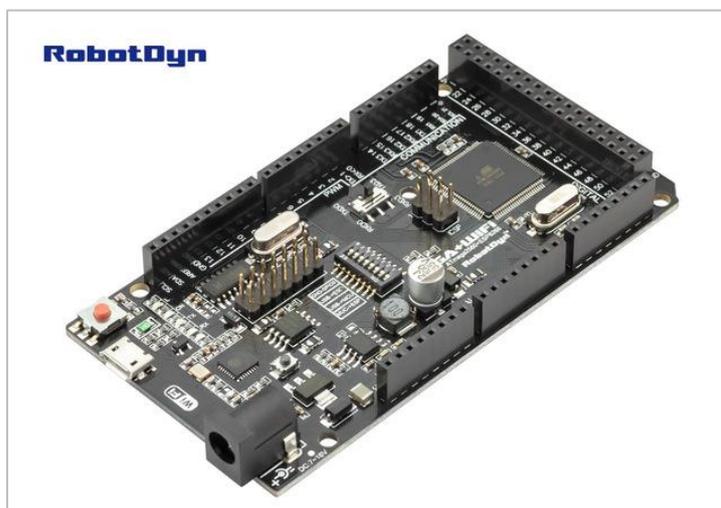


Figura 3.16. Microcontrolador MEGA + ESP8266

La principal ventaja de esta placa es su versatilidad. Dispone de 53 pines digitales que lo hacen perfecto para su uso en este proyecto, debido al gran número de componentes electrónicos que se requerirán conectarse a él y consumirán estas entradas digitales. Esto sumado al módulo ESP8266 que permite conectarse a una red wifi a través de la que disponer de una conexión a internet, lo hacen un candidato perfecto.

Como inconveniente destaca la dificultad de su programación. Para programar este tipo de placas es necesario crear dos tipos de programas diferentes, uno para el módulo controlador MEGA 2560 y otro para el módulo wifi ESP8266 y realizar las comunicaciones a

¹¹ RobotDyn fue fundada en 2014 en el sur de China en Zhuhai por el grupo internacional de desarrolladores de electrónica y expertos en producción. Su actividad principal es el desarrollo y producción de controladores y dispositivos para diversas plataformas, incluida la popular plataforma Arduino.

través del puerto serial entre ambos controladores. Aun así, ambas programaciones pueden realizarse desde el propio IDE de Arduino.

Otro inconveniente es que se trata una placa de reciente desarrollo, lo cual, sumado a la dificultad de su programación, complica considerablemente la labor de encontrar documentación al respecto.

3.4.2.3 Alternativas de Componentes Electrónicos

Para la construcción del prototipo de este proyecto se requerirán un gran número de componentes electrónicos distribuidos por el mismo y que dotarán de la funcionalidad necesaria al microcontrolador y al propio prototipo.

En este apartado se valorarán aquellos componentes electrónicos cruciales para el desarrollo del prototipo y del propio proyecto.

3.4.2.3.1 Motores para el dispensador

En el caso de optar por el diseño y construcción del prototipo siguiendo el enfoque de dispensador, será necesario instalar algún tipo de componente electromecánico que controle el microcontrolador y permita expulsar las pastillas de los dispensadores.

3.4.2.3.1.1 Servomotores

Los servomotores de rotación continua son una variante de los servos convencionales que permiten girar 360 grados y controlar la velocidad de giro. Además, disponen de un par de fuerza bastante alto que permite seguir girando sin pararse ante una resistencia.



Figura 3.17. Servomotor de rotación continua

El principal problema de este tipo de servo, enfocado a su uso en este proyecto, es no poder garantizar los grados de giro. Aun configurando la velocidad mínima que permiten estos servos, esta es demasiado alta como para garantizar su movimiento un número de grados determinado, lo cual representa un requisito indispensable para su uso en el dispensador.

Otro inconveniente son sus grandes dimensiones que harían el diseño del prototipo más grande de lo establecido en los requisitos iniciales del proyecto

3.4.2.3.1.2 Motores paso a paso

Los motores paso a paso son dispositivos electromecánicos que permiten girar 360 grados, controlar la velocidad de giro y garantizan una rotación determinada. Los desplazamientos angulares discretos son el punto fuerte de este tipo de motores.

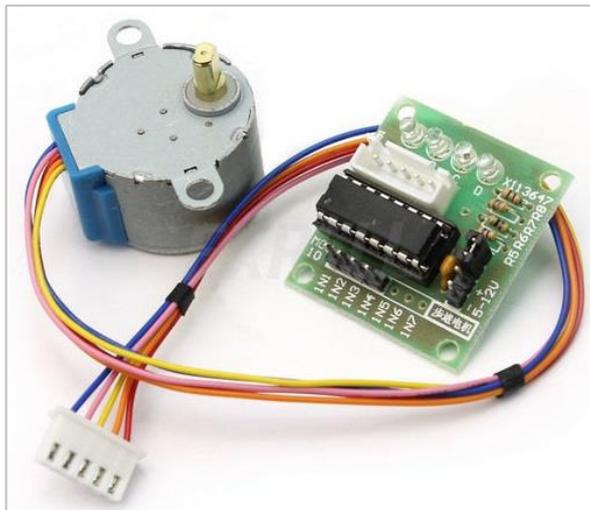


Figura 3.18. Motor paso a paso 28BYJ-48

Existen distintos tipos de motores paso a paso en el mercado. El que mejor se ajusta a las necesidades de este proyecto es el 28BYJ-48 debido principalmente a sus reducidas dimensiones y al tratarse del mejor motor paso a paso en relación par de fuerza – consumo.

Este motor basa su funcionamiento en un driver de control interpuesto entre el motor y el microcontrolador. Además, dispone de una resolución de hasta 512 pasos por vuelta de giro.

Como único contra, cada unidad requiere de cuatro pines digitales del microcontrolador para su funcionamiento, además de dos pines correspondientes a la alimentación.

3.4.2.3.2 Detección de toma de medicación

Garantizar la detección o retira de las pastillas o medicamentos de su contenedor en el prototipo es uno de los requisitos fundamentales de este proyecto y una de las funcionalidades de mayor aporte al mismo.

En este apartado se valorarán dos tipos de sensores que pueden permitir abordar esta funcionalidad.

3.4.2.3.2.1 Báscula de carga

Una báscula de carga o sensor de peso permite al microcontrolador calcular el peso del elemento posicionado sobre el mismo. Este sensor podría usarse para detectar si dispone de pastillas sobre él o, por el contrario, los medicamentos han sido retirados.



Figura 3.19. Balanza de carga

El principal inconveniente de este sensor es su resolución. El sensor con menor cálculo de peso es de 1 kg, obteniendo una resolución de 5 g, la cual resulta insuficiente para su uso en el dispensador debido a que una pastilla no suele pesar más de 2g.

Además, las dimensiones de este sensor son relativamente grandes y requiere de un elevado número de pines digitales para su funcionamiento.

3.4.2.3.2 Módulo infrarrojos IR

El sensor de infrarrojos IR permite detectar objetos que se interpongan en el mismo. El uso que se daría a este componente sería el de detectar si el usuario está retirando las pastillas del dispensador y de esa manera poder garantizar que, al menos, el usuario ha mostrado la intención de tomar la medicación.

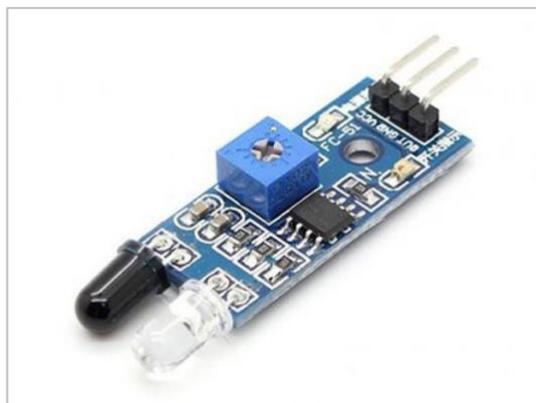


Figura 3.20. Sensor infrarrojos IR

Se trata de un sensor pequeño, con ajuste de resolución y con un alto rendimiento frente al bajo consumo de recursos que requiere para su funcionamiento. El único inconveniente es que se trata de un sensor muy sensible a la luz. Esto puede confundir la lectura de detección si el ajuste de resolución no está configurado correctamente. Configurando el ajuste de resolución a las condiciones del lugar en el que se encuentre el dispensador no debería haber errores en la lectura del sensor.

3.4.2.3.3 Integración de componentes

Con el objetivo de unificar las conexiones entre el microcontrolador y todos los componentes electrónicos y sensores conectados al mismo, se busca una solución en forma de placa de expansión del microcontrolador que garantice que los cables de conexión permanezcan conectados en su posición.

3.4.2.3.3.1 Creación de placa adaptada al microcontrolador y componentes electrónicos

Las placas de expansión PCB adaptadas al microcontrolador permiten agrupar en una única placa todas las conexiones entre los componentes electrónicos y el controlador. Para ello, es necesario soldar todo el cableado de los componentes electrónicos a dicha placa.

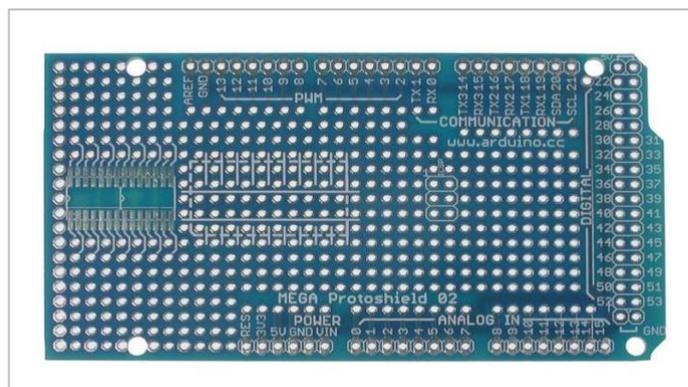


Figura 3.21. Placa de expansión PCB

Construir una placa PCB con todas las conexiones de los componentes es una de las soluciones más limpias y compactas a la hora de crear un proyecto de IoT. El principal inconveniente es su propia construcción. Se necesita de muchos otros componentes, tales como conectores y cables especiales, además de una gran habilidad estañando para no cortocircuitar otras conexiones.

3.4.2.3.3.2 Placa de expansión

Las placas de expansión ya construidas son usadas en proyectos de IoT una vez fijada y cerrada la distribución y elección de todos los componentes electrónicos y sensores que se usaran en el mismo.

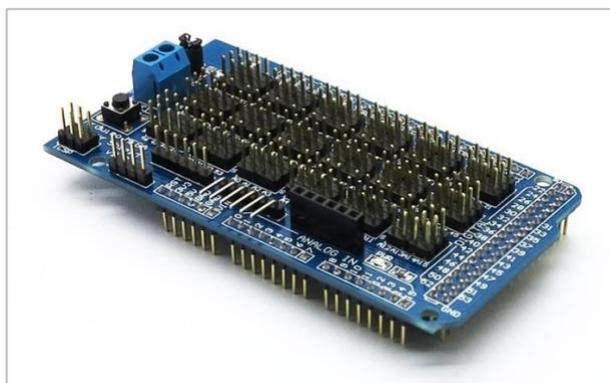


Figura 3.22. Placa de expansión MEGA

Este tipo de placas se conectan directamente al microcontrolador y permiten conectar a él de forma sencilla y segura todos los componentes electrónicos y sensores requeridos en proyecto.

Para su uso no necesita soldar ningún componente ya que todo lo necesario para su funcionamiento lo integra la propia placa.

El único inconveniente de este tipo de placas, tanto de las placas de expansión como de las PCB, es que ocultan el acceso al hardware del microcontrolador. El microcontrolador MEGA + ESP8266 visto anteriormente, dispone de un pequeño switch que es necesario configurar para programar independientemente el controlador MEGA y el módulo wifi ESP8266. El uso de estas placas dificultaría la programación de este microcontrolador, siendo necesario acoplar y desacoplar la placa de expansión en cada programación del microcontrolador.

3.4.3 Aplicación Web - Frontend

Para el diseño, desarrollo e implementación de aplicaciones web, existen innumerables tecnologías, librerías y frameworks que facilitan su creación en términos de acceso a datos, navegación, diseño y experiencia de usuario.

En esta sección se valoran algunas de las alternativas para la parte frontend de la aplicación web objeto del proyecto con el propósito es elegir aquellas librerías y/o frameworks que mejor se adapten a este proyecto y con las que obtener el mejor diseño y experiencia de usuario posible.

3.4.3.1 Alternativas de desarrollo de la Aplicación Web

Las alternativas frontend para el desarrollo de una aplicación web son enormes y cada una de ellas esta orientada a un objetivo específico, algunas adolecen por exceso de funcionalidades y ser demasiado pesadas y otras por escasez (teniendo como punto de referencia el desarrollo de la parte *front* de este proyecto). Se presentan a continuación las que mejor se adaptan a este proyecto y puedan facilitar y mejorar su desarrollo.

3.4.3.1.1 Frameworks y librerías Front-End

En este apartado se enumeran algunos de los frameworks¹² y librerías¹³ frontend más utilizados para el desarrollo de interfaces de usuario. Estos frameworks y librerías suelen basarse en el patrón MVC (*Model, View, Controller*), con el objetivo de separar las capas de presentación, lógica y componentes de una aplicación

¹² Un **framework** se diseña para apoyar el desarrollo de sitios web dinámicos, aplicaciones web y servicios web. Estos frameworks intenta aliviar el exceso de carga asociado con actividades comunes usadas en desarrollos web.

¹³ Una **librería** es un conjunto de implementaciones funcionales, codificadas en un lenguaje de programación, que ofrece una interfaz bien definida para la funcionalidad que se invoca.

3.4.3.1.1.1 AngularJS

AngularJS es un framework JavaScript de código abierto para el desarrollo frontend creado y mantenido por Google. A partir de la versión 2 fue completamente rediseñado. Actualmente se encuentra en la versión 5.



Figura 3.23. Logo Angular 5

Las principales ventajas de Angular son el *two-way-data-binding* que permite lograr que todo cambio visual se actualice a tiempo en el modelo y viceversa, el uso de directivas para conseguir un comportamiento específico de los elementos del DOM¹⁴ y la gran comunidad de desarrolladores que lo respalda.

A pesar de estas ventajas, el rendimiento de este frameworks se ve comprometido con el uso de *two-way-data-binding*, que requiere de reevaluación de variables en cada ciclo. El inyector de dependencias de Angular inicializa todos los módulos al cargar la aplicación, lo que implica un volumen de tráfico mayor del necesario. Y, por último, la curva de aprendizaje de este framework es demasiado pronunciada, siendo necesario mucho tiempo y amplios conocimientos de Angular para crear únicamente el esqueleto de un proyecto.

3.4.3.1.1.2 ReactJS

ReactJS es una librería JavaScript para el desarrollo de interfaces de usuario mantenida por Facebook. Esta librería dispone de un gran ecosistema de aplicaciones, herramientas y componentes que facilitan la creación de aplicaciones web y SPA (*Single Page Application*).



Figura 3.24. Logo React

Como ventajas, React es una librería con una curva de aprendizaje menos pronunciada que la de Angular, los componentes de React son sencillos de mantener y dispone de una capacidad de renderizado rápido que mejora la experiencia de usuario.

Aun así, React no se integra correctamente con librerías de manipulación del DOM como jQuery. A pesar de que la curva de aprendizaje es menor a la de [Angular](#), acostumbrarse

¹⁴ Document Object Model o DOM ('Modelo de Objetos del Documento' o 'Modelo en Objetos para la Representación de Documentos') es esencialmente una interfaz de plataforma que proporciona un conjunto estándar de objetos para representar documentos HTML, XHTML y XML.

al flujo de datos en una sola dirección puede requerir un tiempo. Además, la renderización de grandes cantidades de datos puede emplear más tiempo del deseado, lo que penalizaría el rendimiento de la aplicación.

3.4.3.1.1.3 *Vue.js*

Vue.js es otro framework de desarrollo de interfaces de usuario que se ha popularizado en los últimos años. El principal motivo de su popularización es que permite la construcción de aplicaciones progresivas, esto es, el desarrollo en Vue.js será tan complejo como lo sea la aplicación web que se desea crear. Esto permite usar únicamente aquellas funcionalidades de Vue.js necesarias para el desarrollo en un momento concreto e ir ampliándolas con las necesidades de la aplicación de forma incremental.



Figura 3.25. Logo Vue.js

Otras de las ventajas de usar Vue.js vienen dadas por el hecho de que el framework apenas ocupa 18kb de memoria con las mejoras de rendimiento que esto supone, es fácil de entender y de desarrollar, fácil de integrar, dispone de una documentación muy bien elaborada, es flexible y facilita la comunicación *two-way*.

A pesar de ser uno de los mejores frameworks del momento, presenta una serie de limitaciones, puesto que aún no dispone de una comunidad de desarrolladores tan grande como la de [Angular](#) o [React](#) y la mayoría de la comunidad se ubica en China, lo que implica una barrera de idioma en las comunicaciones.

3.4.3.1.1.4 *Moment.js*

Moment.js es una librería JavaScript que permite la manipulación de horas y fechas de una forma sencilla y sin requerir de otras dependencias para su uso. Permite *parsear*, validar y mostrar fechas según la localización y dando soportando la internacionalización.



Figura 3.26. Logo Moment.js

Dado que uno de los requisitos del proyecto es la programación de horarios de toma de medicamentos, resulta indispensable la utilización de una librería que facilite parsear las fechas y horas que deberán ser enviadas al microcontrolador del dispensador.

Esta librería permite ser utilizada tanto en el lado del cliente como en el lado del servidor [Node.js](https://nodejs.org/). Siendo un aspecto importante debido a que, de este modo, las fechas pueden ser manipuladas en el servidor y mostradas en el lado del cliente.

Además, Moment.js es una librería ligera, fácil de usar, con soporte a la internacionalización y localización con un número inmenso de funcionalidades que permiten parsear cualquier fecha y hora a cualquier tipo de formato.

3.4.3.1.1.5 jQuery

jQuery es la librería JavaScript multiplataforma por excelencia usada para simplificar la interacción con las páginas HTML, manipular el DOM de la página, manejar eventos, crear animaciones y el uso de la tecnología AJAX¹⁵ en las páginas web.



Figura 3.27. Logo jQuery

Se trata de una librería de código abierto, ligera, flexible y rápida con una excelente integración con AJAX, además de ser muy sencilla de usar. Su principal facultad es la de manipular los objetos del DOM de forma mucho más sencilla que JavaScript, el manejo de eventos y otras muchas funcionalidades que mejoran la experiencia de usuario en la aplicación web.

3.4.3.1.2 Bootstrap y CSS puro para la organización de estilos

Bootstrap es un framework desarrollado por Twitter, publicado bajo una licencia de código abierto para el diseño de sitios y aplicaciones web. Contiene todo tipo de plantillas de diseño de todos los elementos basados en HTML, CSS y tipografías, así como extensiones de JavaScript adicionales.

Fue creado para facilitar la adaptación web a todo tipo de dispositivos, por lo que todos sus componentes se adaptan al diseño responsive. Incluye un sistema de *Grid Layout* muy útil para maquetar las columnas de un sitio web. Además, es una herramienta de uso ágil y sencilla que se integra perfectamente con otras librerías y frameworks JavaScript y con gran cantidad de plugins desarrollados por terceros para extender su funcionalidad.

¹⁵ AJAX, acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications).



Figura 3.28. Logo Bootstrap

Las únicas desventajas vienen dadas por requerir familiarizarse con su estructura y nomenclatura debido a que esta no es trivial. Algunas adaptaciones de diseño y de componentes requiere de CSS externo a Bootstrap y, además, se trata de un framework pesado y para algunas de las funcionalidades requiere usar la librería [jQuery](#).

Otra alternativa es el uso de CSS puro para la organización de estilos de la aplicación web. Usar CSS como único recurso de diseño puede dificultar la implementación del diseño de la aplicación, debido a que controlar todas las características disponibles de CSS3 no es sencillo y puede requerir mucho tiempo.

Además, usando únicamente CSS para el diseño, sería necesario añadir demasiado código JavaScript para mejorar la experiencia de usuario e importar otras librerías que faciliten otras tareas de frontend, como por ejemplo el selector de fechas y horas de la programación de horarios.

La única ventaja que surge de utilizar CSS3 puro es la creación de un sitio o aplicación web totalmente personalizado, lo que incrementaría notablemente las horas de trabajo.

3.4.3.1.3 Librería D3 para la creación de infogramas

D3.js (*Data-Driven Documents*) es una librería JavaScript para crear infogramas dinámicos e interactivos en sitios y aplicaciones web a partir de datos. Para ello hace uso de SVG¹⁶, HTML y CSS3. Además, permite tener el control completo sobre el resultado visual final.



Figura 3.29. Logo D3

¹⁶ Gráficos vectoriales escalables o SVG es un formato de gráficos vectoriales bidimensionales, tanto estáticos como animados, en formato XML, cuya especificación es un estándar desarrollado por el W3C.

Dispone de una gran comunidad de desarrolladores que la respalda y desarrolla visualizaciones e infogramas de ejemplo que se publican en la web. Además, no requiere de ningún framework propietario para su uso como la gran mayoría de las alternativas disponibles.

También se trata de una librería muy versátil en la manipulación de datos del DOM, flexible, con llamativas escenas y niveles de interacción y animaciones y compatible tanto con navegadores modernos como antiguos.

La gran versatilidad y flexibilidad que ofrece esta librería impacta en la curva de aprendizaje, siendo ésta demasiado pronunciada. Esto es debido fundamentalmente a que requiere tener unos conocimientos mínimos de cómo funciona D3.js para empezar a trabajar con esta librería, incluyendo un amplio catálogo de funcionalidades entre las que se encuentran la selección de datos, formateo, creación del infograma, sus ejes, insertar los datos formateados en el infograma, actualizarlos, etc.

También es necesario destacar la dificultad de la funcionalidad *selections* que permite manipular elementos del DOM que contienen el SVG de la visualización. Esta funcionalidad es muy amplia, y en ocasiones difícil de entender.

Por último, esta librería sería utilizada para abordar las visualizaciones de los datos ambientales y anomalías, requisito del proyecto. Los datos serían recogidos del dispensador, enviados al servidor, almacenados en la base de datos y visualizados por medio de infogramas interactivos a través de la aplicación web.

3.6 Diagrama Preliminar de las Funciones del Microcontrolador

Las funciones principales que serán necesarias desarrollar en la implementación del microcontrolador del dispensador son las siguientes:

Función	Descripción
setup()	Será la función encargada de inicializar todas las variables y pines del dispensador.
controlMotores()	Función encargada del control de los motores del dispensador. Pararlos y arrancarlos.
eventosHardware()	Función encargada de procesar los eventos surgidos en los componentes electrónicos de entrada del microcontrolador.
rutinaApiHorarios()	Rutina de solicitud de horarios de toma de medicación. El microcontrolador solicita cada minuto al servidor los horarios programados.
comprobacionToma()	Función encargada del control de la toma de la medicación dispensada.
conexionWifi()	Función encargada de establecer conexión a Internet a través del router wifi especificado.
sendRequestApi()	Función que enviará las peticiones del microcontrolador a la API del servidor. Estas peticiones pueden ser GET o POST.
actuacionComponente()	Función encargada de realizar una actuación sobre alguno de los componentes electrónicos de salida del microcontrolador.

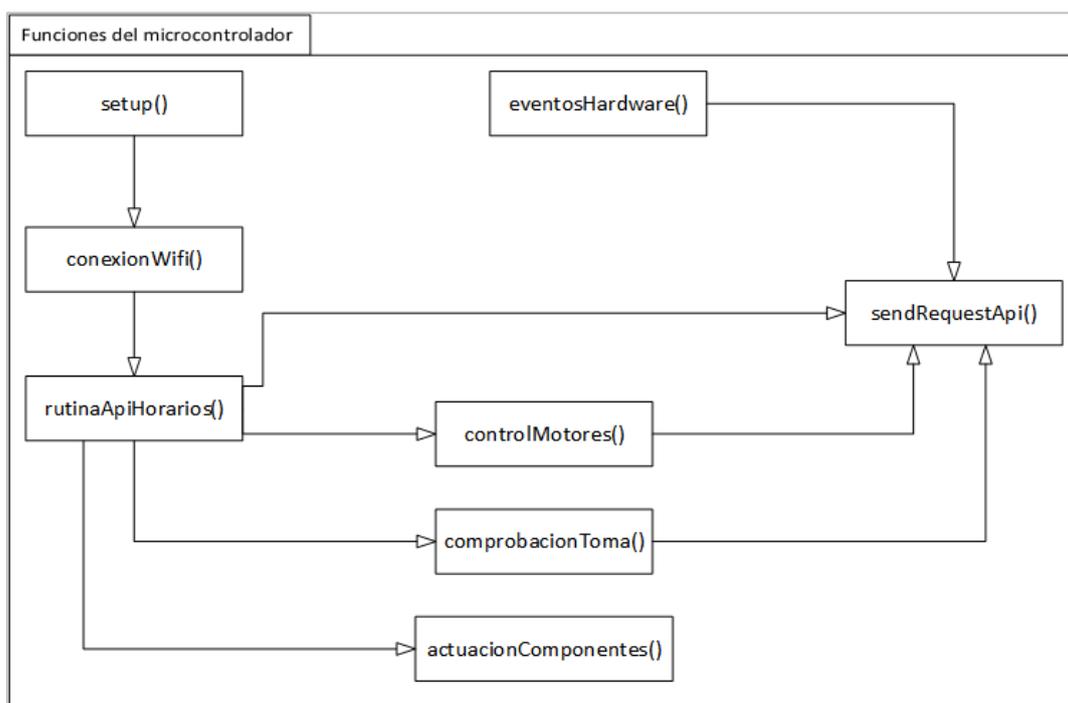


Figura 3.31. Interacción entre funciones preliminares del microcontrolador

3.7 Análisis de Interfaces de Usuario de la Aplicación web

En esta sección se describirá por medio de mockups los principales aspectos de la aplicación web de control, su comportamiento y navegabilidad. Por otra parte, la interfaz física del dispositivo dispensador de pastillas será descrita mediante bocetos.

3.7.1 Descripción de la Interfaz de la Aplicación web

A continuación se detallan los mockups que serán utilizados para el diseño de la aplicación. Cada uno de los bocetos corresponderá a una página de la aplicación. También se presenta la interfaz adaptada a dispositivos móviles.

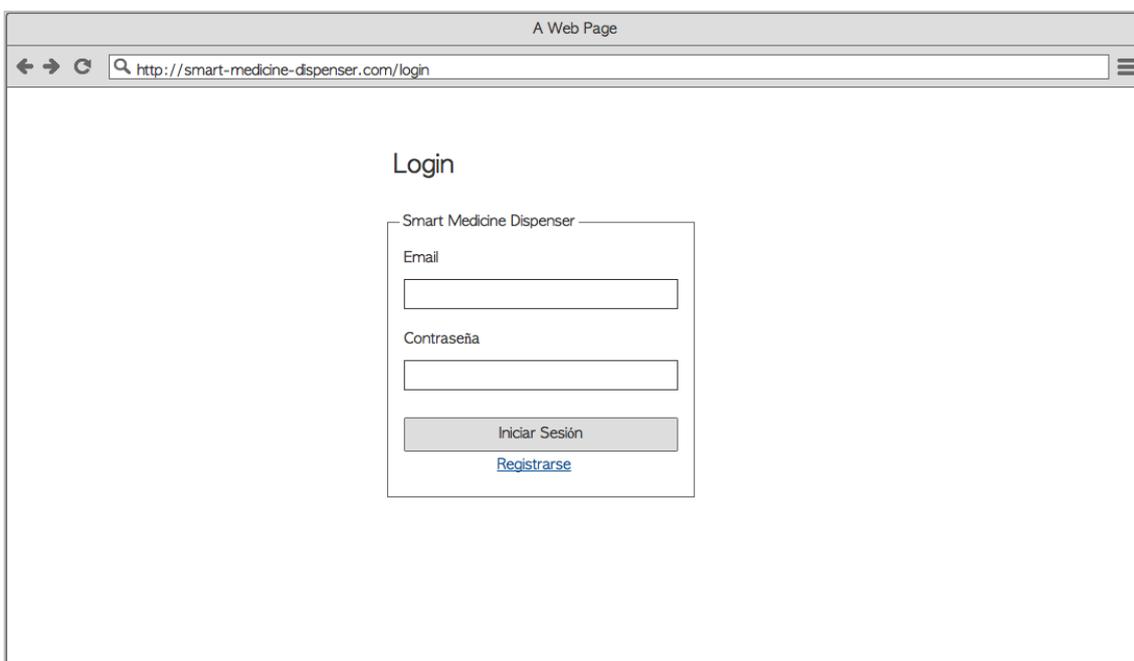


Figura 3.32. Mockup interfaz de la aplicación – Login

La ventana anterior muestra el aspecto que tendrá la página de inicio de sesión de la aplicación web. En ella, un usuario anónimo podrá identificarse como usuario registrado o como usuario administrador además de acceder a la página de registro.

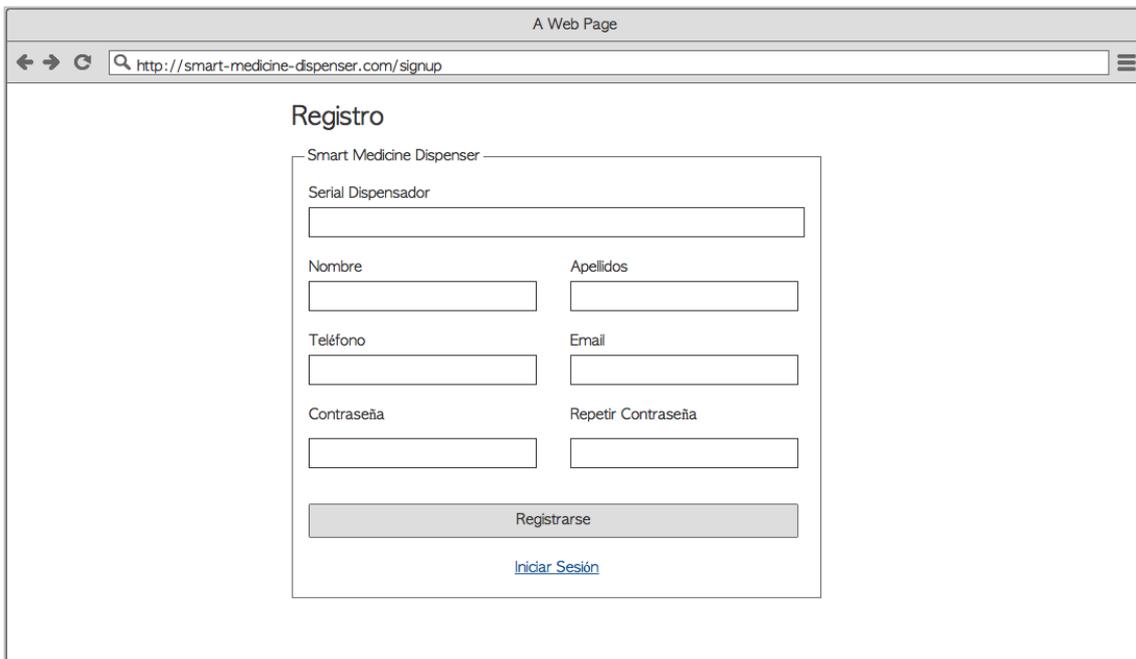


Figura 3.33. Mockup interfaz de la aplicación – Registro

El mockup anterior muestra la página de registro de la aplicación. Un usuario anónimo podría registrarse en el sistema introduciendo los datos del formulario. Desde esta ventana, el usuario puede regresar a la página de login a través del enlace *Iniciar Sesión*.

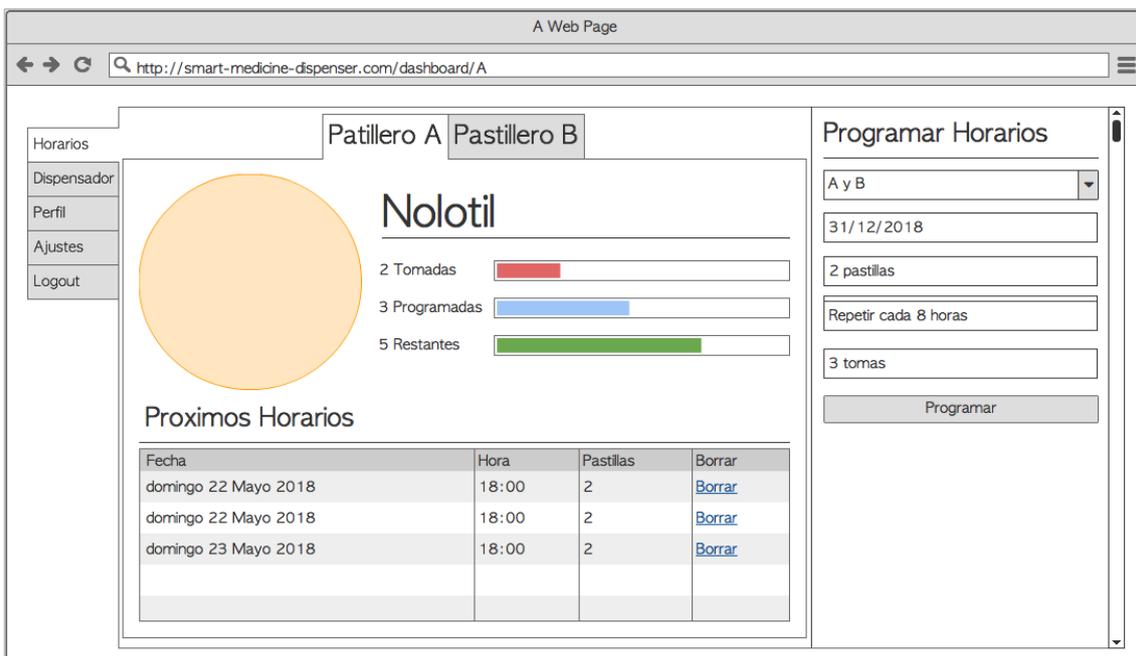


Figura 3.34. Mockup interfaz de la aplicación – Horarios

Esta sería la pantalla de inicio de un usuario cliente identificado en el sistema. En esta ventana se muestra el estado del dispensador, un listado con los próximos horarios programados para cada uno de los pastilleros y un formulario a través del que el usuario podrá programar nuevos horarios de toma de medicación.

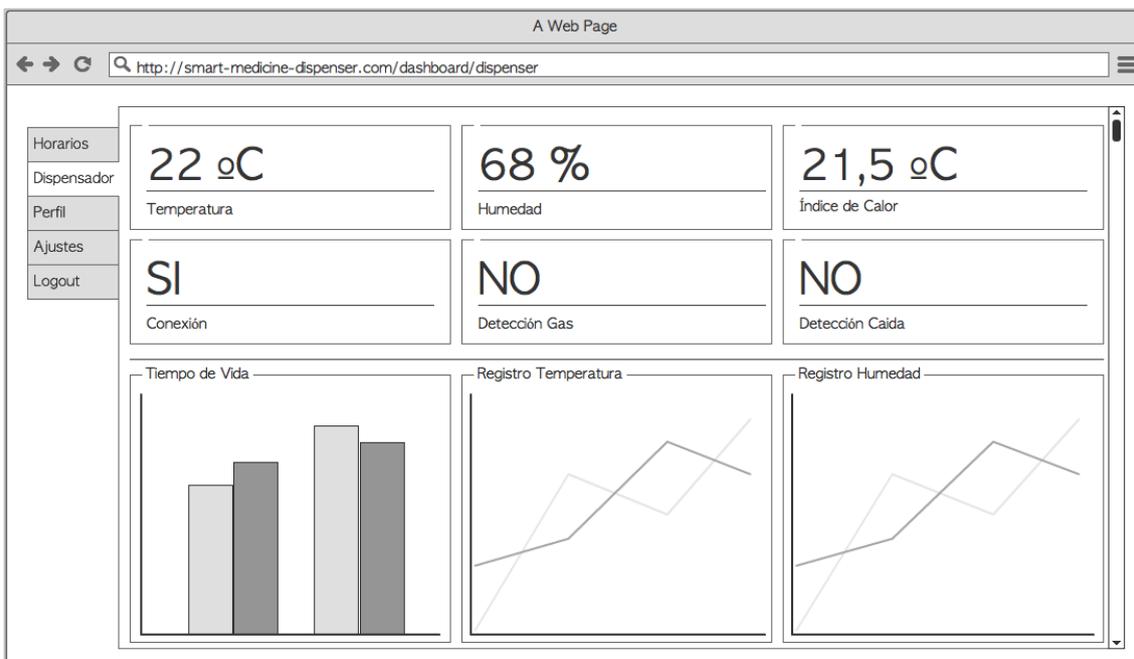


Figura 3.35. Mockup interfaz de la aplicación – Dispensador

Como usuario cliente identificado podrá acceder a la página del dispensador. En ella se muestra un resumen de datos en tiempo real del dispensador en forma de tarjetas y tres gráficos. Uno de ellos resume el número de conexiones del dispensador y los otros dos presentan la distribución horaria de la temperatura y humedad del entorno del dispensador.

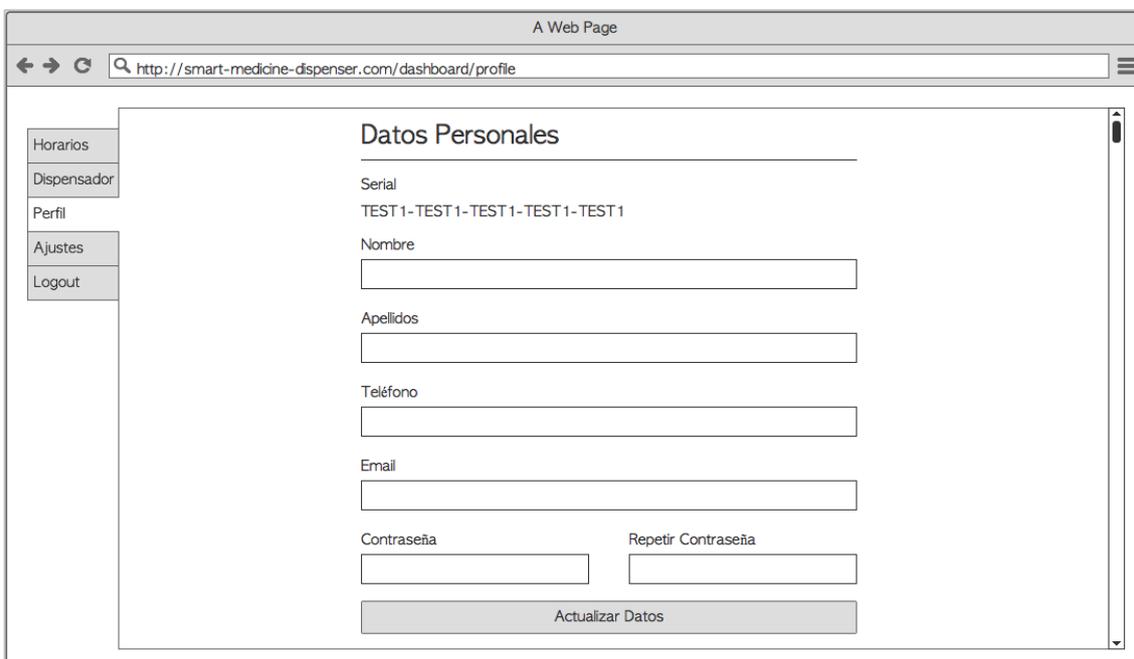


Figura 3.36. Mockup interfaz de la aplicación – Perfil

La página de perfil estará disponible tanto para el usuario cliente como para el administrador y en ella se podrán modificar los datos personales del usuario. Todos salvo el número de serie del dispensador utilizado para el registro.

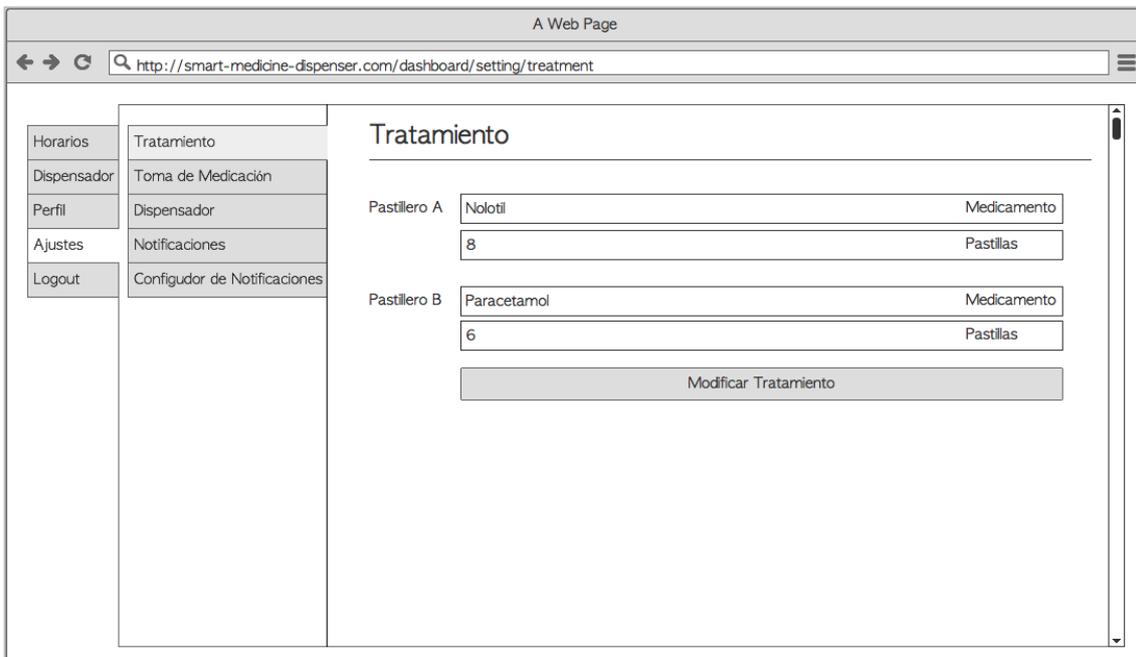


Figura 3.37. Mockup interfaz de la aplicación - Ajustes – Tratamiento

El mockup anterior muestra la opción de menú de *Tratamiento* dentro de la página de *Ajustes*. La opción de ajustes solo estará disponible para el usuario cliente identificado y en ella se dispondrá de un submenú que permitirá configurar las diferentes áreas del dispensador y del sistema. Cada elemento del submenú dispondrá de su propia página.



Figura 3.38. Mockup interfaz de la aplicación – Serials

La página de gestión de seriales solo estará disponible para el usuario administrador identificado en sesión. A través de esta página, el usuario podrá añadir nuevos seriales de dispensador, borrarlos y consultar sus detalles.

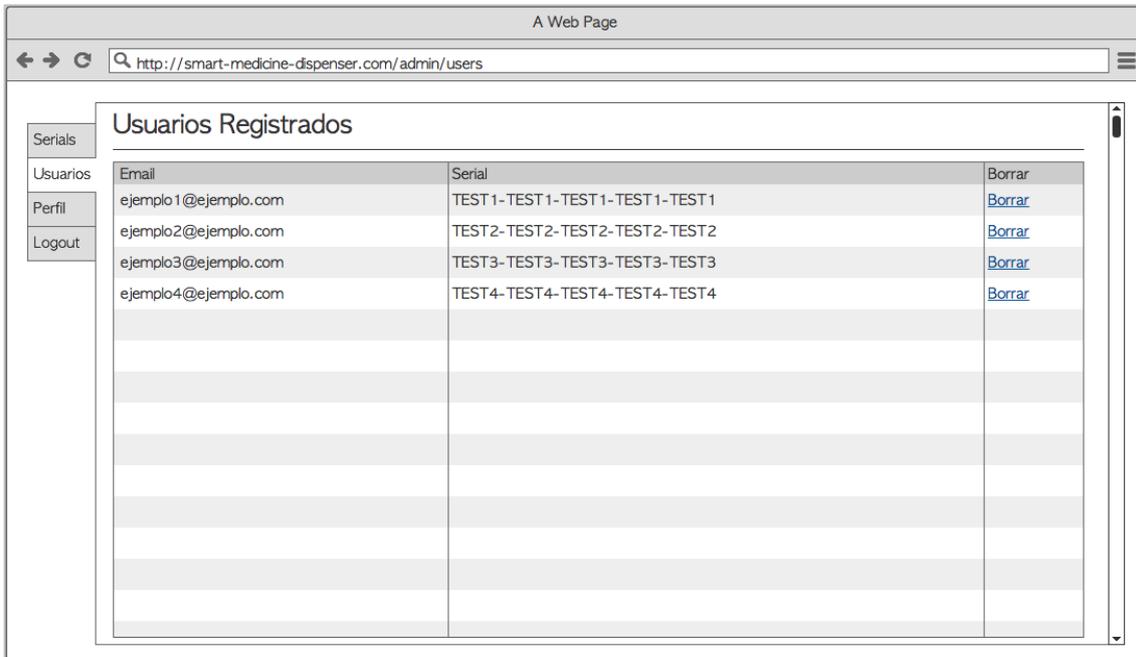


Figura 3.39. Mockup interfaz de la aplicación – Usuarios

Al igual que la página de seriales, la gestión de usuarios solo estará disponible para el usuario administrador iniciado en sesión. En ella, podrá consultar los usuarios registrados y borrarlos.

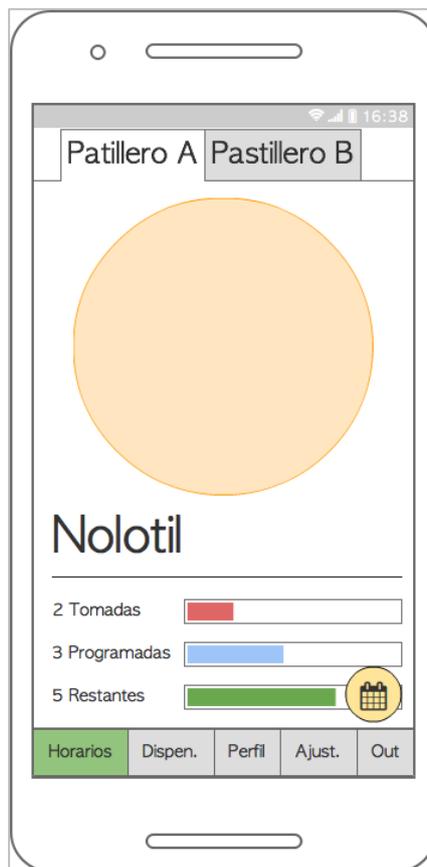


Figura 3.40. Mockup interfaz móvil de la aplicación – Horarios

Dado que uno de los requisitos del sistema es la creación de una aplicación responsive capaz de adaptarse a cualquier pantalla de cualquier dispositivo, se considera necesario crear un mockup específico para la versión móvil de la aplicación.

El mockup anterior muestra la misma pantalla de [horarios](#) pero renderizada en un dispositivo móvil. Puede observarse que la barra de menú será desplazada a la parte inferior de la pantalla y el formulario de programación de horarios ocultado. El formulario de programación pasará a estar accesible por medio de un botón flotante situado en la parte inferior de la pantalla. La distribución del estado de los pastilleros también será reorganizada. Esta página dispondrá de scroll y es por ello por lo que los horarios programados estarán situados debajo del estado de los pastilleros.

En esta sección solo se muestra el mockup de la interfaz móvil de la página de horarios debido a que se trata de la pantalla más compleja de la aplicación. El resto de páginas de la aplicación se adaptarán de igual forma que lo hace este mockup.

3.7.2 Descripción del Comportamiento de la Interfaz de la aplicación

Dado que tanto el panel de acceso del usuario cliente como el de administración será un dashboard de programación, análisis, consulta y ajustes del dispensador y del propio sistema, la barra de menú de la aplicación en la versión de escritorio será mostrada en formato vertical. En la versión móvil, esta barra se reposicionará en la parte inferior de la pantalla, lugar donde se suelen posicionar las barras de menú en las aplicaciones móviles.

La inserción, modificación y eliminación de información en el sistema por parte del usuario serán opciones comunes en la aplicación. Siempre que el usuario realice una acción que requiera modificar datos de información en el sistema, se le informará por medio de notificaciones al inicio de cada página con el resultado de la ejecución, bien sea por una acción llevada a cabo con éxito o sin él. En caso de error, se informará al usuario del motivo por el que no se ha llevado a cabo correctamente la acción solicitada.

En el formulario de programación de horarios descrito en el mockup de [Horarios](#) puede resultar complejo al inicio, sobre todo en la programación de múltiples horarios. Se incluirá un ejemplo de programación y su resultado en dicha sección con el objetivo de familiarizar al usuario con su uso.

El círculo mostrado en los mockups de la página de [Horarios](#) representará en formato visual el estado de cada uno de los pastilleros del dispensador. En él se visualizará la posición actual del pastillero, las posiciones dispensadas, las programadas, las restantes y las no disponibles. Esto se entenderá mejor analizando la sección [Análisis del Interfaz físico de Dispensador de pastillas conectado](#), que explica mediante un boceto los dos pastilleros que contendrá el dispensador.

También se procurará utilizar en la medida de lo posible los mismos estilos, colores y organización de componentes en toda la aplicación web.

3.7.3 Diagrama de Navegabilidad

Las vistas de las que dispondrá la aplicación y su interfaz fueron descritas en la sección anterior. A continuación se detalla la navegabilidad entre todas las páginas de la aplicación y los eventos que implicarán el cambio entre páginas.

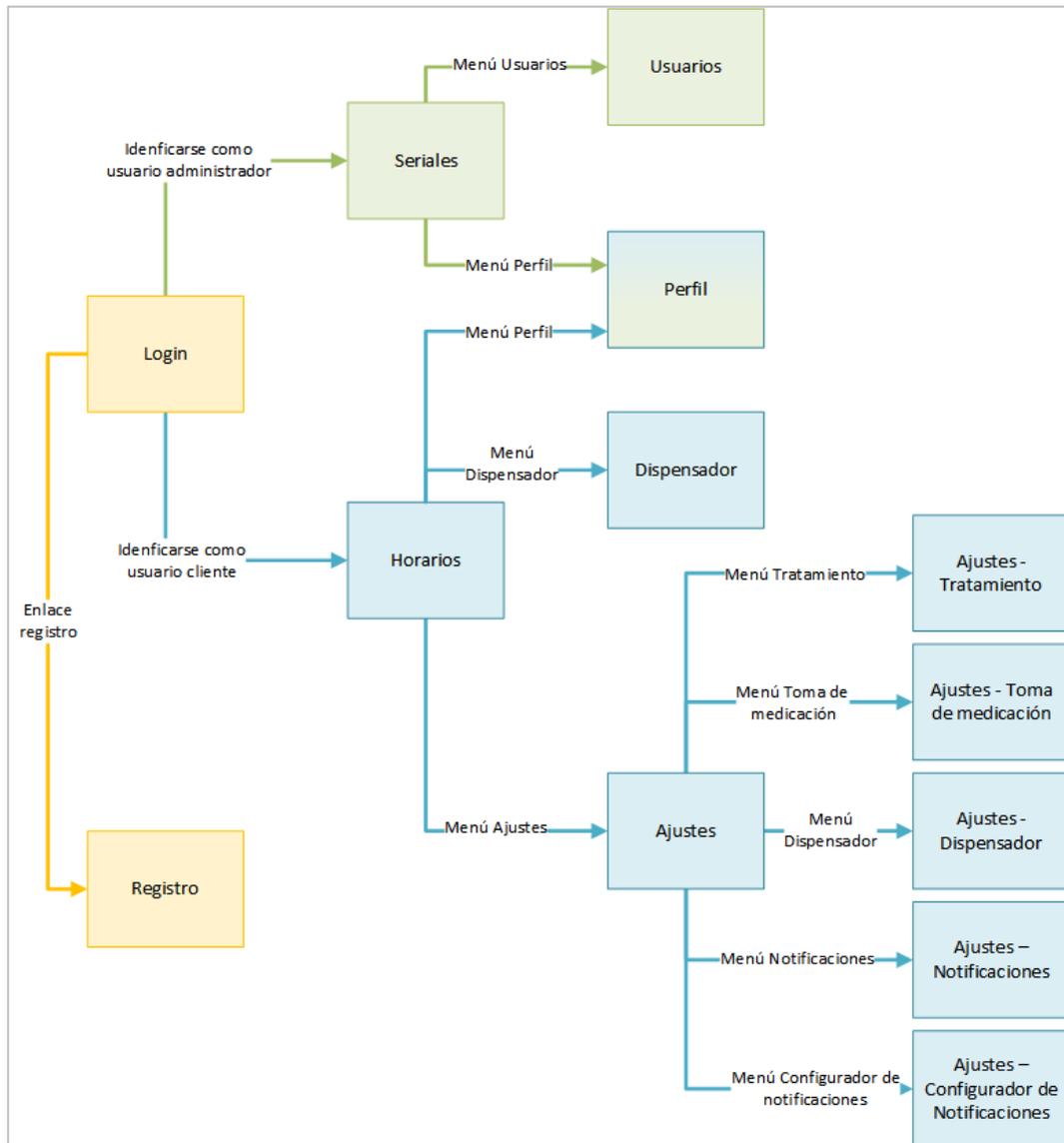


Figura 3.41. Diagrama de navegabilidad entre interfaces de la aplicación web

En el diagrama anterior se pueden apreciar tres colores diferentes en las páginas que componen la aplicación. Cada uno de ellos representa las páginas de navegabilidad de cada uno de los actores de la aplicación.

En amarillo se muestran las páginas a las que tendrá acceso el usuario anónimo o no identificado de la aplicación. La aplicación se inicia en la página de login. El usuario puede intercambiar entre la pantalla de inicio de sesión y la de registro por medio de enlaces en las mismas. Cuando un usuario anónimo se registra correctamente en la aplicación es redirigido a la pantalla de login.

Cuando un usuario anónimo se identifica en el sistema como usuario cliente del mismo, se le redirige a la pantalla de horarios. Las pantallas de color azul representan las páginas a las que tiene acceso un usuario cliente identificado en el sistema. Por medio de elementos de menú, el usuario puede acceder a la pantalla de dispensador, perfil, ajustes y por supuesto, a la de horarios. La navegabilidad entre estas cuatro páginas se puede realizar entre cualquiera de ellas a través de la barra de menú.

Dentro de la pantalla de ajustes, el usuario dispone de un submenú de configuraciones del sistema y del propio dispensador. Este submenú da acceso a cinco pantallas de ajustes: tratamiento, toma de medicación, dispensador, notificaciones y configurador de notificaciones. De nuevo, al tratarse de un menú, la navegación entre cualquiera de las páginas de la pantalla de ajustes se puede realizar entre todas ellas.

Por último, el color verde muestra las pantallas de acceso para el usuario administrador identificado en el sistema. La pantalla de inicio del usuario una vez identificado es la de seriales. Este rol de usuario tiene acceso a la pantalla de usuarios registrados en el sistema, perfil y gestión de seriales. Una vez más, cada una de las páginas será accesible por medio de un menú de navegación. Por este motivo se permite la navegación entre cualquiera de las tres pantallas.

3.8 Análisis del Interfaz físico del Dispensador de pastillas conectado

El dispensador de pastillas físico conectado será uno de los subsistemas principales de este proyecto. Será diseñado para ser impreso por completo en 3D y estará formado por varias piezas que, una vez impresas, se unirán para formar el dispensador completo. Cabe destacar que el diseño debe tener en cuenta todos los componentes electrónicos que deberán ser acoplados al mismo.

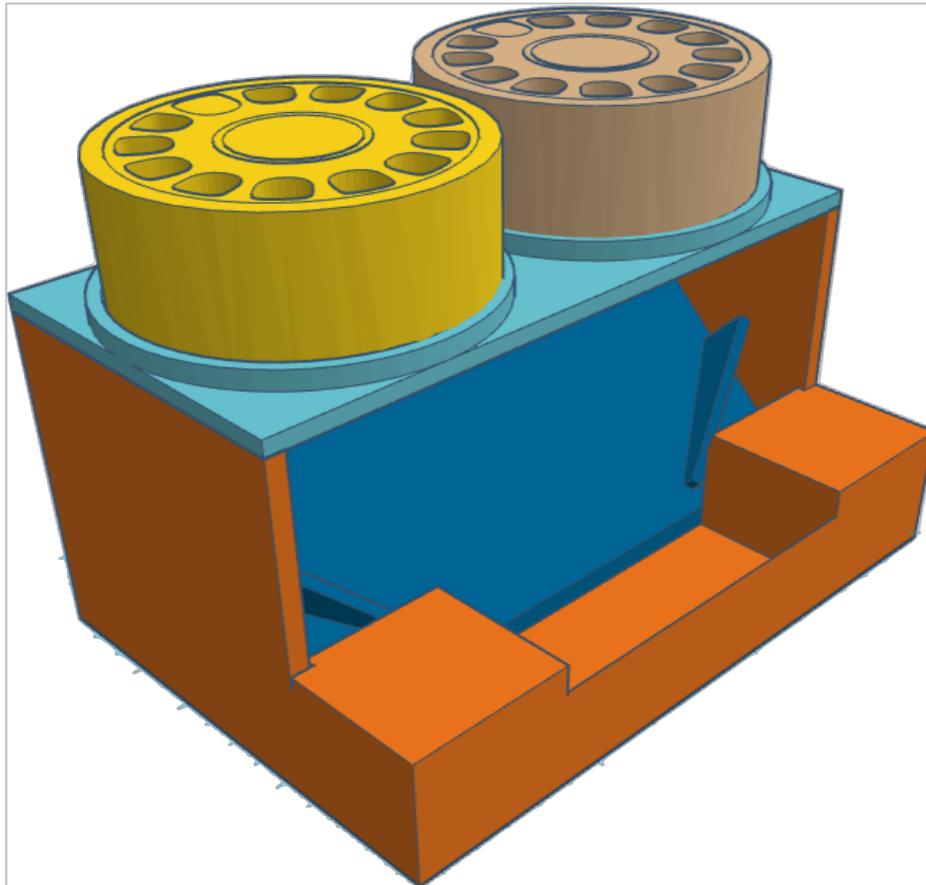


Figura 3.42. Boceto del dispensador de pastillas conectado

El boceto anterior muestra a grandes rasgos el aspecto que tendrá el dispensador. Se trata de un primer análisis del dispensador, por lo que no necesariamente dispondrá del aspecto mostrado en el boceto. Cabe recordar que uno de los requisitos no funcionales del sistema es que las dimensiones del dispensador no sobre pasen los 20 cm en la mayor de sus dimensiones.

El dispensador presentado en este primer boceto cuenta con dos ruletas o pastilleros donde se podrán alojar hasta 12 pastillas por cada uno de ellos. Cada pastillero contará con un único tipo de medicamento, pudiendo alojar hasta dos medicamentos diferentes.

Las pastillas serán dispensadas de una en una desde los pastilleros y caerán por la rampa de color azul hasta la cavidad situada en centro del dispensador de donde el usuario las recogerá. Los dos cubos situados a los lados del lugar de recogida de las pastillas han sido diseñados para alojar en su interior componentes y sensores interaccionables y que requieran de acciones por parte del usuario del dispensador.

En la parte trasera del dispensador y debajo de la rampa azul se alojarán el microcontrolador y todos los componentes electrónicos y sensores requeridos para completar la funcionalidad del dispensador.

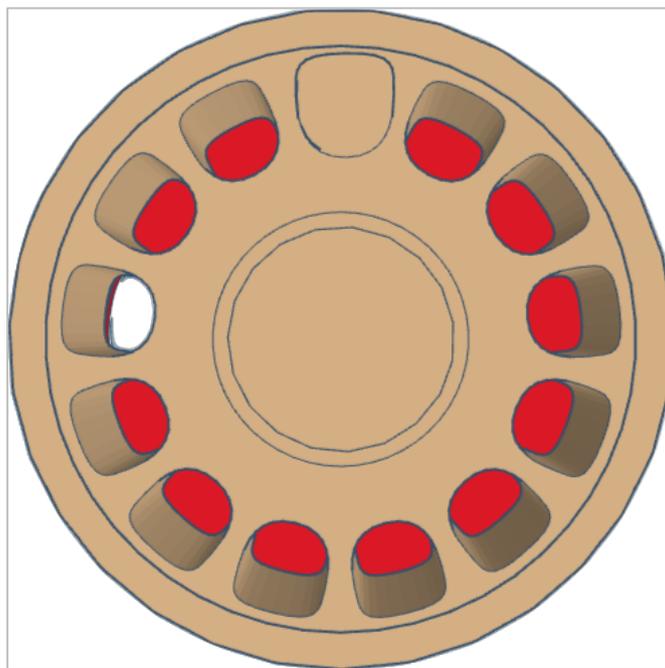


Figura 3.43. Boceto de uno de los pastilleros del dispensador

En el boceto anterior puede analizarse en detalle uno de los componentes más importantes del prototipo, la ruleta o pastillero que será el encargado de dispensar las pastillas contenidas en su interior de forma controlada.

Cuenta con 12 cavidades donde se insertará una pastilla por posición del pastillero. En el centro se alojará un componente electromecánico que permitirá controlar el selector rojo situado en la parte inferior de la ruleta (componente no soportado en el boceto). El selector contará con una única cavidad. El componente electromecánico hará girar el selector tantas veces como pastillas requieran ser dispensadas. Al girar, las pastillas del interior de la ruleta serán liberadas e irán cayendo una a una por la cavidad del selector hasta el lugar de recogida del dispensador.

3.9 Hipótesis y Restricciones

El presente proyecto trata de construir un sistema conectado compuesto por un prototipo de dispensador de medicamentos, una aplicación web de control y gestión y una base de datos junto a un completo sistema de notificaciones. A pesar de perseguir la construcción de un dispensador íntegro, compacto y sobre todo usable, debe tratarse como un primer prototipo con ciertas limitaciones.

Las hipótesis de partida y las restricciones y limitaciones que serán empleadas para el desarrollo y redacción del proyecto y que deberán tenerse en cuenta para la ejecución del mismo son las siguientes:

- **Número de medicamentos o tratamiento diferentes disponibles para ser dispensados**

El prototipo de dispensador únicamente permitirá dispensar dos tipos de medicamentos o tratamientos diferentes, uno por cada una de las *ruletas* disponibles en el mismo. Esto es debido a que se trata de un primer prototipo de dispensador que servirá como fuente referencia para futuros rediseños del mismo.

A pesar de dispensar únicamente dos tipos de medicamentos diferentes, desde el primer momento se ha pensado como un proyecto de desarrollo modular en el que, según las necesidades del usuario, se puedan incorporar nuevos pares de *ruletas* y de este modo hacerlo escalable para dispensar tantos medicamentos diferentes como sean necesarios.

Aunque en esta primera versión el prototipo de dispensador no contemple el sistema modular de incorporación de nuevas *ruletas*, el desarrollo del proyecto se realizará teniendo en cuenta este enfoque modular. De forma que un futuro rediseño del dispensador que soporte este enfoque, este pueda ser integrado al sistema completo (aplicación web de gestión, servidor y base de datos) de la forma mas sencilla posible.

- **Número máximo de pastillas disponibles para cada tratamiento**

El número máximo de pastillas que podrán ser insertadas en cada una de las *ruletas* (número máximo de pastillas por tratamiento) será de 12. Por lo que el dispensador podrá alojar un máximo de 24 pastillas, 12 por cada tipo de medicación o tratamiento.

A pesar de no ser un número excesivamente bajo, 12 pastillas tampoco se podría considerar un número alto en cuanto al número de pastillas se refiere. Para este primer prototipo de dispensador se consideran suficientes.

La situación ideal sería disponer de algún tipo de contenedor en el que depositar un gran número de pastillas (cada contenedor tendría su tipo medicamento) y el dispensador fuera capaz de dispensar una a una según la programación realizada por el usuario.

Este supuesto es realmente complejo debido a que la diversidad de formas y tamaños de medicamentos es muy alta, y por ello el dispensador debería ser adaptado a las dimensiones del medicamento del contenedor para que fuera capaz de dispensar una única pastilla cuando se solicite.

Además, el número de pastillas del que dispondría el contenedor tampoco podría ser muy elevado. Debido a que estas se encontrarían fuera de su embalaje y podrían degradarse con el tiempo en contacto con el aire.

- **Conexión wifi del microcontrolador**

Para que el microcontrolador del dispensador pueda conectarse a un router WiFi con el que tener acceso a internet, este debe conocer el SSID¹⁷ y la contraseña del router WiFi al que desea conectarse.

Con el objetivo de simplificar el desarrollo hardware de este proyecto, el SSID y la contraseña de router wifi al que se conectará el microcontrolador del dispensador será incluido en el código del propio microcontrolador.

Una posible solución a esta restricción sería incorporar un lector de tarjetas microSD¹⁸ al dispensador y conectarlo al microcontrolador. De esta manera, el código de desarrollo del microcontrolador soportaría la lectura de los datos contenidos en esta tarjeta. El usuario final modificaría los datos de la tarjeta para añadir el SSID y la contraseña de su router wifi. Al insertar de nuevo la tarjeta en el lector del microcontrolador, leería el SSID y la contraseña del router wifi al que debería conectarse e iniciaría este proceso de conexión.

Así, se eliminaría la restricción de tener hard-codeado¹⁹ el SSID y la contraseña del router wifi en el código del microcontrolador y se pasaría a consultar estos datos a través de un fichero de configuración contenido en la tarjeta SD.

¹⁷ El SSID (Service Set Identifier) es una secuencia de 0-32 octetos incluida en todos los paquetes de una red inalámbrica para identificarlos como parte de esa red. A menudo al SSID se le conoce como “nombre de la red”.

¹⁸ Secure Digital (SD) es un dispositivo en formato de tarjeta de memoria para dispositivos portátiles.

¹⁹ Hard-code, término del mundo de la informática hace referencia a una mala práctica en el desarrollo de software que consiste en incrustar datos directamente (a fuego) en el código fuente del programa.

3.10 Presupuesto para el Cliente

A continuación se detalla el presupuesto para cliente a partir de los principales hitos que serán desarrollados para la ejecución del proyecto. Para cada una de las tareas se asignan el número de horas necesarias de cada uno de los roles para cumplir con el hito.

Código Ítem	Concepto	Cantidad Horas	Precio Unitario	Total Concepto
01	Análisis y estudio de viabilidad del proyecto	160	-	5.825,00 €
01.1	Director del proyecto	15	50,00 €	750,00 €
01.2	Analista	145	35,00 €	5.075,00 €
02	Diseño y construcción del dispensador	104	-	2.740,00 €
02.1	Diseñador del dispensador	14	35,00 €	490,00 €
02.2	Constructor del dispensador	90	25,00 €	2.250,00 €
03	Desarrollo de la aplicación web	320	-	7.170,00 €
03.1	Analista	10	35,00 €	350,00 €
03.2	Programador	310	22,00 €	6.820,00 €
04	Desarrollo del microcontrolador del dispensador	204	-	4.488,00 €
04.1	Programador	204	22,00 €	4.488,00 €
05	Documentación del proyecto	160	-	3.858,00 €
05.1	Analista	26	35,00 €	910,00 €
05.2	Programador	134	22,00 €	2.948,00 €
06	Presentación y entrega al cliente	76	-	2.092,00 €
06.1	Director del proyecto	15	50,00 €	750,00 €
06.2	Programador	61	22,00 €	1.342,00 €
07	Hardware, licencias y otros conceptos	-	-	2.000,00 €
			SUBTOTAL	28.173,00 €
			21% IVA	5.916,33 €
			TOTAL	34.089,33 €

3.11 Especificación del Plan de Pruebas

Los objetivos principales de las pruebas que serán realizadas en este proyecto son; comprobar el funcionamiento del sistema conectado, su correcta integración entre todos los subsistemas que lo componen y garantizar que los usuarios de la aplicación web y del dispensador de pastillas comprenden su funcionamiento y se familiarizan con su entorno.

Para comprobar el funcionamiento del sistema y su correcta integración, se realizarán **pruebas unitarias** sobre cada uno de los requisitos que componen el catálogo de funcionalidades requeridas para este sistema. Se probará la funcionalidad de cada requisito, se detallarán los resultados que deben obtenerse y finalmente se especificará el resultado obtenido que deberá ser igual al que fue identificado como el resultado esperado. En el caso de que el resultado obtenido y el esperado sea diferente, se tratará como un error en el sistema que deberá ser corregido en la implementación del mismo.

Destacar que las pruebas unitarias de cada uno de los requisitos serán realizadas de forma manual y plasmadas en un informe de resultado comprobando tanto el funcionamiento del sistema como su correcta integración entre todos los subsistemas. Las pruebas sobre cada uno de los requisitos pueden involucrar a diferentes subsistemas para su ejecución, es por este motivo por el que se decide realizar las pruebas unitarias de forma manual en lugar de automatizarlas y limitar el resultado a probar un único subsistema. Las pruebas unitarias manuales también permiten realizar pruebas de integración del sistema a diferencia de las automáticas.

Los dos principales componentes del sistema con los que interactuará el usuario serán el dispensador de pastillas conectado y la aplicación web de control. El dispensador será el componente físico sobre el que actuará el usuario del dispensador y la aplicación web el elemento de control y gestión del dispensador.

Tanto para el dispensador como para la aplicación web se realizarán **pruebas de usabilidad** adaptadas a cada uno de estos dos subsistemas que permitirán garantizar que los usuarios comprenden su funcionamiento, se familiarizan con su uso y comprobar como interactúan con cada uno de ellos.

Las pruebas de usabilidad ayudarán a detectar fallos en la interfaz de la aplicación web o en el diseño del dispensador que transitan al usuario final cierta confusión en su funcionamiento. Con el resultado de estas pruebas, tanto la aplicación como el dispensador pueden ser adaptados en su implementación para suplir las carencias detectadas y solucionar ciertos aspectos de diseño y funcionamiento que no queden del todo claros.

Capítulo 4. Diseño del Sistema

4.1 Arquitectura del Sistema

La arquitectura del sistema conectado está formada por los tres principales subsistemas identificados en el capítulo anterior; la aplicación web de control, el dispensador de pastillas conectado y la base de datos. En este sistema desempeña un papel fundamental el servidor. Será el elemento físico que orquestará todos los subsistemas y será el encargado de recibir peticiones, procesarlas, ejecutar la lógica necesaria y elaborar la respuesta a la petición.

Para comprender la arquitectura del sistema, se detallarán los elementos y subsistemas de los que se compone el sistema conectado junto a las comunicaciones que se realizan entre todos ellos.

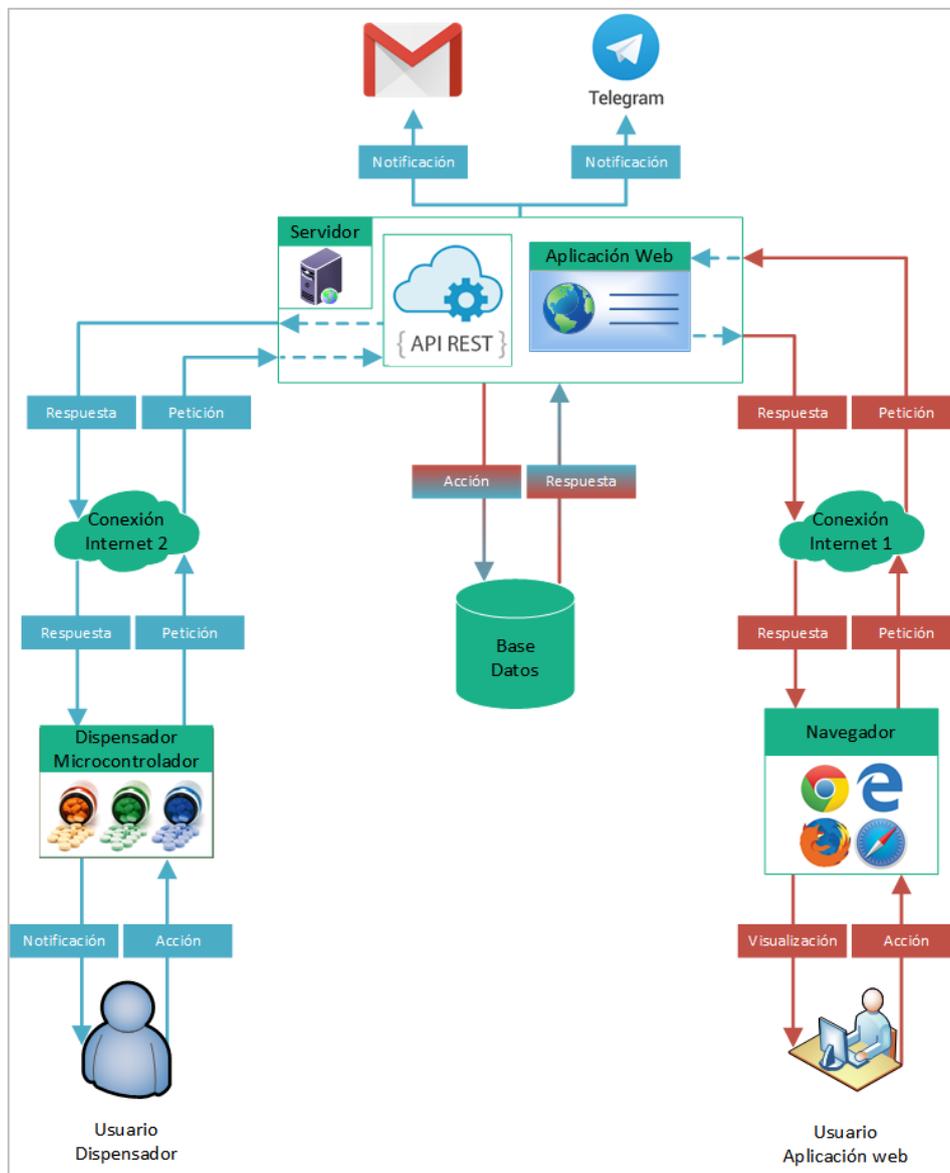


Figura 4.1. Arquitectura del sistema

El esquema anterior de la arquitectura del sistema muestra todos los componentes, elementos y subsistemas de los que se compone el proyecto. Para comprender realmente el funcionamiento y la arquitectura del sistema, se detallará el flujo de ejecución de las dos líneas de comunicación principales del sistema iniciadas por el usuario del dispensador y por el usuario de la aplicación web de control.

- **Línea de comunicación 1: Usuario de la Aplicación web – Navegador – Conexión Internet 1 – Servidor – Aplicación web – Base de datos**

El usuario de la aplicación ingresa en el navegador la dirección donde se encuentra desplegada la aplicación web. A través de Internet, se solicita al servidor la página asociada a esa dirección. El servidor realiza las acciones pertinentes (obteniendo información de la base de datos en el caso de que sea necesario) y elabora la respuesta que envía de nuevo al navegador del cliente en forma de visualización de la aplicación web.

El resto de las comunicaciones de esta línea y acciones realizadas por el usuario de la aplicación en la aplicación web seguirá el mismo proceso descrito anteriormente. Bien sea programar un horario, navegar por las páginas de la aplicación o configurar las notificaciones del dispensador entre otras acciones.

- **Línea de comunicación 2: Usuario del Dispensador – Dispensador/Microcontrolador – Conexión Internet 2 – Servidor – API REST – Base de datos y Gmail/Telegram**

Esta línea de comunicación es algo mas compleja que la anterior debido a que depende de quién realice la acción (el dispensador o el usuario del dispensador), el flujo de comunicación puede ser uno u otro:

- **Acción iniciada por el microcontrolador del dispensador**

Es el flujo de comunicación que se realiza cuando el dispensador solicita al servidor horarios de toma de medicamentos programados. El microcontrolador del dispensador solicita a la API REST del servidor a través de internet, la existencia de horarios de toma de medicamentos programados. La API del servidor recibe la petición, consulta en la base de datos los horarios programados y elabora la respuesta. Esta es enviada a través de internet y recibida por el microcontrolador. El microcontrolador evalúa la respuesta y en el caso de que sea necesario dispensar algún medicamento emite notificaciones sonoras y luminosas con el objetivo de que el usuario tome la medicación dispensada que ha sido programada para ese momento.

Con el objetivo de aprovechar la petición de horarios que realiza el dispensador a la API del servidor cada minuto, en esta misma petición el microcontrolador del dispensador envía parámetros ambientales y de estado del dispensador recabados de los distintos componentes electrónicos y sensores conectados a él. El servidor los procesa y almacena en la base de datos con el objetivo de ser mostrados a través de la aplicación web. Y en el caso de que alguno de los parámetros indique alguna anomalía en el

dispensador o en su entorno, el servidor enviará notificaciones a los usuarios del dispensador a sus correos electrónicos (servidor de correo Gmail) y al canal de Telegram (creado específicamente para el sistema).

- **Acción iniciada por el usuario del dispensador de pastillas**

El usuario realiza una acción sobre alguno de los componentes hardware conectados al microcontrolador del dispositivo, como puede ser pulsar un botón o retirar las pastillas dispensadas (esto es detectado por un sensor instalado en el dispensador con este objetivo). El microcontrolador envía dicha acción en forma de petición a la API del servidor a través de Internet. El servidor procesa la acción y toma las acciones oportunas (almacenar acción en la base de datos y/o enviar notificaciones a través de correos electrónicos y Telegram). Finalmente, responde al microcontrolador con el objetivo de confirmar que ha recibido la petición.

En este flujo de comunicación, el dispensador no emite ninguna notificación al usuario más allá de un simple *bip* de sonido. Este *bip* no requiere respuesta por parte del servidor para ser emitido, ya que su único objetivo es notificar al usuario que algún elemento hardware del dispensador ha emitido un evento por su acción.

En este punto cabe destacar el motivo por el que el dispositivo que dispone del navegador se encuentra conectado a un punto de acceso a Internet distinto al que se encuentra el microcontrolador del dispensador. El usuario de la aplicación únicamente hace uso de la aplicación web (programar horarios, monitorizar el dispensador, ajustar parámetros, etc.) y el usuario del dispensador solo hace uso del dispensador de medicamentos. El objetivo de esta situación descrita en el esquema anterior es aclarar que el dispensador podrá ser usado independientemente de quien gestione las acciones del dispensador a través de la aplicación web. Esto no quiere decir que el sistema completo no pueda ser usado por la misma persona, sino que la arquitectura por la que se rige permite usar independientemente el dispensador y la aplicación web.

Con la arquitectura del sistema y su flujo de ejecución descritos, podemos resumir el proyecto en la creación de un dispensador de medicamentos conectado a un servidor a través internet y gobernado por las acciones realizadas en la aplicación web de gestión.

Siendo un proyecto enmarcado en el ámbito del *Internet de la Cosas* cuyo objetivo es facilitar la actividad diaria de toma de medicamentos y poder garantizar que ninguna toma sea olvidada delegando el recordatorio de esta tarea al dispensador conectado.

4.2 Diseño de Casos de Uso

En el capítulo de análisis de sistema se realizó la [especificación de casos de uso](#) del sistema acorde al catálogo de requisitos. En esta sección se detallará el diseño de cada uno de los casos de uso especificados haciendo uso de diagramas de secuencia.

Al igual que la especificación, los casos de uso serán organizados según el actor que origina los mismos o sobre el que el sistema actúa en el caso de tratarse de actores secundarios.

4.2.1 Diagramas de secuencia de los casos de uso del actor Usuario Anónimo

4.2.1.1 Diagrama de secuencia del caso de uso "Registrarse en el sistema"

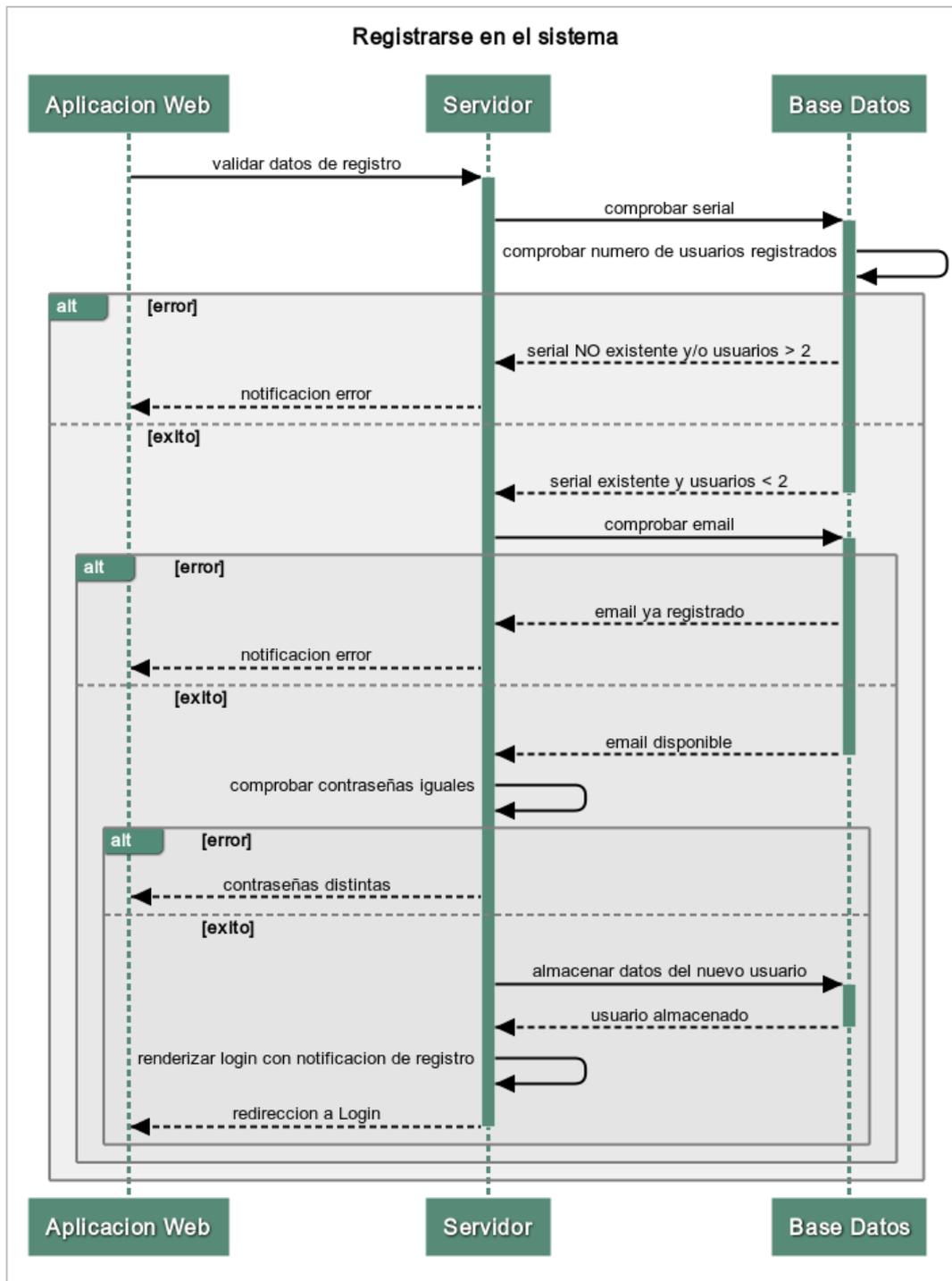


Figura 4.2. Diagrama de secuencia "Registrarse en el sistema"

4.2.1.2 Diagrama de secuencia del caso de uso "Identificarse como cliente"

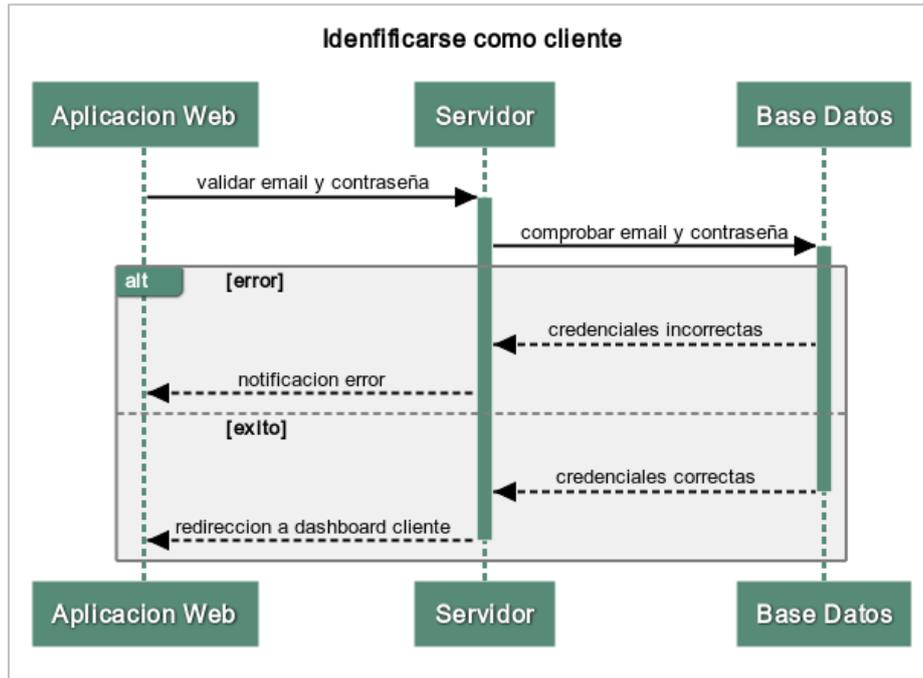


Figura 4.3. Diagrama de secuencia "Identificarse como cliente"

4.2.1.3 Diagrama de secuencia del caso de uso "Identificarse como administrador"

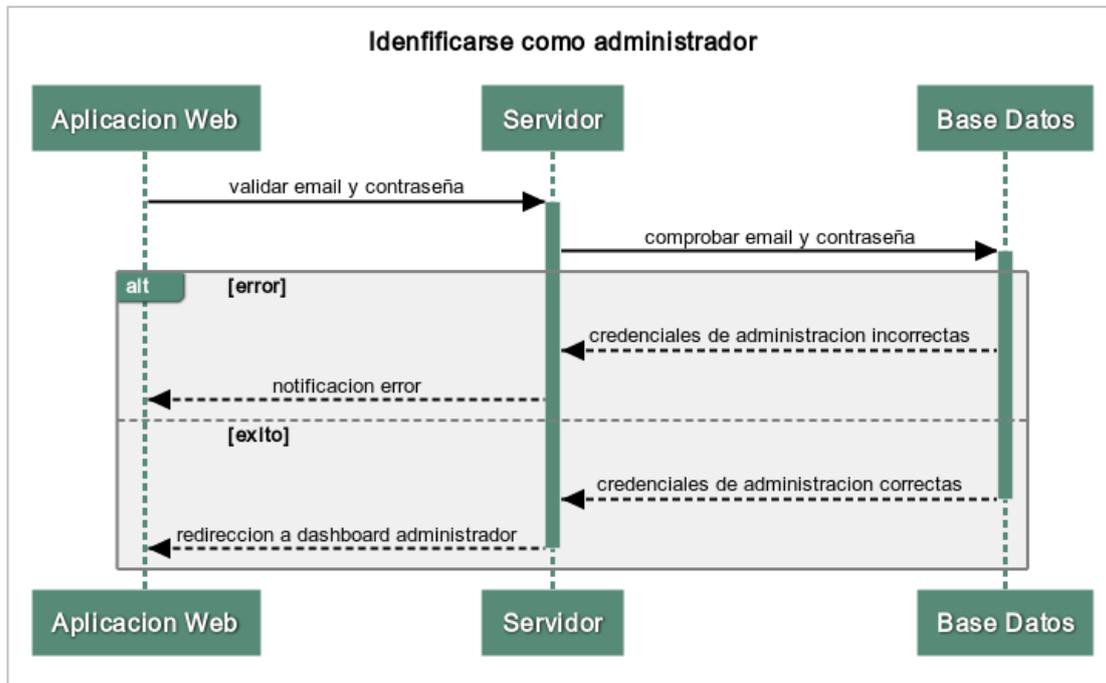


Figura 4.4. Diagrama de secuencia "Identificarse como administrador"

4.2.2 Diagramas de secuencia de los casos de uso del actor Usuario Cliente

4.2.2.1 Diagrama de secuencia del caso de uso "Consultar estado del dispensador"

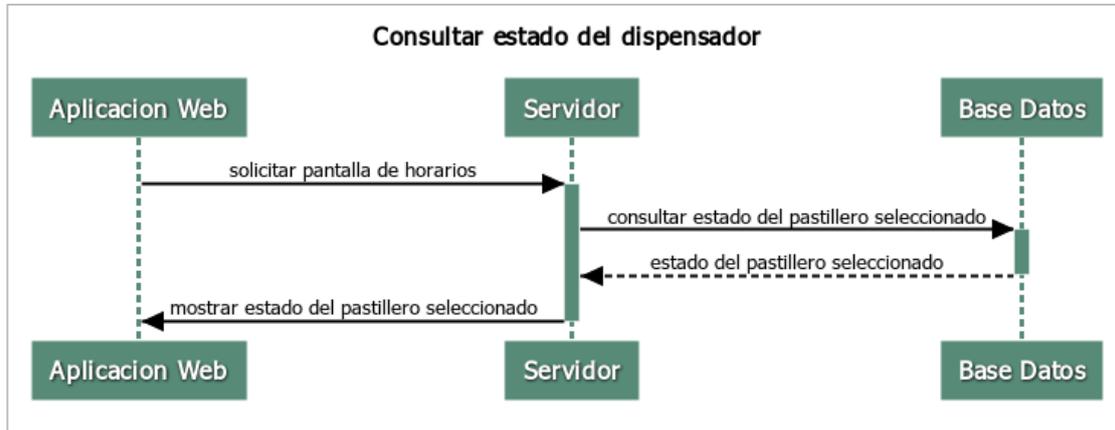


Figura 4.5. Diagrama de secuencia "Consultar estado del dispensador"

4.2.2.2 Diagrama de secuencia del caso de uso "Programar horarios de toma de medicación"

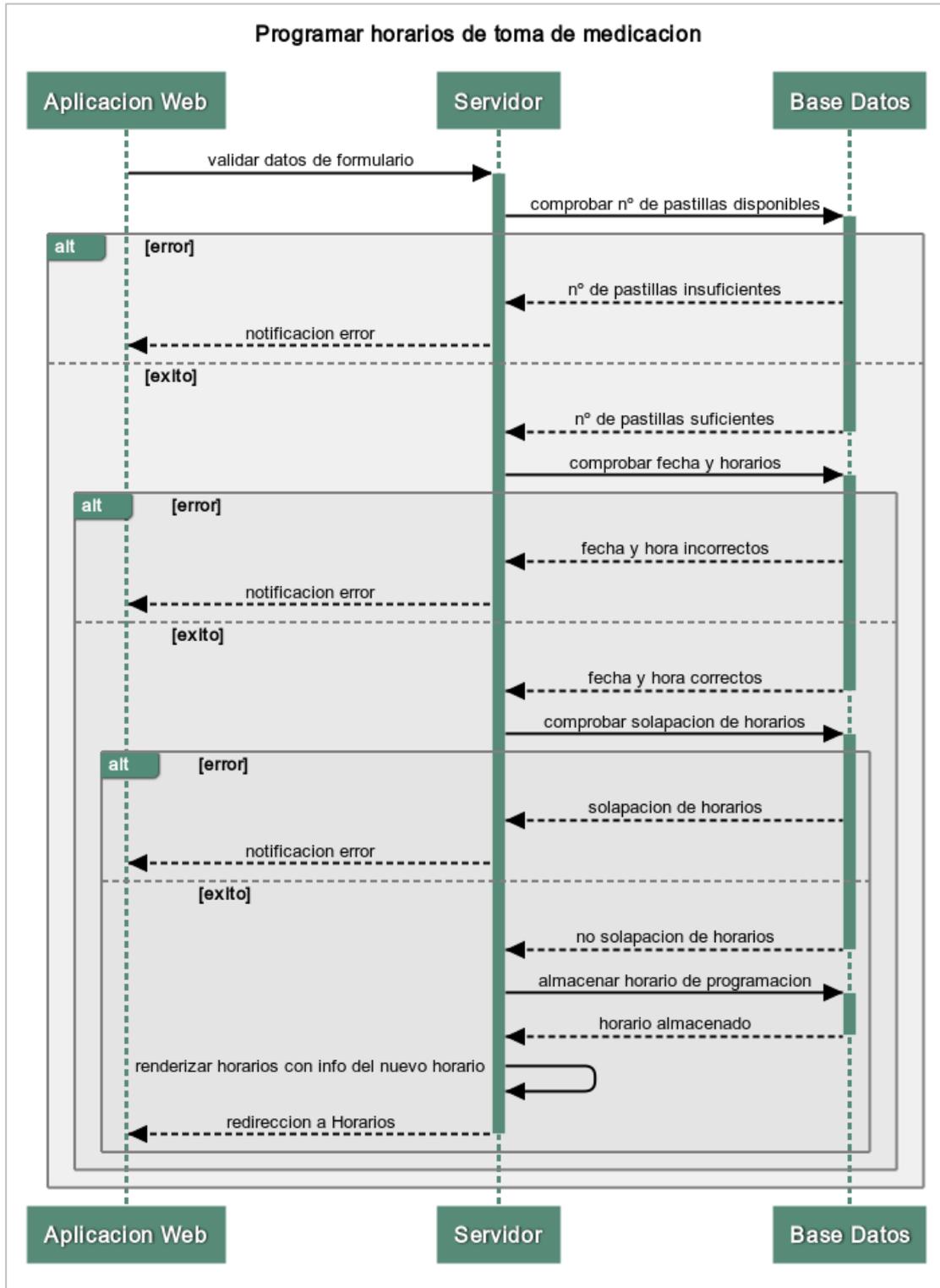


Figura 4.6. Diagrama de secuencia "Programar horarios de toma de medicación"

4.2.2.3 Diagrama de secuencia del caso de uso "Consultar horarios"

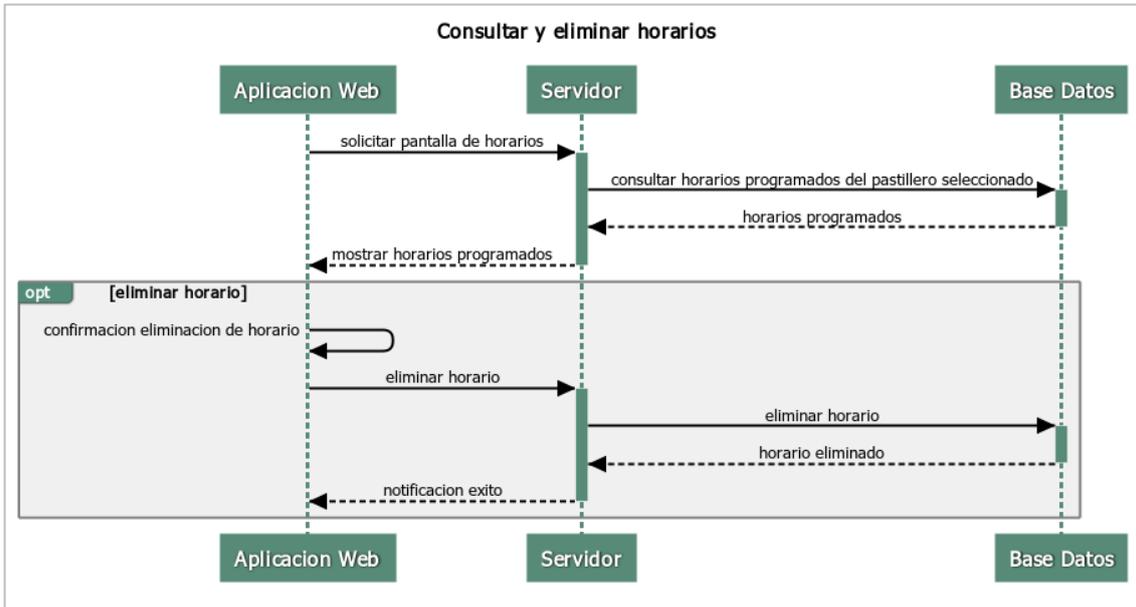


Figura 4.7. Diagrama de secuencia "Consultar y eliminar horarios"

4.2.2.4 Diagrama de secuencia del caso de uso "Visualizar información del dispensador"

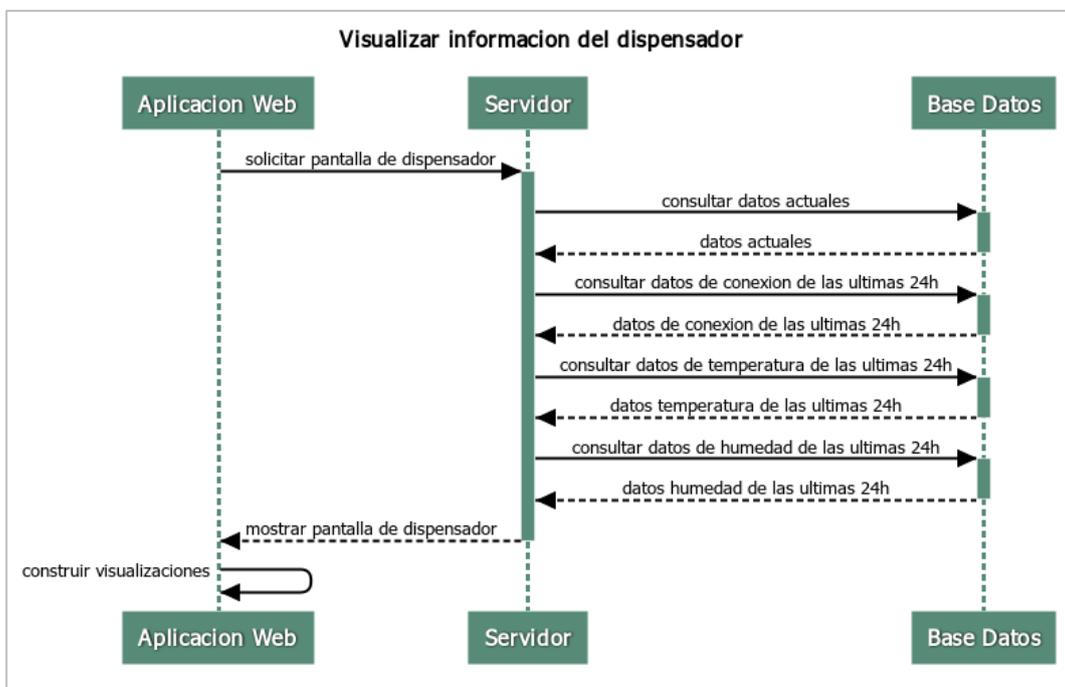


Figura 4.8. Diagrama de secuencia "Visualizar información del dispensador"

4.2.2.5 Diagrama de secuencia del caso de uso "Modificar datos personales"

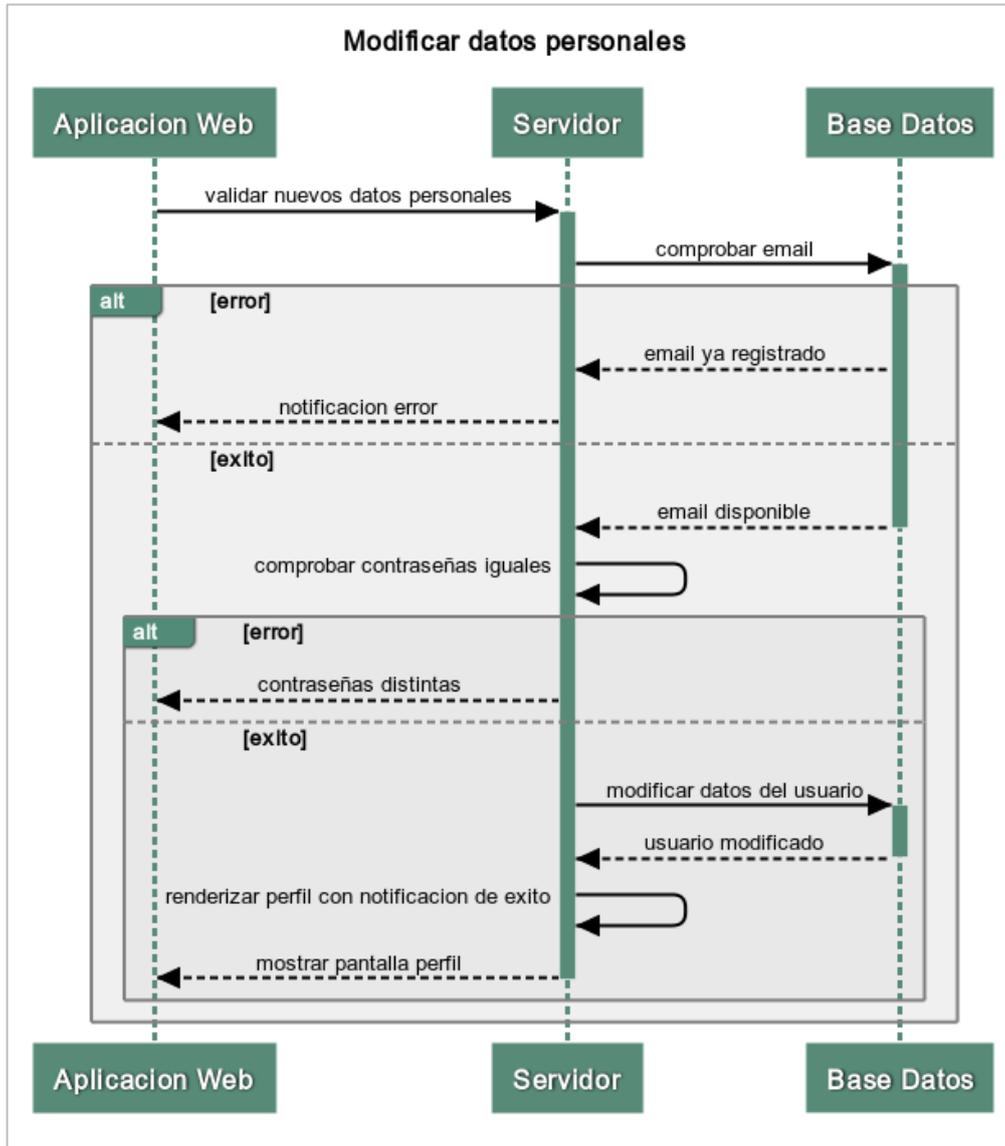


Figura 4.9. Diagrama de secuencia "Modificar datos personales"

4.2.2.6 Diagrama de secuencia del caso de uso “Especificar pastillas introducidas”

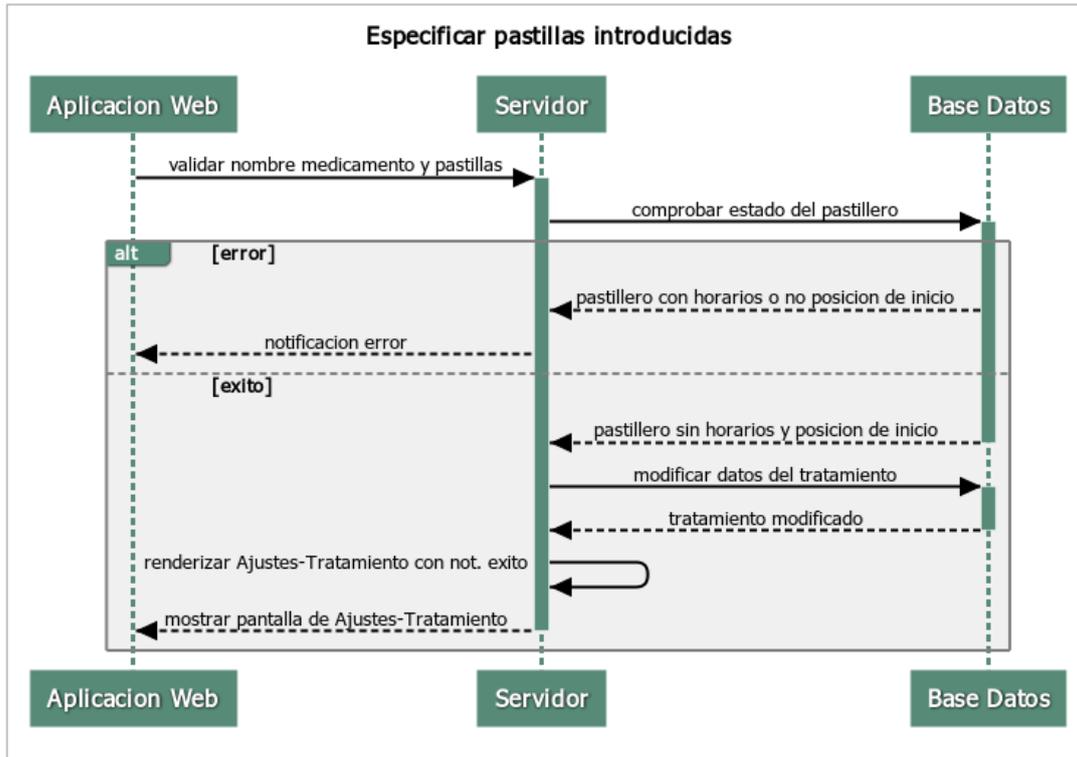


Figura 4.10. Diagrama de secuencia “Especificar pastillas introducidas”

4.2.2.7 Diagrama de secuencia del caso de uso “Configurar toma de medicación”

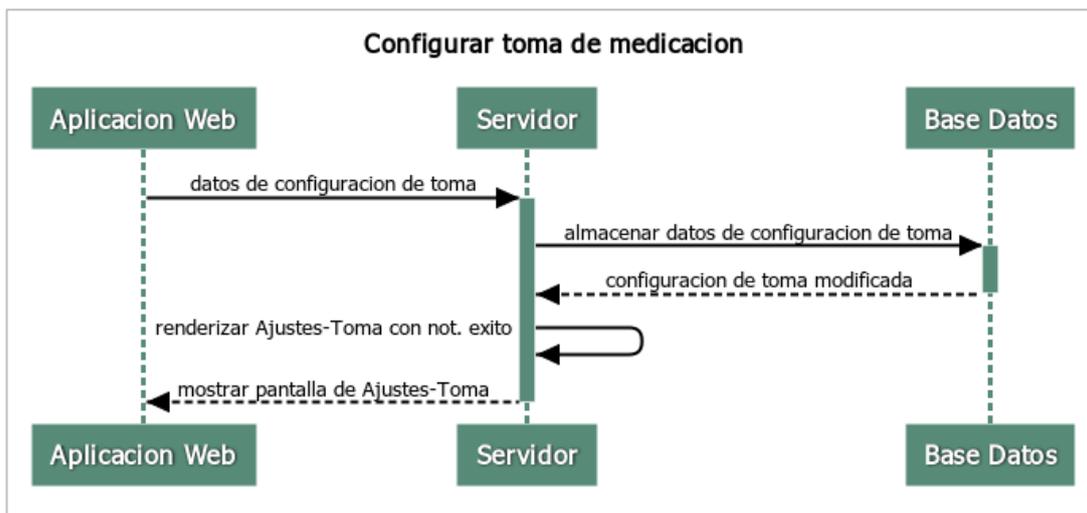


Figura 4.11. Diagrama de secuencia “Configurar toma de medicación”

4.2.2.8 Diagrama de secuencia del caso de uso "Configurar avisos de toma en el dispensador"

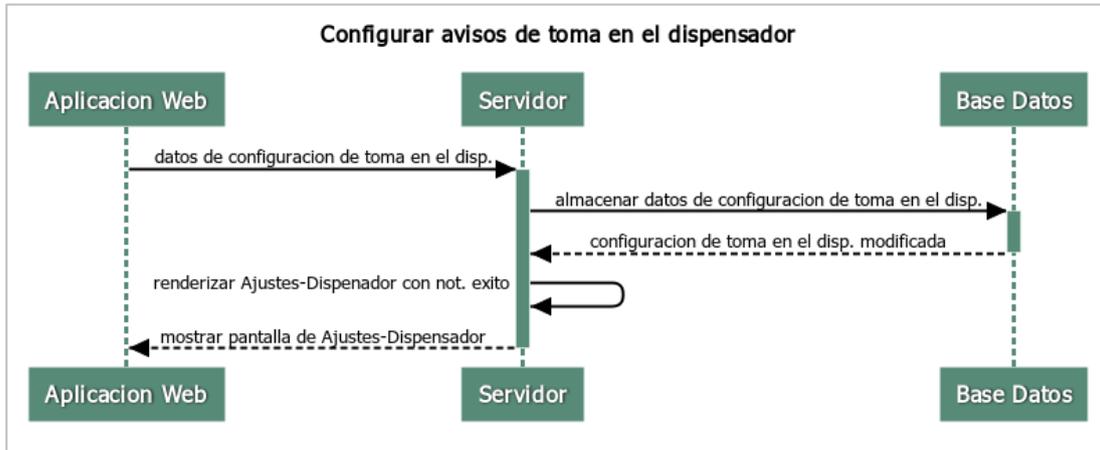


Figura 4.12. Diagrama de secuencia "Configurar avisos de toma en el dispensador"

4.2.2.9 Diagrama de secuencia del caso de uso "Añadir o eliminar emails de notificación"

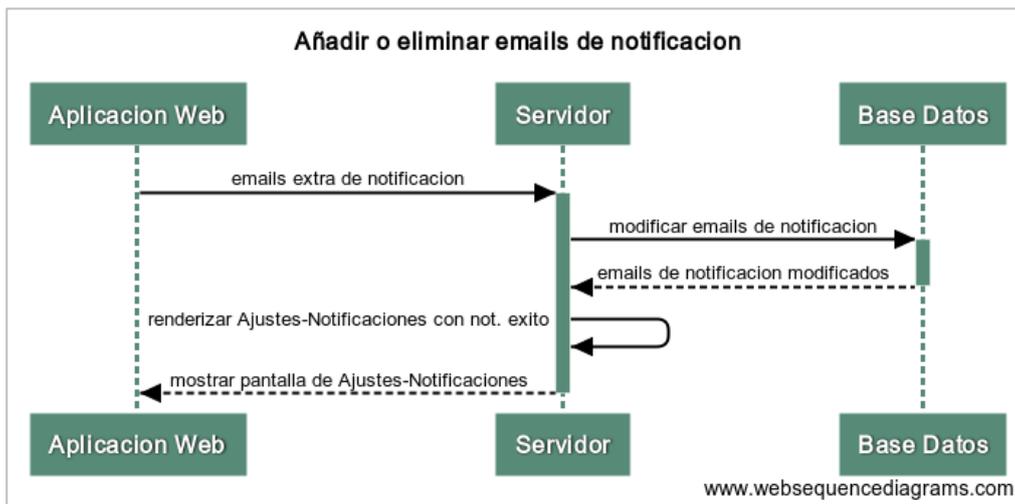


Figura 4.13. Diagrama de secuencia "Añadir o eliminar emails de notificación"

4.2.2.10 Diagrama de secuencia del caso de uso “Suscribirse al canal de Telegram”

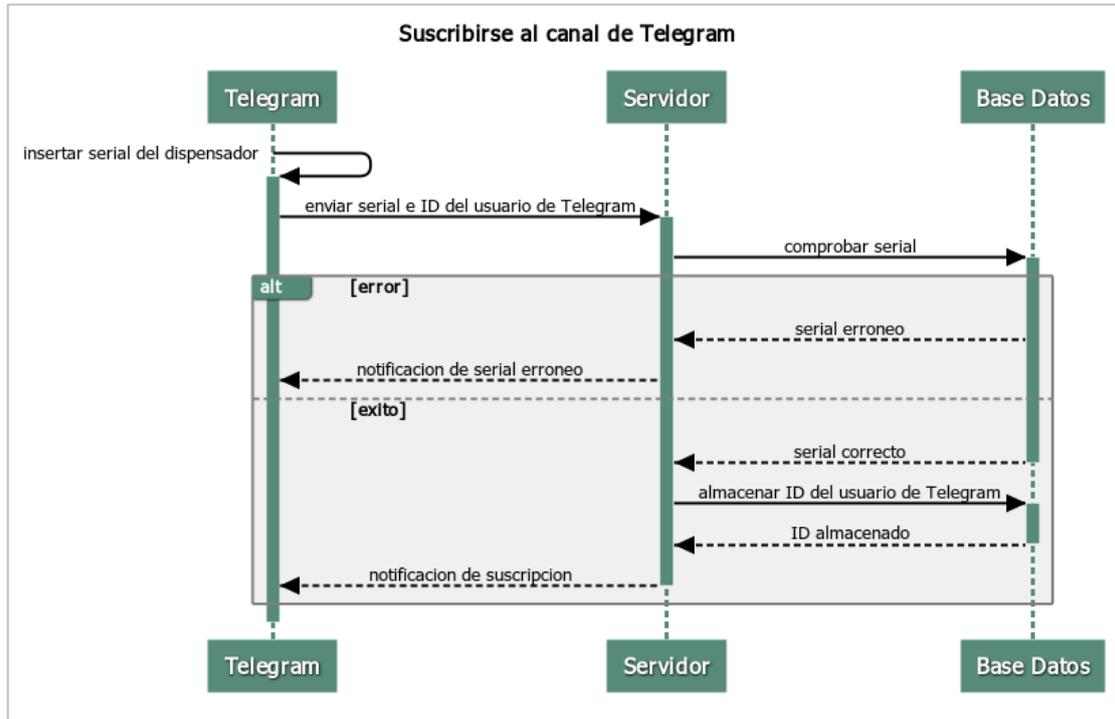


Figura 4.14. Diagrama de secuencia “Suscribirse al canal de Telegram”

4.2.2.11 Diagrama de secuencia del caso de uso "Configurar notificaciones"

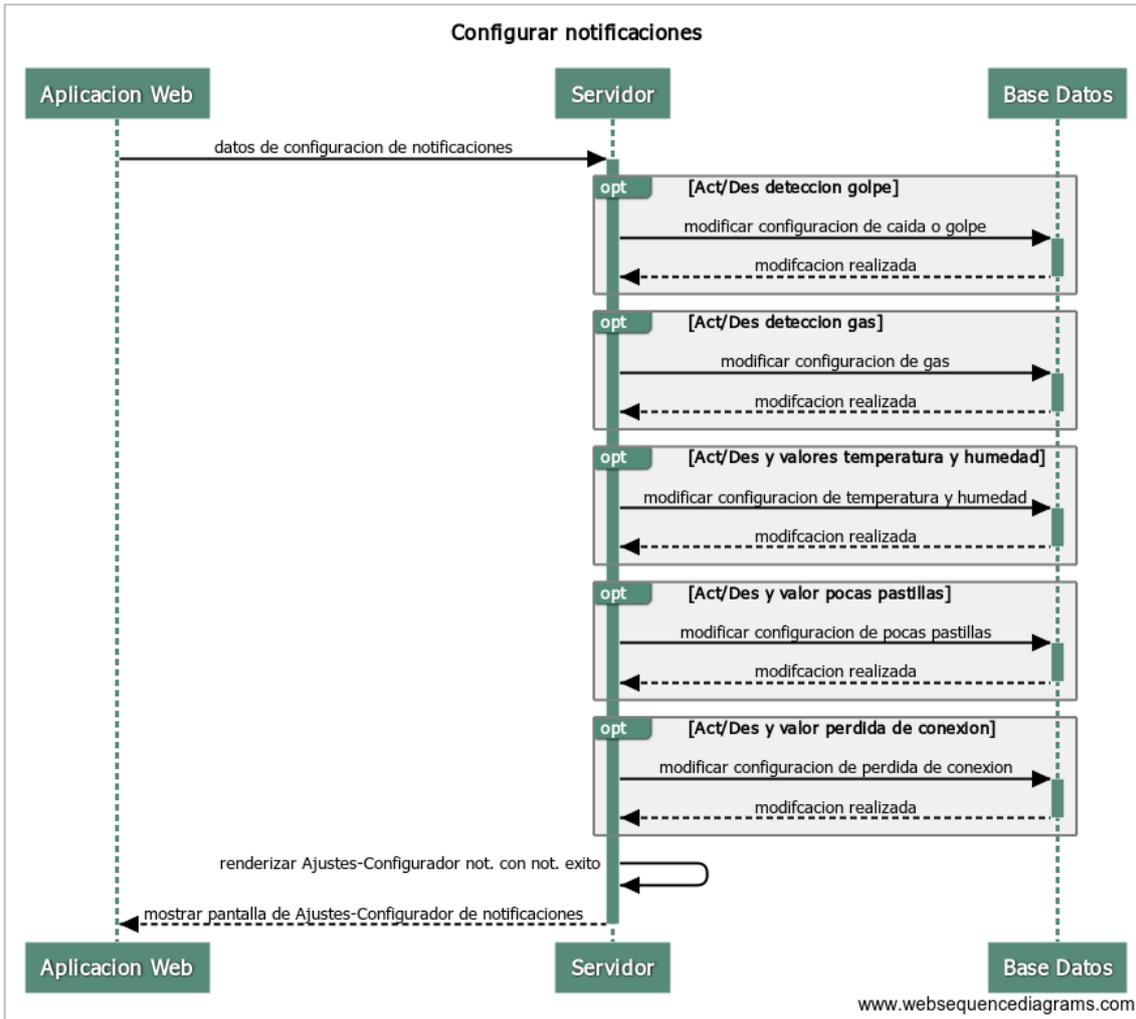


Figura 4.15. Diagrama de secuencia "Configurar notificaciones"

4.2.3 Diagramas de secuencia de los casos de uso del actor Usuario Administrador

4.2.3.1 Diagrama de secuencia del caso de uso "Añadir nuevos seriales de dispensador"

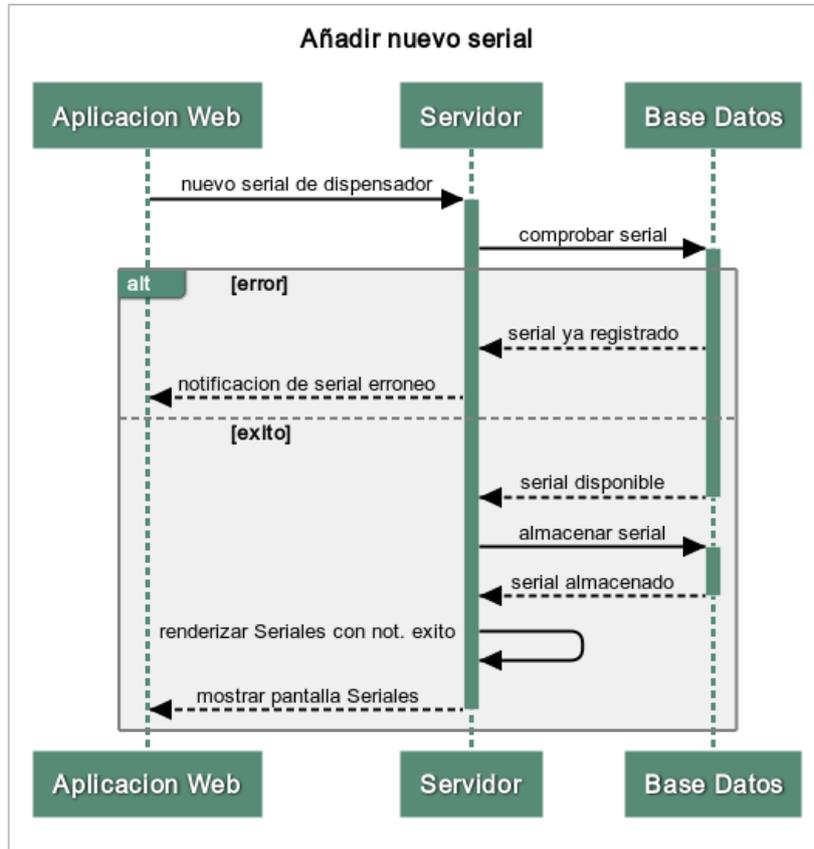


Figura 4.16. Diagrama de secuencia "Añadir nuevo de dispensador"

4.2.3.2 Diagrama de secuencia del caso de uso "Consultar seriales"

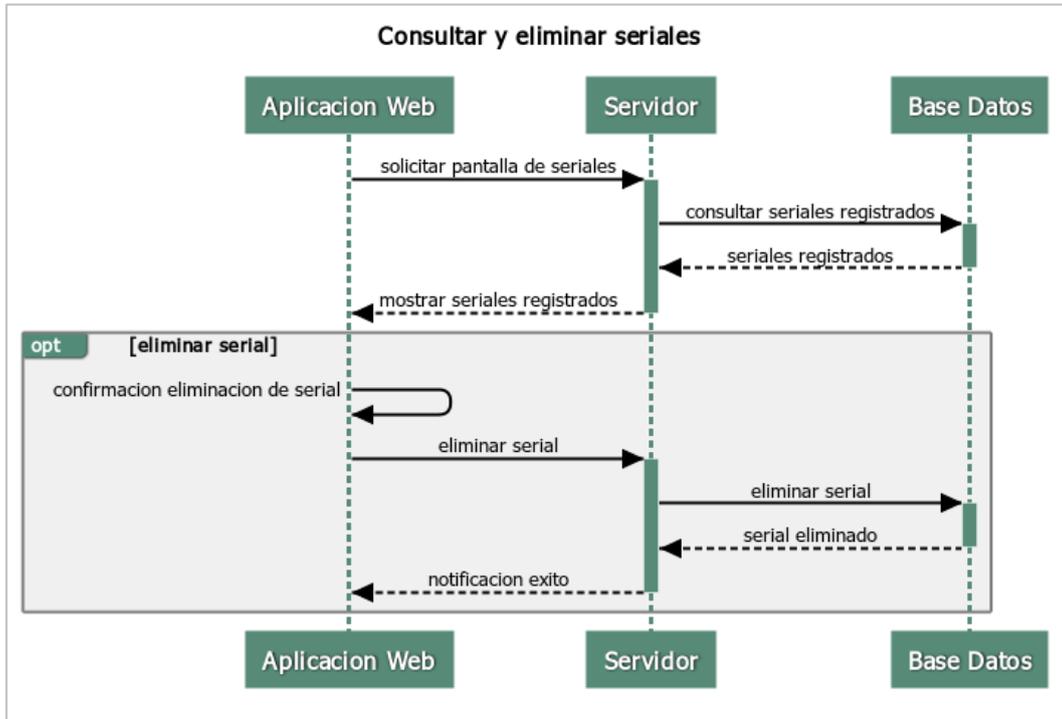


Figura 4.17. Diagrama de secuencia "Consultar y eliminar seriales"

4.2.3.3 Diagrama de secuencia del caso de uso "Consultar usuarios registrados"

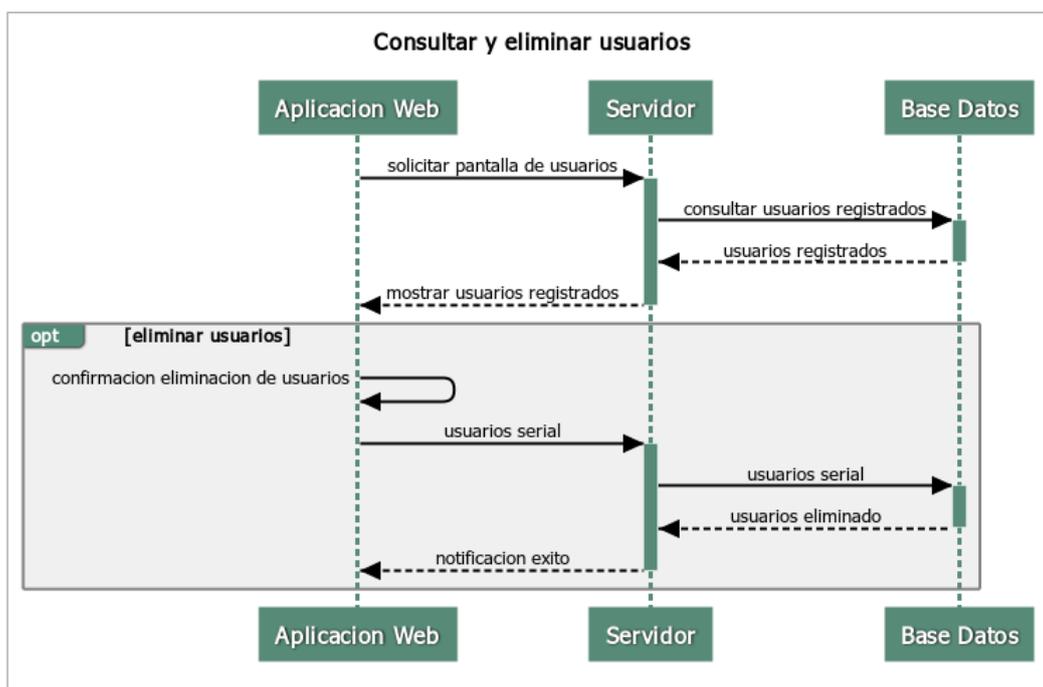


Figura 4.18. Diagrama de secuencia "Consultar y eliminar usuarios registrados"

4.2.3.4 Diagrama de secuencia del caso de uso "Modificar datos personales"

El diagrama de secuencia del caso de uso *Modificar datos personales* del usuario administrador es el mismo al especificado para el [usuario cliente](#).

4.2.4 Diagramas de secuencia de los casos de uso del actor Usuario del Dispensador

4.2.4.1 Diagrama de secuencia del caso de uso "Recoger y tomar medicación"

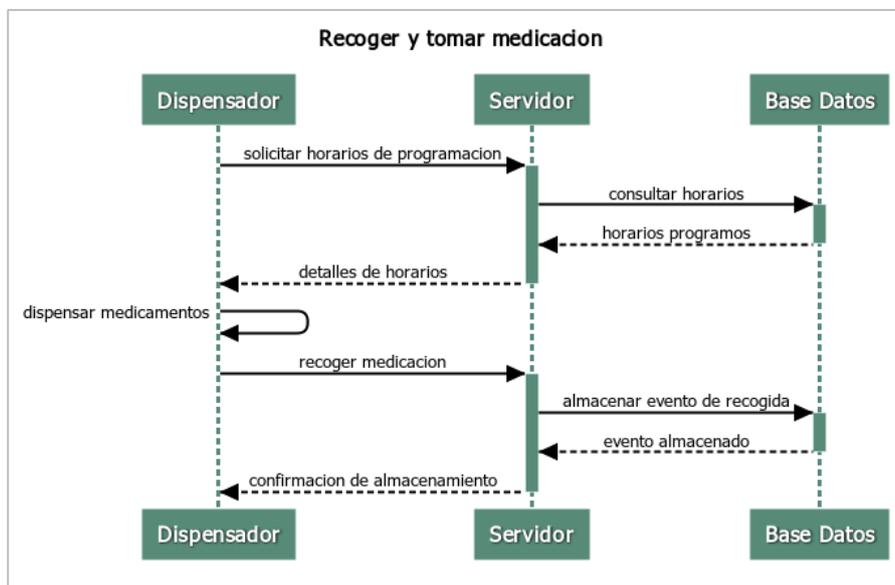


Figura 4.19. Diagrama de secuencia "Recoger y tomar medicación"

4.2.4.2 Diagrama de secuencia del caso de uso "Confirmar medicación"

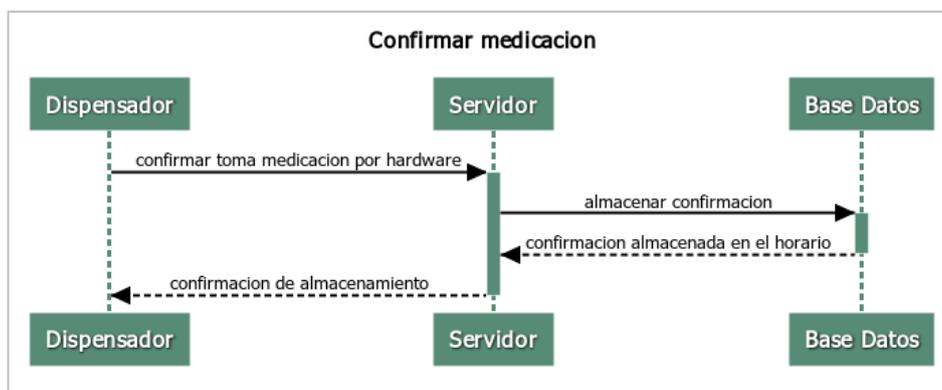


Figura 4.20. Diagrama de secuencia "Confirmar medicación"

4.2.4.3 Diagrama de secuencia del caso de uso "Emitir notificación de emergencia"

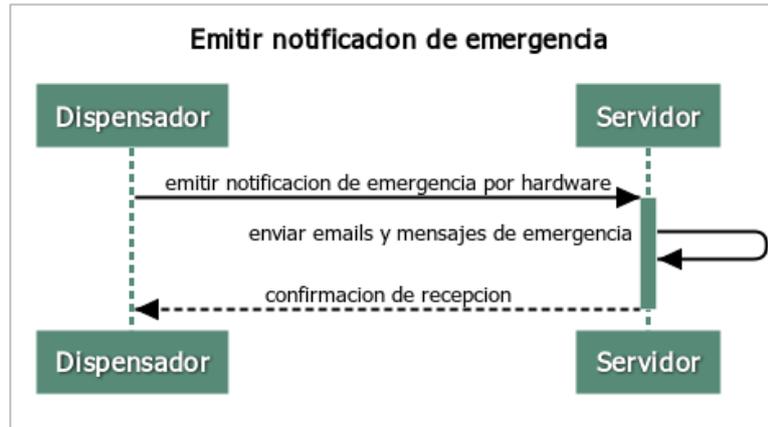


Figura 4.21. Diagrama de secuencia "Emitir notificación de emergencia"

4.2.5 Diagramas de secuencia de casos de uso del actor Dispensador

Los casos de uso del actor secundario *Dispensador* han sido tratados de manera especial. Al tratarse de un actor secundario, el dispensador no realiza cambios directamente sobre el sistema, sino que es el sistema el que realiza cambios sobre el dispensador.

Para el diagrama de secuencia se han unificado todos los casos de uso que involucran al dispensador. El principal motivo es la secuenciación de los casos de este actor ejecutados en conjunto en cada iteración del dispensador con el sistema.

Los casos de uso que fueron identificados para el actor secundario *Dispensador* y que han sido unificado en el diagrama como un único flujo de ejecución entre el dispensador y el sistema son:

- Expulsar pastillas
- Registrar estado de los pastilleros
- Registrar estado de conexión
- Almacenar datos de temperatura y humedad
- Almacenar datos de detección de gas o golpe
- Gestionar horarios de toma de medicación
- Emitir notificaciones sonoras y luminosas
- Configurarse

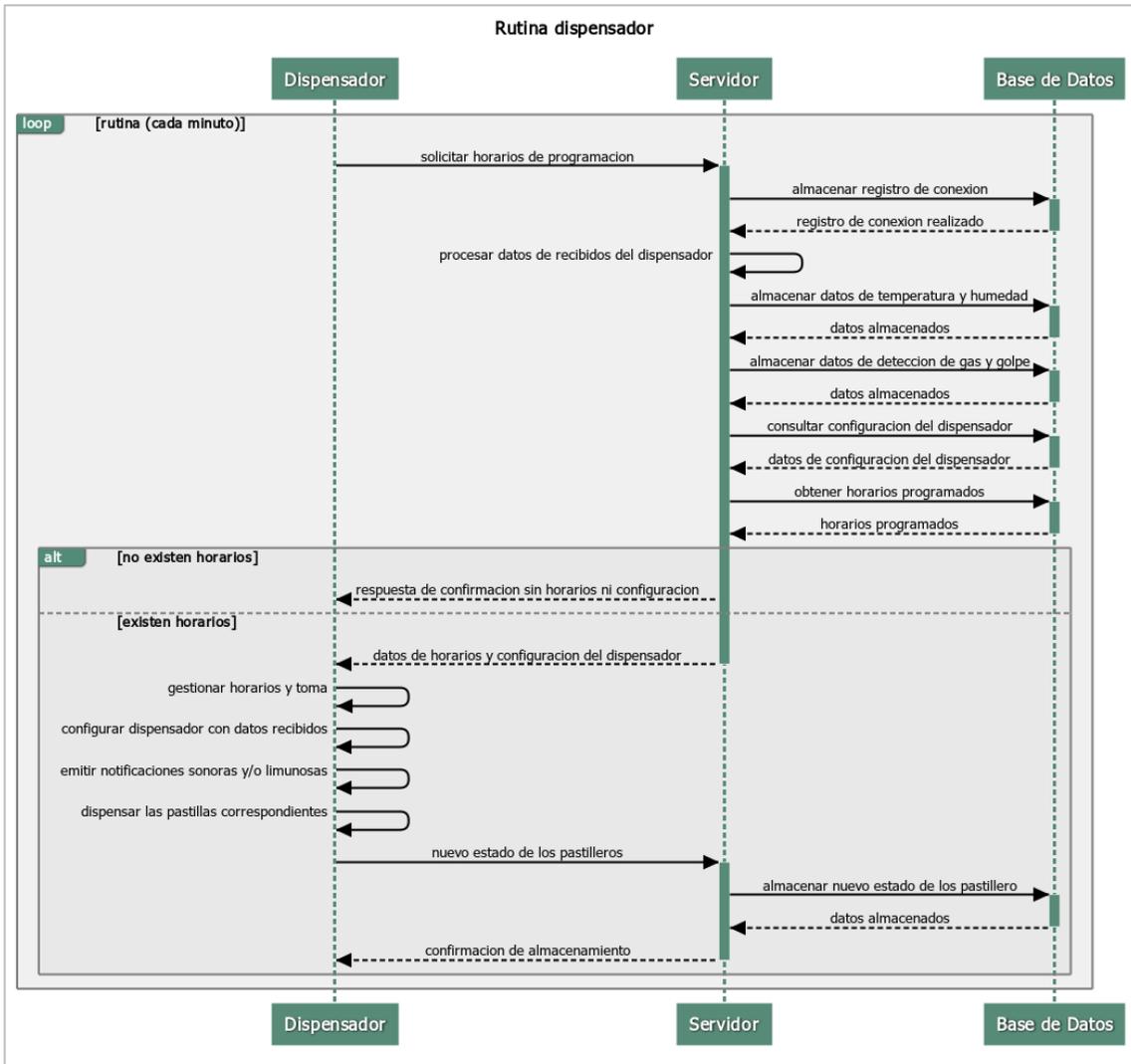


Figura 4.22. Diagrama de secuencia "Rutina del dispensador"

4.2.6 Diagramas de secuencia de casos de uso de los actores Servidor de correo y Telegram

4.2.6.1 Diagrama de secuencia del caso de uso "Emitir notificaciones"

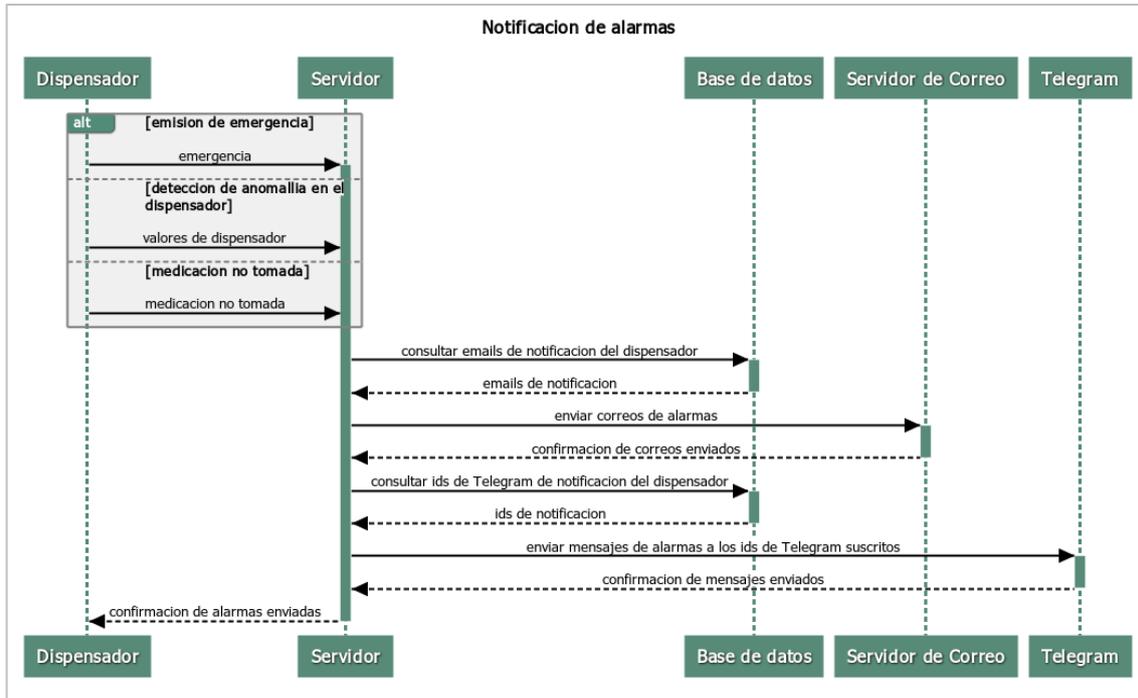


Figura 4.23. Diagrama de secuencia "Emitir notificaciones de alarmas"

4.3 Elección de Alternativas

En este capítulo se detallarán las alternativas elegidas para la ejecución del proyecto. Las opciones más condicionantes ya se analizaron en la sección *Estudio de alternativas*. Aquí se justificará el porqué de la elección de cada una de las alternativas y se analizarán sus necesidades.

Las elecciones a tomar y necesidades para el desarrollo del proyecto son numerosas y abarcan, desde componentes electrónicos y microcontrolador, hasta elecciones de desarrollo software frontend y backend.

4.3.1 Sistema Distribuido y Base de Datos - Backend

Las alternativas elegidas para el desarrollo de la parte backend del proyecto, que incluye entorno y lenguaje de programación en el que se implementará el lado del servidor y la elección de la base de datos que respaldará la información del sistema conectado, se expone a continuación.

4.3.1.1 Capa del servidor

El servidor será el núcleo de este proyecto y el encargado de coordinar todas las comunicaciones entre el resto de los dispositivos y componentes que engloban al sistema.

El entorno elegido para el desarrollo de la parte del servidor ha sido [Node.js](#). El principal motivo de su elección es por tratarse de un entorno de desarrollo mucho más flexible y funcional que sus competidores. Dispone de una gran cantidad de paquetes NPM que permiten incorporar nuevas funcionalidades al servidor de una manera muy sencillas. Además, la facilidad de enrutamiento entre páginas, la sencillez de comunicación con la base de datos y la experiencia personal en el desarrollo de aplicaciones con este tipo de tecnologías lo hacen un candidato perfecto para su uso en este proyecto.

Otra de las funcionalidades requeridas para este proyecto es la comunicación entre el microcontrolador del dispensador y el servidor por medio de una API de soporte. Pues bien, por su estructura de desarrollo, Node.js permite la creación de APIs REST de forma muy sencilla y funcional.

También cabe destacar que el servidor será el encargado de gestionar un sistema de alertas por medio de correos electrónicos y un canal de Telegram creado exclusivamente para este sistema. Por medio de estas alarmas se notificará a los usuarios de cualquier anomalía presente en el propio dispensador o en las inmediaciones del mismo.

4.3.1.2 Base de Datos

La base de datos será el lugar donde se almacenarán todos los datos del sistema conectado entre los que se incluyen datos de usuarios, de programación de horarios, seriales de dispensadores o datos ambientales recogidos de los dispensadores.

El servidor será el único encargado de comunicarse con la base datos. Por ello, la elección de la base datos ha sido condicionada al entorno elegido para el desarrollo de la parte del servidor.

La base de datos elegida para trabajar con el servidor Node.js no podía ser otra que [MongoDB](#). La integración de MongoDB con Node.js es total y existen varias librerías NPM que facilitan mucho la comunicación entre ambos, intercambiando datos en forma de documentos JSON.

Además, MongoDB es una base de datos no relacional que permite realizar, *en caliente*, cambios en los esquemas haciendo la integración de los datos con la aplicación una tarea sencilla y escalable.

4.3.2 Dispensador, Microcontrolador y Componentes Electrónicos

En esta sección se justificarán las decisiones tomadas en cuanto al hardware y construcción del prototipo. Todos estos componentes trabajarán el conjunto para abordar los requisitos del proyecto requeridos para el prototipo físico del dispensador de medicamentos.

4.3.2.1 Dispensador

El enfoque elegido para el diseño y construcción del prototipo es el de [dispensador](#). El enfoque de pastillero analizado limitaría demasiado el uso de la aplicación web que será desarrollada para este proyecto por la necesidad de colocar cada tipo de medicamento de forma secuencial para su futura toma, sirviendo únicamente como sistema de alarmas.

El enfoque de dispensador permitirá crear *a la carta* una toma de medicación en un momento concreto en el tiempo, con los medicamentos y cantidad de cada uno de ellos y sin necesidad de estar presente en el mismo lugar del prototipo debido a que esta programación se realizará por medio de una aplicación web desarrollada con este objetivo.

Inicialmente, el prototipo contará con dos *dispensadores* y cada uno de ellos dispondrá un tipo de medicamento. Para poder expulsar los medicamentos de cada uno de estos dispensadores, será necesario el uso de componentes electromecánicos que permitan mover ciertos elementos de los dispensadores para expulsar los medicamentos correspondientes. Mas adelante veremos qué motor eléctrico se selecciona para ello.

Cada uno de los dos dispensadores tendrá forma de *ruleta*. Contará con 12 posiciones o huecos donde se alojará una única pastilla por posición. Estos medicamentos estarán

contenidos por medio de un elemento móvil con un único orificio. Este componente girará y permitirá dispensar uno a uno los medicamentos que contiene.

De esta forma se soluciona el problema de las dimensiones de la pastilla, pudiendo alojar en el interior de cada una de las dos *ruletas* cualquier tipo de medicamento por muy grande o pequeño que sea.

4.3.2.2 *Microcontrolador*

El microcontrolador será el centro de control del prototipo de dispensador y por ello el elemento hardware más importante del proyecto. Uno de los requisitos fundamentales del microcontrolador es que disponga de módulo wifi con el que poder conectarse a un router que le proporcione internet para comunicarse con el servidor.

El microcontrolador elegido para este proyecto es Arduino [MEGA + ESP8266 WIFI](#). Este controlador dispone de dos módulos independientes, el módulo MEGA 2560 que posibilita la comunicación con todos los componentes electrónicos conectados al microcontrolador y el módulo ESP8266 WIFI que proporciona conexión a un router WiFi.

A pesar de que se trata de un microcontrolador de reciente aparición, con muy poca documentación asociada y de complicada programación, debido a que es necesario programar de forma independiente los dos módulos que componen el controlador y comunicarlos por medio del puerto serial del mismo, su capacidad y versatilidad lo hacen el candidato perfecto para su uso en este proyecto.

El principal motivo de su elección es debido a que dispone de 53 pines digitales para controlar todos los componentes electrónicos necesarios para el dispensador. Los 11 pines de los microcontroladores estándar no son suficientes para controlar todos los componentes electrónicos requeridos en este proyecto. Y por supuesto, esta placa dispone del módulo WiFi indispensable para realizar llamadas a la API del servidor e intercambiar información.

4.3.2.3 *Componentes Electrónicos y Sensores*

El dispensador contará con un gran número de componentes electrónicos, sensores, LEDs y muchos otros dispositivos que afrontan todos los requisitos del dispensador fijados para este proyecto.

En esta sección se elegirán algunos de componentes más importantes en cuando a la funcionalidad que desempeñan el prototipo.

4.3.2.3.1 *Motores*

Como ya se indicó anteriormente, el diseño y la construcción del prototipo seguirá un enfoque de dispensador. El prototipo contará con dos dispensadores de medicamentos y cada uno de ellos albergará con un motor encargado de dispensar las pastillas de cada una de las *ruletas* o dispensadores.

El motor elegido que será incorporado a cada dispensador es el [motor paso a paso](#). El principal motivo de su elección es que permite realizar desplazamientos angulares discretos con una resolución de 512 pasos en un vuelta completa o 360 grados.

La idea es que el motor esté acoplado a una pieza móvil circular con un único orificio. Sobre esta se posicionará la *ruleta* o dispensador que contendrá las 12 pastillas del mismo tipo. Cada vez que el motor gire una posición de la *ruleta*, la pastilla correspondiente a esa posición será dispensada. Es por este motivo por el que resulta crucial el control en los ángulos de giro del motor.

La elección de este tipo de motores también ha sido determinante para la elección del microcontrolador. Cada uno de los motores paso a paso necesita de 4 pines digitales para su control. 8 entradas junto a los pines que consumirán el resto de los componentes necesarios para el proyecto superan con creces las 11 entradas disponibles de los microcontroladores convencionales. De ahí la elección del microcontrolador MEGA con 53 pines digitales.

4.3.2.3.2 Detección de toma de medicación

La detección de si la medicación ha sido retirada del dispensador o no es otra de las importantes funcionalidades de este proyecto.

El sensor elegido para esta tarea es el [módulo infrarrojos IR](#). El principal motivo de su elección ha sido debido a que el resto de las alternativas analizadas no lograrían abordar esta funcionalidad correctamente, mientras que el sensor IR sería el candidato perfecto para esta tarea.

El principio de funcionamiento sería el siguiente: el dispensador expulsaría las pastillas correspondientes a una toma de medicación, dejando caer los medicamentos siempre sobre el mismo lugar del dispensador de donde será retirados. En esta parte del dispensador se encontraría instalado del sensor IR. Cuando el usuario proceda a retirar los medicamentos para su toma, el sensor detectaría la mano del usuario e informaría de esta acción al microcontrolador que tomaría las acciones oportunas al respecto.

De esta manera se podría garantizar, al menos, la intención de la toma de medicación correspondiente por parte del usuario, siendo otra funcionalidad de monitorización de los medicamentos y su correspondiente toma.

4.3.2.3.3 Integración de componentes

Debido al elevado número de componentes electrónicos que requiere el dispensador, el microcontrolador tendrá conectados numerosos cables de comunicación con cada uno de los componentes además de los correspondiente a la alimentación de los mismo.

Dejando todos los cables conectados a una protoboard²⁰ o similares, se corre el riesgo de que alguno de los cables se desconecte de su posición. Esto no solo haría que el

²⁰ Una placa de pruebas (protoboard o breadboard) es un tablero con orificios que se encuentran conectados eléctricamente entre sí de manera interna, en el cual se pueden insertar componentes electrónicos y cables para el armado y prototipado de circuitos electrónicos y sistemas similares.

dispensador no funcionara correctamente, sino que podría ocasionar un cortocircuito con otros cables dañando los componentes, o lo que es aún peor, provocar un incendio.

Para solucionar este problema, se ha elegido una [placa de expansión](#) que se conecta al microcontrolador y facilita las conexiones de todos los componentes electrónicos al mismo. También asegura que los cables permanezcan conectados y no sean retirados de su posición a menos que esta acción sea intencionada.

4.3.3 Aplicación Web – Frontend

La última parte con la que contará el sistema inteligente será la aplicación web de gestión desde la que se podrán programar todos los horarios de toma de medicamentos correspondientes a un dispensador, visualizar datos ambientales, anomalías y datos estadísticos y configurar un gran número de parámetros del mismo. Además de otras funcionalidades como el registro de usuarios o el panel de administración del sistema.

En esta sección se elegirán y descartarán frameworks, librerías y otras tecnologías para la parte frontend de la aplicación. Esto incluye el diseño, organización y estructura de la aplicación además de mejoras en la experiencia de usuario.

4.3.3.1 Frameworks y librerías Front-End

Para el desarrollo de la interfaz de usuario se ha decidido no usar ninguna librería ni framework del estilo Angular, React o Vue, debido principalmente a la no necesidad de estas para la ejecución del proyecto.

El servidor Node.js se encargará de renderizar cada una de las páginas de la aplicación web y servir las directamente al usuario para su visualización en el navegador en lugar de encargarse esta tarea al navegador del cliente. El propio servidor también se encargará de la navegación de la aplicación web. Para el resto de las animaciones y experiencia de usuario, se hará uso de JavaScript ejecutado en la parte del cliente junto a otras librerías que facilitarán esta tarea.

Para la manipulación del DOM de las páginas de la aplicación, manejar eventos, crear animaciones y el uso de la técnica AJAX con la que obtener una mejor experiencia de usuario, se hará uso de la librería [jQuery](#).

Como ya vimos anteriormente, la programación de horarios de toma de medicamentos es una de las principales funcionalidades del sistema. Por ello, será necesario controlar un gran número de fechas y horas, su manipulación, validación y localización. Para facilitar esta tarea se hará uso de la librería [Moment.js](#) tanto en la parte del cliente como en la del servidor.

Otras librerías JavaScript que serán utilizadas para su ejecución en el navegador del cliente son:

- **Bootstrap-confirmation.js**: empleado para mostrar una pequeña ventana modal de confirmación en las acciones de borrado, por ejemplo, de un horario programado o de un usuario del sistema desde la parte de administración.
- **Bootstrap-datetimepicker.js**: utilizado en los inputs de fechas y horas en la programación de horarios de toma de medicación.
- **Fontawesome**: aporta el uso de iconos como fuentes de la aplicación.
- **jQuery-maskedinput.js**: permite definir patrones en los campos de formularios y que será utilizado en los inputs de los seriales del dispensador que deben seguir un formato específico.

4.3.3.2 *Bootstrap y CSS puro para la organización de estilos*

En cuanto a la organización de las páginas de la aplicación web y su diseño, se ha decidido usar la librería [Bootstrap](#) para definir el layout de la aplicación, el diseño de otros muchos componentes visuales y facilitar la adaptación responsive de la aplicación.

Bootstrap será utilizado junto a gran cantidad de CSS3 puro con la que personalizar componentes y secciones visuales de la aplicación junto a animaciones respaldadas por código JavaScript que mejorarán el aspecto y la experiencia de usuario de la aplicación notablemente.

También cabe destacar que la tipografía que será utilizada en toda la aplicación será [OpenSans](#) por preferencia personal.

4.3.3.3 *Librería D3 para la creación de infogramas*

Una parte importante de la aplicación y que aporta gran valor a la misma es la visualización de los datos ambientales y anomalías del dispensador. Para crear infogramas y visualizaciones de estos datos se hará uso de la librería [D3.js](#).

A pesar de ser una librería con una curva de aprendizaje pronunciada, la versatilidad, flexibilidad y la gran comunidad de desarrolladores que la respalda la hacen la librería líder en cuando a la creación de visualizaciones dinámicas e interactivas.

Con esta librería se implementarán visualizaciones con datos correspondientes a las últimas 24 horas del tiempo de vida del dispensador (peticiones que ha realizado el microcontrolador del dispensador a la API del servidor) y variaciones en la temperatura y en la humedad.

En esta página de la aplicación también se podrá visualizar datos en tiempo real como la temperatura y humedad o conocer si el dispensador se encuentra conectado a Internet.

4.3.4 Conclusiones

En esta sección se han desglosado las decisiones de entornos, base de datos, diseños, componentes, sensores, controlador, tecnologías y librerías que serán utilizados para el desarrollo y construcción del sistema inteligente completo.

En las próximas secciones y apartados se estudiará más en detalle el sistema conectado. Esto permitirá entrar en detalle en el desarrollo, construcción e implementación de todos los subsistemas y elementos que componen el proyecto.

En resumen, el sistema completo constará de:

- **Backend – Sistema distribuido y base de datos:** núcleo del sistema inteligente que coordinará todas las peticiones tanto del dispensador como de la aplicación.
- **Hardware – Dispensador de medicamentos:** elemento del sistema que dispondrá de un microcontrolador alojado en su interior y una gran cantidad de componentes electrónicos y sensores que completarán la funcionalidad del mismo.
- **Frontend – Aplicación web:** será renderizada y servida al cliente por el servidor del sistema. Desde la aplicación web se gestionarán y programarán todos los aspectos del dispensador.

Con esta división se describirán en detalle los componentes, características, diseño y otros detalles de cada uno de los subsistemas y su comunicación desde una perspectiva de ejecución del proyecto.

4.4 Diseño de Clases del Sistema Distribuido

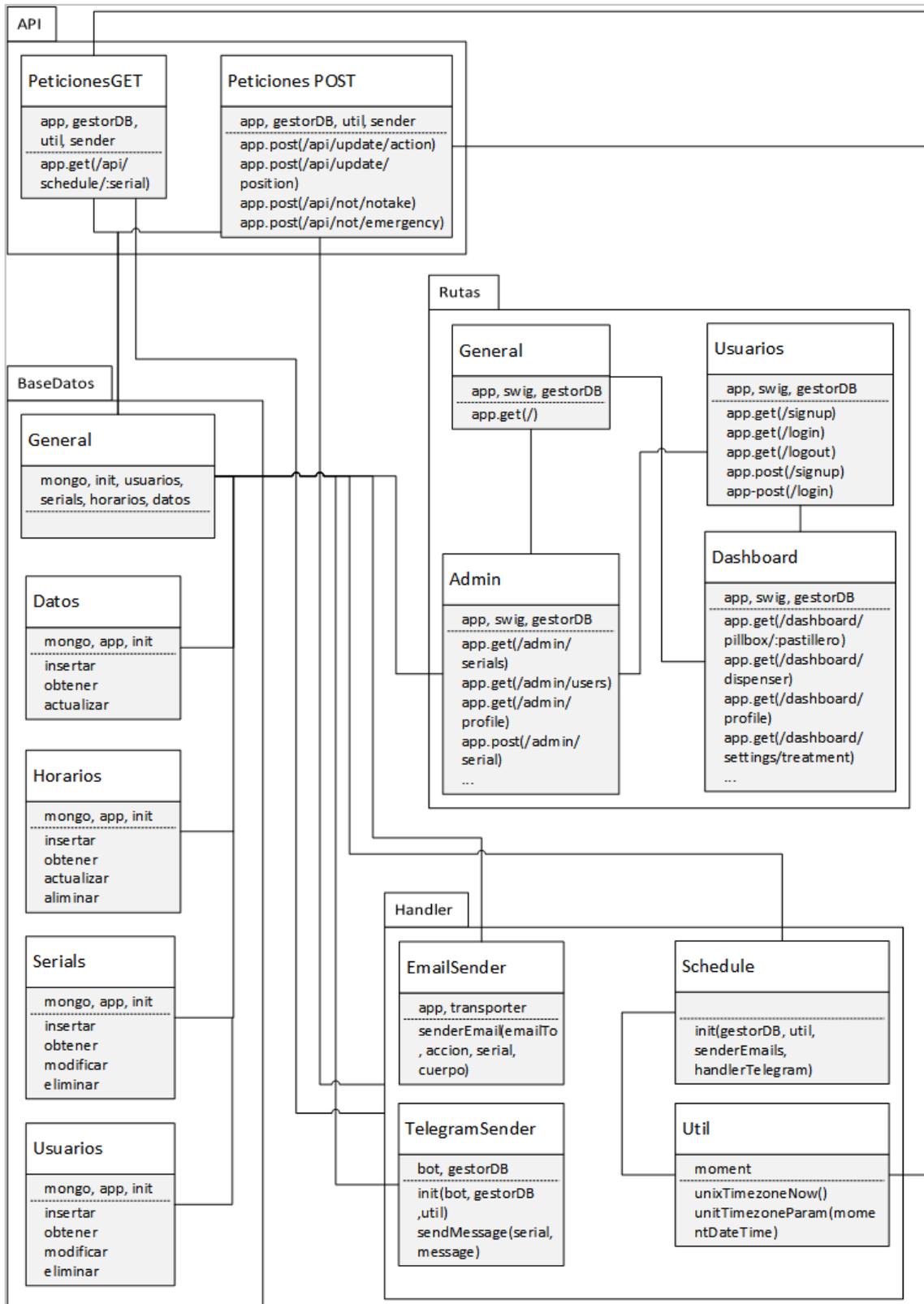


Figura 4.24. Diseño de clases del sistema distribuido

4.5 Diseño de las Funciones del Microcontrolador

Las funciones del microcontrolador se agrupan por cada uno de los módulos de programación. Estos son, el módulo ATmega2560 y el módulo ESP8266.

ATmega2560	
Función	Descripción
void setup()	Función encargada de inicializar todas las variables y pines del dispensador.
void serial3Event()	Esta función es la encargada de comunicarse con el módulo ESP8266 del microcontrolador a través del puerto serial.
controlMotores()	Función encargada del control de los motores del dispensador. Pararlos y arrancarlos.
eventosHardware()	Función encargada de procesar los eventos surgidos en los componentes electrónicos de entrada del microcontrolador.
voiz bip()	Esta función es la encargada de emitir un BIP de notificación a través del zumbador cuando del microcontrolador recibe un evento de componente electrónico de entrada.
String split(data, separator, index)	Función de utilidad para hacer un <i>Split</i> de un <i>String</i> . Devuelve la posición del <i>String</i> pedida a través del parámetro <i>index</i> .

ESP8266 WiFi	
Función	Descripción
void setup()	Función encargada de inicializar todas las variables y librerías necesarias para la programación del módulo.
void serial1Event()	Esta función es la encargada de comunicarse con el módulo ATmega2560 del microcontrolador a través del puerto serial.
void rutinaApiHorarios()	Rutina de solicitud de horarios de toma de medicación. El microcontrolador solicita cada minuto al servidor los horarios programados.
void comprobarToma()	Función encargada del control de la toma de la medicación dispensada.
void reseteaNecesariaToma()	Función para resetear todas las variables de control de tiempo de toma de una medicación dispensada.
conexionWifi()	Función encargada de establecer conexión a Internet a través del router wifi especificado.
void sendPostToApi(route, body)	Función que enviará las peticiones POST del microcontrolador a la API del servidor. Recibe la ruta de la API a la que lanzará la petición y el cuerpo de la petición POST.
String split(data, separator, index)	Función de utilidad para hacer un <i>Split</i> de un <i>String</i> . Devuelve la posición del <i>String</i> pedida

a través del parámetro *index*.

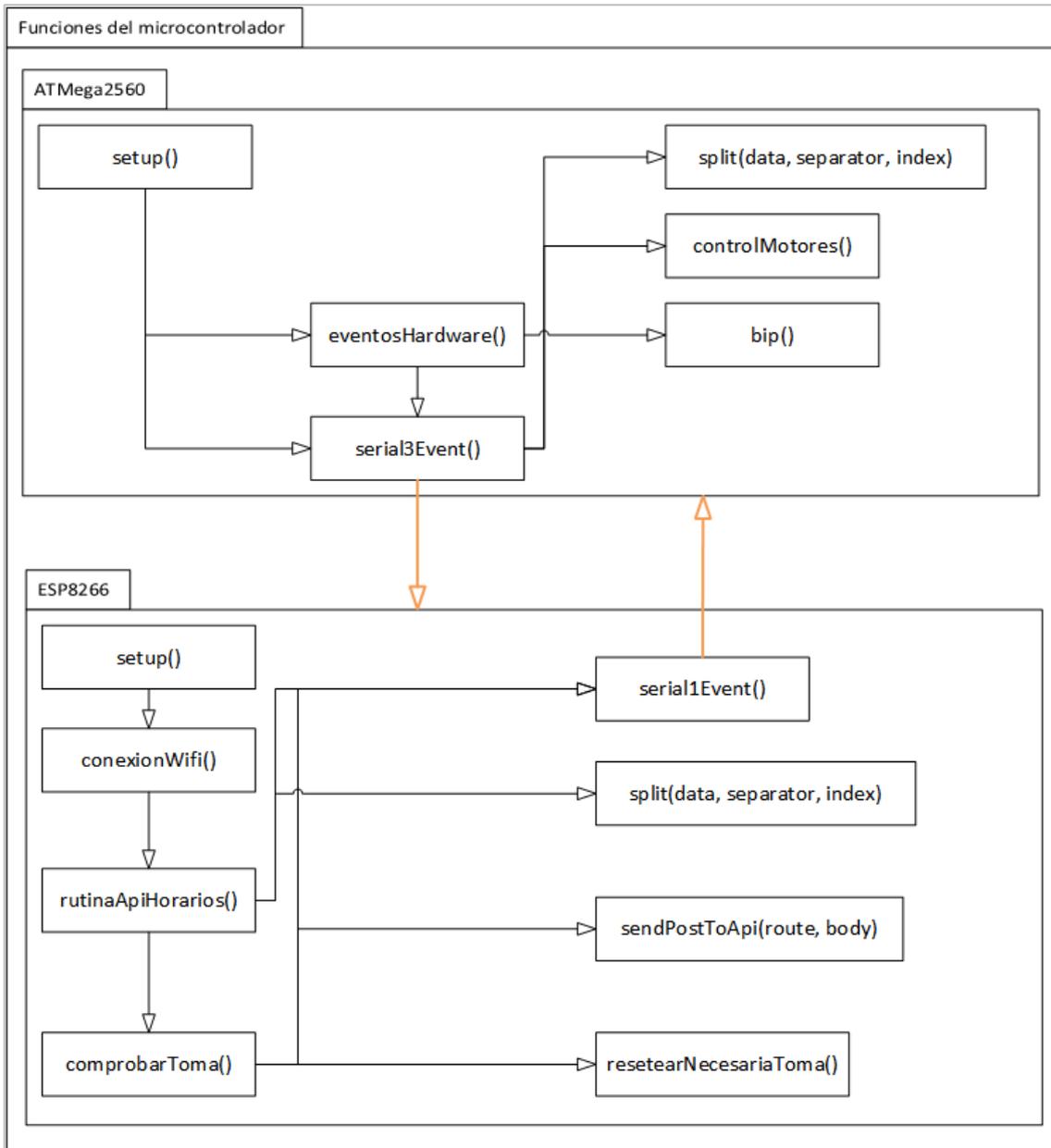


Figura 4.25. Interacción entre las funciones de los módulos del microcontrolador

4.6 Diseño del Sistema Distribuido – Backend

En esta sección se detallarán todos los aspectos relativos al diseño del sistema distribuido que orquesta el resto de los componentes del sistema conectado. Se detallará el funcionamiento de la API REST que será puerta de acceso a la comunicación con el microcontrolador del dispensador, los procesos ejecutados en la parte backend de la aplicación web y el sistema de notificaciones y alarmas.

El sistema distribuido del sistema conectado dispone de dos vías principales de comunicación: la comunicación por medio de rutas realizada desde la aplicación web y la efectuada por el microcontrolador del dispositivo mediante una API REST.

El servidor también es capaz de comunicarse con la instancia de base de datos MongoDB para el intercambio de información y con el servidor de correo Gmail y Telegram.

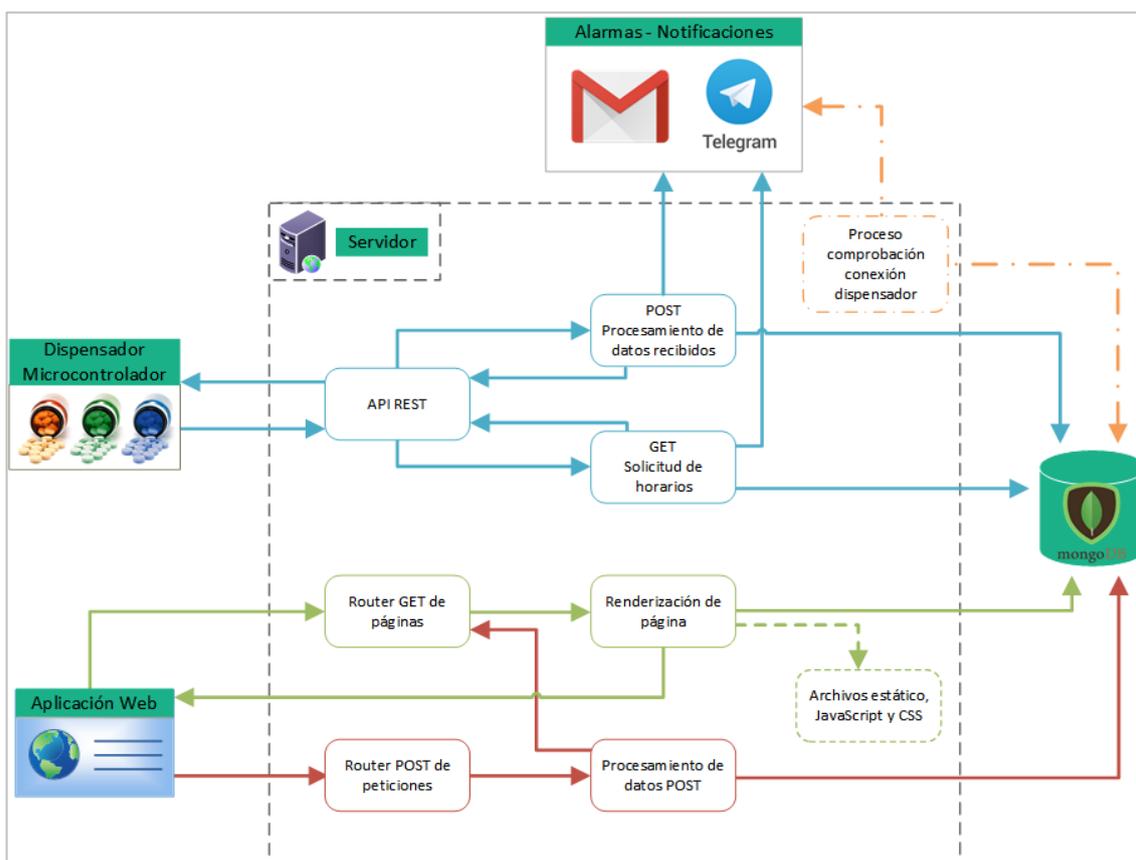


Figura 4.26. Diseño del sistema distribuido – Backend

En el esquema anterior podemos observar las principales funcionalidades y procesos que ejecuta el servidor Node.js cuando recibe peticiones del microcontrolador del dispensador y de la aplicación web. A continuación se detallarán estos procesos disparados por peticiones del dispensador y de la aplicación.

4.6.1 Aplicación web en el servidor

Los procesos ejecutados por peticiones realizadas por la aplicación web son:

- **Solicitud de página**

Al hacer click, por ejemplo, en una de las opciones de la barra de navegación, se solicita al servidor la página asociada a ese componente de navegación. El servidor recibe la petición de página, realiza peticiones a la base de datos para solicitar aquella información que deberá estar contenida en la página solicitada. El servidor renderiza la página con los datos recibidos e incorpora los archivos estáticos, CSS y JavaScript necesario para la página y que será ejecutado en el cliente. Por último, el servidor responde a la petición del navegador del cliente en forma de página renderizada con los estilos, archivos y JavaScript que será ejecutado en el navegador.

- **Envío de datos de formulario**

Cuando el usuario envía un formulario con datos al servidor, este recibe la petición POST con los datos introducidos por el usuario. El servidor ejecuta la lógica necesaria para procesar los datos recibidos e interacciona con la base de datos para insertar o actualizar la información recibida del formulario. Cuando finaliza todas las acciones necesarias para procesar los datos, el servidor dirige el control a una ruta asociada al procesamiento de la petición POST. Esta redirección no es más que la solicitud de la renderización de una página al igual que se solicitan páginas en la navegación por la aplicación. En este momento se ejecutan los mismos pasos que en la acción descrita en el punto anterior y el servidor responde a la petición POST del navegador del cliente con una página renderizada.

4.6.2 Microcontrolador del dispensador en el servidor

El microcontrolador del dispensador realiza llamadas a la API del servidor cada minuto para comprobar si existen horarios de toma de medicamentos correspondientes a la hora de la petición. También realiza otro tipo de peticiones POST a la API generadas por un evento en el dispensador.

- **Solicitud de horarios de toma de medicamentos**

El dispensador realiza cada minuto peticiones GET a la API del servidor para solicitar horarios de toma de medicamentos correspondientes a la fecha y hora de la solicitud. En la *query* de esta solicitud, el dispensador envía valores tales como la temperatura y humedad actual, detección de gas en las inmediaciones o si se ha detectado una caída, además del serial del dispensador

correspondiente. El servidor procesa esos datos, los almacena y en el caso de que detecte alguna anomalía en los datos recibidos, envía una notificación en forma de correo electrónico y mensaje de Telegram a los correos electrónicos registrados para el serial del dispensador y a los usuarios suscritos al serial del canal de Telegram correspondiente.

Cuando la información ha sido procesada, el servidor solicita horarios de toma de medicamentos a la base de datos correspondientes a la fecha y hora actual. El servidor recibe estos datos junto a las últimas configuraciones que se han realizado desde la aplicación web para el dispensador. Por último, el servidor responde a la petición del microcontrolador del dispensador en formato JSON con los detalles de la programación de los horarios para cada una de las dos *ruletas* del dispensador. En esta misma respuesta también es enviada la configuración del dispensador realizada por el usuario en la aplicación web.

- **Eventos surgidos en el dispensador**

Existen eventos ocasionados en el dispensador como la no toma de la medicación correspondiente, pulsar el botón de emergencia del dispensador o la actualización de la posición actual en la que se encuentran las dos *ruletas* del dispensador tras expulsar pastillas. Algunos de estos eventos originan una notificación a los correos electrónicos y canal de Telegram, otros únicamente actualizan información en la base de datos.

Estos eventos son enviados en forma de peticiones POST al servidor. El servidor procesa los datos recibidos del evento ocasionado y toma las acciones oportunas de envío de notificaciones de alarmas o actualización de información en la base de datos. Finalmente, el servidor responde al microcontrolador del dispensador para garantizar que la petición se ha procesado correctamente.

4.6.3 Proceso de comprobación de conexión del dispensador

En este punto cabe destacar el proceso que se ejecuta en segundo plano en el sistema distribuido y que se encarga de comprobar si el microcontrolador del dispensador está realizando las peticiones oportunas al servidor y de este modo garantizar que se encuentra conectado a internet y funcionando correctamente.

Para ello, solicita información a la base de datos de la fecha y hora de la última conexión del dispensador. En el caso de que se supere el tiempo máximo establecido sin conexión, el servidor notifica de este suceso a los usuarios del dispensador por medio de correos electrónicos y el canal de Telegram.

4.6.4 Sistema de Alertas y Notificaciones

El sistema de alarmas y notificaciones es una funcionalidad primordial y de gran valor en este proyecto. Notificar a los usuarios del dispensador de anomalías o de eventos importantes surgidos en el dispensador y en sus inmediaciones es uno de los requisitos de del proyecto.

Para ello, las notificaciones se realizan por dos vías diferentes:

- **Notificación por correo electrónico**

Los correos electrónicos de registro de usuarios y correos extra añadidos para el dispensador desde el panel de configuración de la aplicación web recibirán todas las notificaciones por medio de estos emails.



Figura 4.27. Ejemplo de alarma por correo electrónico

En la imagen anterior se puede ver un ejemplo de correo electrónico de notificación que envía automáticamente el servidor del sistema. En este caso, el correo notifica que el dispensador se encuentra sin conexión y no está realizando peticiones al servidor.

- **Notificación por mensajes al canal de Telegram**

Para la notificación por medio de mensajes de Telegram se ha desarrollado un canal o bot en dicha plataforma. Este bot tiene el nombre de [@SmartMedicineDispenserBot](#) y permite a los usuarios suscribirse a las notificaciones del serial de su dispensador. Para ello el usuario debe iniciar una conversación con el bot y seguir los pasos que le solicita para suscribirse. Estos pasos se resumen en facilitar el serial del dispensador al bot para que este notifique al servidor del id de usuario de Telegram que desea suscribirse y sea almacenado en la base de datos para futuras notificaciones.

El sistema distribuido es el encargado de enviar las notificaciones al bot de Telegram. El bot notificará a los usuarios suscritos de igual manera que lo hace una conversación entre dos usuarios. El servidor también recibe todos los mensajes que los usuarios envían al bot y actúa en consecuencia.



Figura 4.28. Ejemplo de notificaciones por el bot de Telegram

En la imagen anterior se puede ver un ejemplo de notificaciones realizadas por el bot de Telegram a un usuario suscrito al canal de su dispensador por medio de mensajes.

Las notificaciones de alarmas que se envían a los usuarios por correos electrónicos y mensajes a través del bot de Telegram son: botón de emergencia del dispensador pulsado, medicación no tomada, desconexión del dispensador, pocas pastillas restantes, anomalía en la temperatura o humedad de las inmediaciones del dispensador y si se ha detectado gas o una caída o golpe.

Todas estas notificaciones se pueden configurar en la sección de ajustes de la aplicación web que detallarán en las próximas secciones.

4.7 Diseño de la Base de Datos – Backend

La base de datos MongoDB será la encargada de almacenar toda la información relativa al sistema conectado de forma persistente. Su acceso se limita únicamente al servidor, quien inserta nuevos datos, actualiza, elimina y consulta muchos otros.

En esta sección se detallarán las colecciones de las que se compone la base de datos, esquemas con la información contenida en las colecciones y los propios datos que se almacenan.

MongoDB dispone de 4 colecciones que almacenan toda la información relativa a usuarios, dispensadores, horarios de toma de medicamentos y datos recogidos de los sensores y componentes electrónicos del dispensador.

- **Colección *usuarios***

Esta colección almacena los datos de los usuarios registrados en el sistema además del usuario administrador.

```

1.  {
2.    "_id" : ObjectId("5af9ec4722bcd50014bcada9"),
3.    "serial" : "NZHH3-DHK2W-8SX7I-SESB3-7RLDP",
4.    "nombre" : "Jose Antonio",
5.    "apellido" : "Cabaneros",
6.    "telefono" : "657459306",
7.    "tipo" : "USUARIO",
8.    "email" : "joseacabaneros@gmail.com",
9.    "password" : "d5dc619d365058bfc2273672c59d872a97419da40f1871bb4
    4bcda3cb24ca421"
10. }
    
```

```

▼ object {1}
  ▼ array {8}
    _id : 5af9ec4722bcd50014bcada9
    serial : NZHH3-DHK2W-8SX7I-SESB3-7RLDP
    nombre : Jose Antonio
    apellido : Cabaneros
    telefono : 657459306
    tipo : USUARIO
    email : joseacabaneros@gmail.com
    password : d5dc619d365058bfc2273672c59d872a97419da40f1871bb44bcda3cb24ca421
    
```

Figura 4.29. Árbol JSON usuario

Cada entrada de la colección contiene: un identificador único generado por Mongo, el serial del dispensador con el que se ha registrado el usuario, nombre, apellido y teléfono, tipo (usuario o administrador), email y contraseña.

Cabe destacar la encriptación de la contraseña. Todas las contraseñas de usuarios almacenadas en Mongo serán en formato encriptado. El servidor

Node.js será el encargado de encriptar las contraseñas antes de ordenar su almacenamiento en la base de datos y de desencriptarlas.

- **Colección *serials***

Esta colección almacena todos los datos relativos a cada uno de los dispensadores registrados en el sistema.

```

1.  {
2.    "_id" : ObjectId("5af9ec2d22bcd50014bcada8"),
3.    "serial" : "NZHH3-DHK2W-8SX7I-SESB3-7RLDP",
4.    "usuarios" : 1,
5.    "posicion" : {
6.      "A" : 0,
7.      "B" : 0
8.    },
9.    "ultimoping" : 1527544863,
10.   "emailsnotificacion" : [
11.     "",
12.     ""
13.   ],
14.   "idstelegram" : [
15.     990958
16.   ],
17.   "tratamiento" : {
18.     "A" : {
19.       "medicamento" : "Nolotil",
20.       "pastillas" : 8
21.     },
22.     "B" : {
23.       "medicamento" : "Paracetamol",
24.       "pastillas" : 6
25.     }
26.   },
27.   "toma" : {
28.     "tiempoespera" : 2,
29.     "irdeteccion" : true,
30.     "btnconfirmacion" : true
31.   },
32.   "dispensador" : {
33.     "lednotificacion" : true,
34.     "sonidonotificacion" : {
35.       "estado" : true,
36.       "valor" : 1
37.     }
38.   },
39.   "configuracionnotificaciones" : {
40.     "sinconexion" : {
41.       "estado" : true,
42.       "valor" : 6
43.     },
44.     "pocaspastillas" : {
45.       "estado" : true,
46.       "valor" : 2
47.     },
48.     "temperatura" : {
49.       "estado" : true,
50.       "min" : 13,
51.       "max" : 34
52.     },
53.     "humedad" : {
54.       "estado" : true,
55.       "min" : 30,

```

```

56.         "max" : 90
57.     },
58.     "detecciongas" : false,
59.     "deteccioncaida" : true
60. }
61. }
    
```

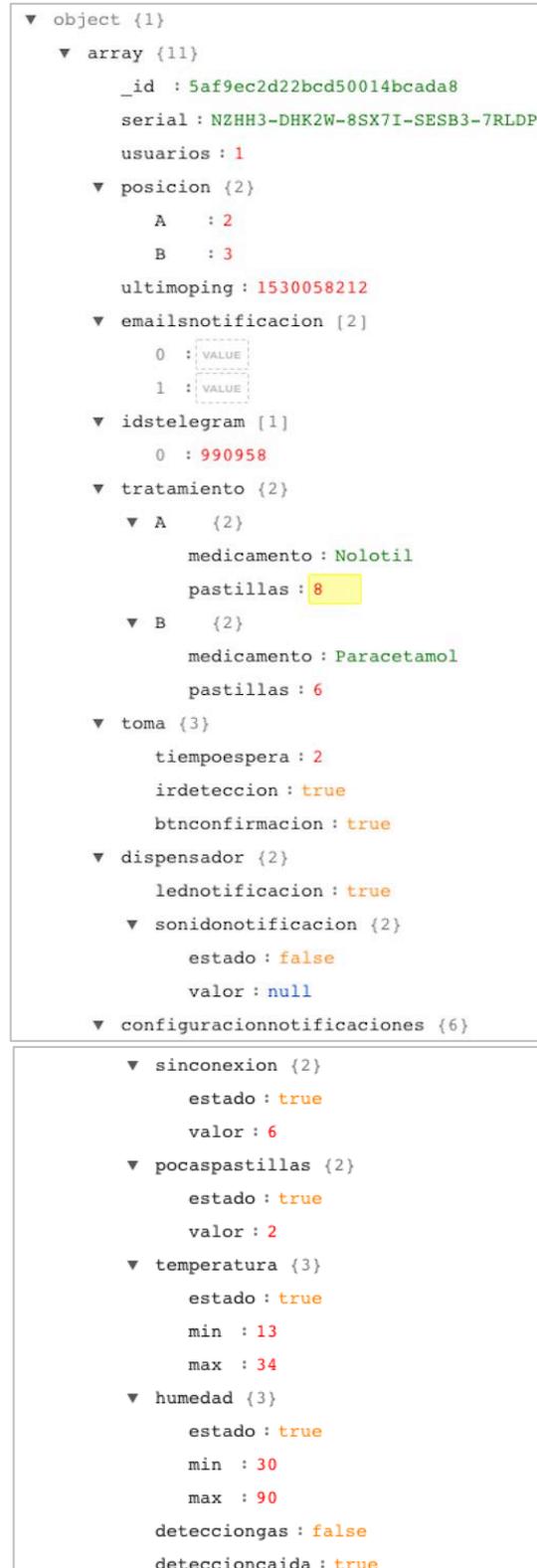


Figura 4.30. Árbol JSON dispensador

Los datos que se almacenan de cada serial registrado son: identificador único generado por Mongo, serial del dispensador, número de usuario registrados con este serial, posición actual de la *ruleta* A y B del dispensador, *UNIX time*²¹ correspondiente a la última conexión del dispensador, emails extra de notificación, identificadores de Telegram (cada uno corresponde a un usuario que se ha suscrito en el canal de Telegram a las notificaciones del serial del dispensador), tratamiento del dispensador (para cada *ruleta*, nombre del medicamento y número de pastillas), configuración de la toma de medicación (tiempo de espera para que la medicación sea tomada, si es necesario que el sensor IR detecte la toma y/o el botón de confirmación sea pulsado), configuraciones de las notificaciones sonoras y luminosas del dispensador y configuraciones de las propias notificaciones (cuándo y con qué valores se emitirá una notificación).

- **Colección *horarios***

Esta colección almacena los horarios de toma de medicación programados por los usuarios en la aplicación.

```

1. {
2.   "_id" : ObjectId("5afb0cad5fb72f0014ee54df"),
3.   "serial" : "NZHH3-DHK2W-8SX7I-SESB3-7RLDP",
4.   "unixtime" : 1526409480,
5.   "pastillero" : "B",
6.   "pastillas" : "2",
7.   "tomadaBtn" : true,
8.   "tomadaIR" : false
9. }
```



Figura 4.31. Árbol JSON horario

Cada entrada de esta colección es un horario programado y cuenta con la siguiente información: identificador único generado por Mongo, serial del dispensador para el que se ha programado la toma, *UNIX time* con la fecha y hora a la que se deben dispensar los medicamentos, pastillero o *ruleta* a la que afecta la programación, número de pastillas que deben dispensarse y dos

²¹ Tiempo Unix o Tiempo POSIX es un sistema para la descripción de instantes de tiempo: se define como la cantidad de segundos transcurridos desde la medianoche UTC del 1 de enero de 1970.

variables de control que será actualizadas según el comportamiento del usuario en la toma de la medicación correspondiente.

- **Colección *datos***

Esta colección almacena los datos recibidos del microcontrolador del dispensador en cada solicitud GET de horarios al servidor.

```

1. {
2.   "_id" : ObjectId("5afb09aa5fb72f0014ee54d7"),
3.   "serial" : "NZHH3-DHK2W-8SX7I-SESB3-7RLDP",
4.   "unixtime" : 1526408400,
5.   "pings" : 6,
6.   "temperatura" : 19,
7.   "humedad" : 59.04,
8.   "icalor" : 18.51,
9.   "gas" : false,
10.  "vibracion" : true
11. }
    
```

```

▼ object {1}
  ▼ array {9}
    _id : 5afb09aa5fb72f0014ee54d7
    serial : NZHH3-DHK2W-8SX7I-SESB3-7RLDP
    unixtime : 1526408400
    pings : 6
    temperatura : 19
    humedad : 59.04
    icalor : 18.51
    gas : false
    vibracion : true
    
```

Figura 4.32. Árbol JSON dato

Cada entrada de esta colección corresponde a 10 minutos de peticiones por parte del dispensador al servidor y contiene la siguiente información: identificador único asignado por Mongo, serial del dispensador, *UNIX time* con la fecha y hora a la que corresponden los datos, pings o número de solicitudes de horarios del dispensador al servidor, grados Celsius de temperatura e índice de calor, tanto por ciento de humedad y si se ha detectado gas o caída del dispensador.

Estas son todas colecciones contenidas en la base de datos MongoDB y sus esquemas que almacenan todos los datos e información del sistema conectado de forma persistente.

4.8 Diseño del Dispensador de medicamentos conectado – Hardware

El prototipo de dispensador de medicamentos conectado ha sido diseñado en 3D con el objetivo de ser impreso. El diseño cuenta con 19 piezas que deben ser impresas por separado y que deberán ser unidas para formar el prototipo completo.

Las dimensiones finales del prototipo serán de 19,3 cm de ancho, 13,7 cm de largo y 14,6 cm de alto, observando el dispensador por el frontal del mismo.



Figura 4.33. Diseño del dispensador de medicamentos conectado

En la imagen anterior puede observarse el prototipo de dispensador una vez impreso, ensamblado y con todos los componentes electrónicos ya instalados en su interior. Las *ruletas* del dispensador disponen de una letra y un color con el objetivo de facilitar su asignación en la aplicación web a la hora de programar horarios.

Los componentes distribuidos por el exterior del dispensador, etiquetados con un número en la imagen anterior son:

1. **Botones:** el botón verde será pulsado por el usuario para confirmar que ha tomado la medicación y el botón rojo será pulsado en caso de emergencia.
2. **Sensor IR:** es el sensor encargado de detectar si la medicación ha sido retirada de la cavidad donde se dispensan los medicamentos.

3. **LED de notificación:** este LED permanecerá encendido cuando sea necesario tomar medicación y hasta que esta medicación sea retirada del dispensador y confirmada por medio del botón verde.
4. **Zumbador:** emite sonido de notificación cuando toque tomar la medicación. Solo permanecerá emitiendo sonido unos pocos segundos.
5. **LED de conexión:** este LED permanecerá encendido mientras el microcontrolador esté conectado a una conexión WiFi con la que tener acceso a Internet.
6. **Sensor ambiental:** es el sensor encargado de medir la temperatura y humedad del ambiente.



Figura 4.34. Diseño del dispensador de medicamentos – Parte trasera

En la parte trasera del dispensador se encuentran los siguientes componentes:

7. **Sensor de gas:** es el sensor encargado de detectar algún tipo de gas en el ambiente.
8. **Alimentación:** conector de entrada de alimentación del microcontrolador.
9. **Programación:** puerto micro USB a través del cual se programa el microcontrolador.

En el interior del prototipo se encuentran otros muchos componentes electrónicos y sensores que no han sido descritos por el momento. En el siguiente apartado se verá en detalle el esquema de todos los componentes electrónicos, motores y microcontrolador y sus conexiones.

4.8.1 Sistema de control del Dispensador

Los componentes electrónicos, motores, sensores y otros componentes hardware conectados al microcontrolador para abordar la funcionalidad del dispensador de medicamentos son numerosos. En este apartado se presentan todos los componentes, sus conexiones y cómo se distribuyen por el dispensador.

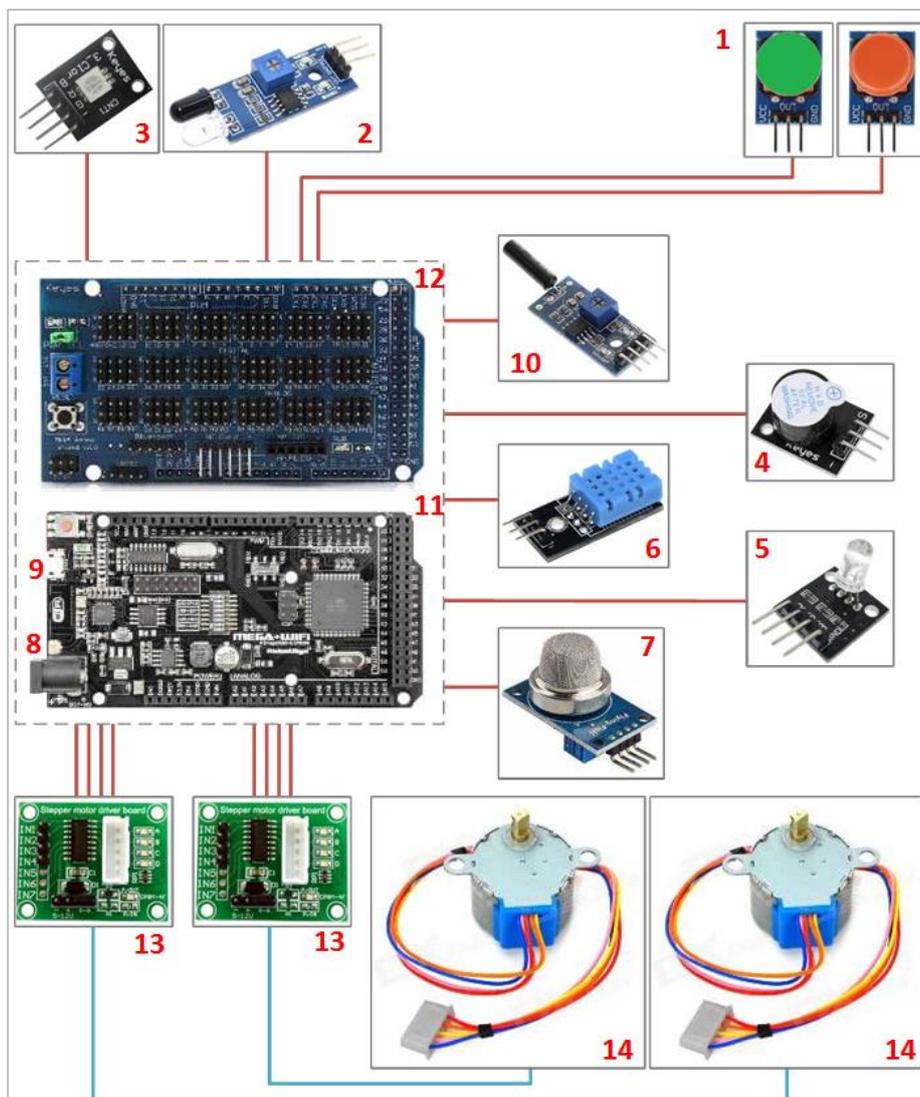


Figura 4.35. Esquema de conexiones del dispensador

En este punto cabe destacar que los números contenidos en cada uno de los componentes del esquema concuerdan con los vistos en la [sección anterior](#). Por ello, solo se detallarán aquellos componentes que no fueron vistos anteriormente:

- 10. **Sensor de vibración:** este sensor se encuentra en el interior del dispensador y sirve para detectar caídas o golpes sufridos por el aparato.
- 11. **Microcontrolador:** núcleo del dispensador que se encarga de comunicarse con todos los componentes conectados a él además de establecer comunicación con el servidor a través de Internet.

12. **Placa de expansión:** placa conectada al microcontrolador y que sirve de intermediario entre los componentes y el microcontrolador. Su única función es facilitar la conexión y asegurar la fijación del cableado de los componentes.
13. **Placas de control:** son los *drivers* que controlan los motores paso a paso.
14. **Motores paso a paso:** son los motores insertados en las *ruletas* y los encargados de la acción mecánica de dispensar los medicamentos.

En el esquema anterior, cada una de las conexiones de color rojo es una conexión a un pin digital del microcontrolador. Como puede observarse, los pines digitales utilizados para el control de todos los componentes son 17. Además de estas conexiones, cada uno de los componentes necesita de otras dos conexiones con el microcontrolador correspondientes a la alimentación y que no han sido representadas en el esquema.

A nivel global, los componentes conectados al microcontrolador son 11, y los pines digitales consumidos por estos componentes 17. Sumando los dos cables de alimentación necesarios para cada componente y los 17 de las conexiones digitales, hacen un total de 39 cables de conexión entre todos los componentes y el microcontrolador.

```

7  /***** DEFINES *****/
8  #define PIN_LED_NOT_TOMA 46
9  #define PIN_ZUMB_NOT_TOMA 25
10 #define PIN_LED_WIFI_OK 29
11 #define PIN_BTN_CONF 50
12 #define PIN_BTN_EMER 52
13 #define PIN_IR_TOMA 48
14 #define PIN_GAS 33
15 #define PIN_VIB 30
16 #define PIN_DHT 22
17
18 //Define para los pines del MOTOR PASO A PASO 'A'
19 #define motorAPin1 42 // IN1 on the ULN2003 driver A
20 #define motorAPin2 43 // IN2 on the ULN2003 driver A
21 #define motorAPin3 44 // IN3 on the ULN2003 driver A
22 #define motorAPin4 45 // IN4 on the ULN2003 driver A
23
24 //Define para los pines del MOTOR PASO A PASO 'B'
25 #define motorBPin1 38 // IN1 on the ULN2003 driver B
26 #define motorBPin2 39 // IN2 on the ULN2003 driver B
27 #define motorBPin3 40 // IN3 on the ULN2003 driver B
28 #define motorBPin4 41 // IN4 on the ULN2003 driver B
29 /*****/

```

Figura 4.36. Pines de conexión de los componentes al microcontrolador

En la imagen anterior puede comprobarse a que número de conexión digital o PIN va conectado cada uno de los componentes especificados en el [esquema de conexiones del dispensador](#).

4.9 Diseño de la Interfaz de la Aplicación web – Frontend

En este capítulo se presentará el diseño y aspecto de la aplicación web junto a una breve descripción de sus funcionalidades. Además, se expondrán las características más reseñables de la interfaz de la aplicación.

4.9.1 Diseño y componentes de la interfaz de usuario

A continuación, se muestran capturas que recogen los diseños de la interfaz de usuario y que componen la aplicación web. Cada una de ellas estará acompañada de una breve descripción del papel que desempeñan en el sistema, así como de otros aspectos que resulten de interés.

4.9.1.1 Interfaz de usuario – Login

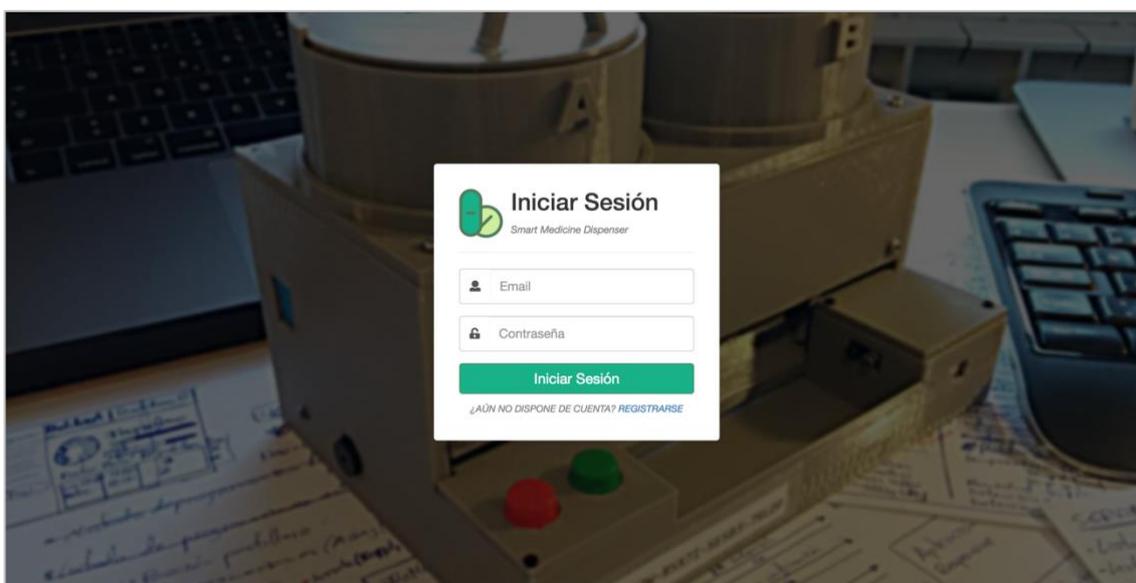


Figura 4.37. Diseño de la interfaz – Login

La ventana de *inicio de sesión* en la aplicación cuenta con un formulario para introducir el correo electrónico del usuario registrado y su contraseña. En esta misma ventana de login también puede iniciar sesión un usuario administrador con su correspondiente redirección al panel de administración.

Haciendo click sobre *Registrarse* se redirige al usuario al formulario de registro de la siguiente sección.

4.9.1.2 Interfaz de usuario – Registro

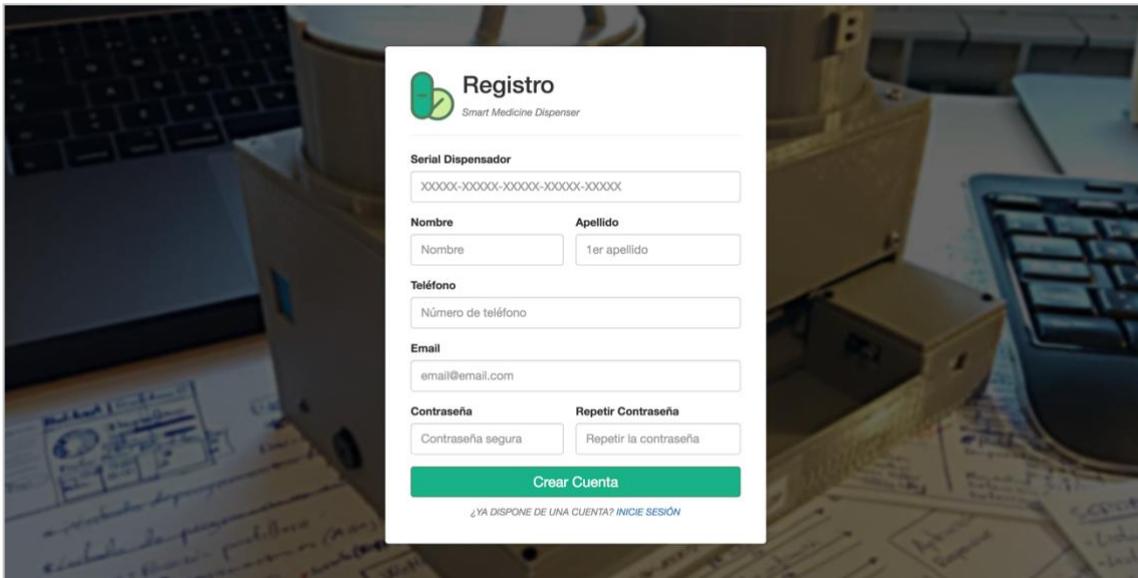


Figura 4.38. Diseño de la interfaz – Registro

En la ventana de *registro*, el usuario debe introducir el serial de su dispensador, nombre y apellido, teléfono de contacto, correo electrónico que servirá para iniciar sesión en la aplicación y como centro de notificaciones, y contraseña de inicio. El serial introducido debe estar registrado en el sistema para poder completar el registro. Además, solo se permite el registro de un máximo de dos usuarios por serial.

4.9.1.3 Interfaz de usuario – Horarios

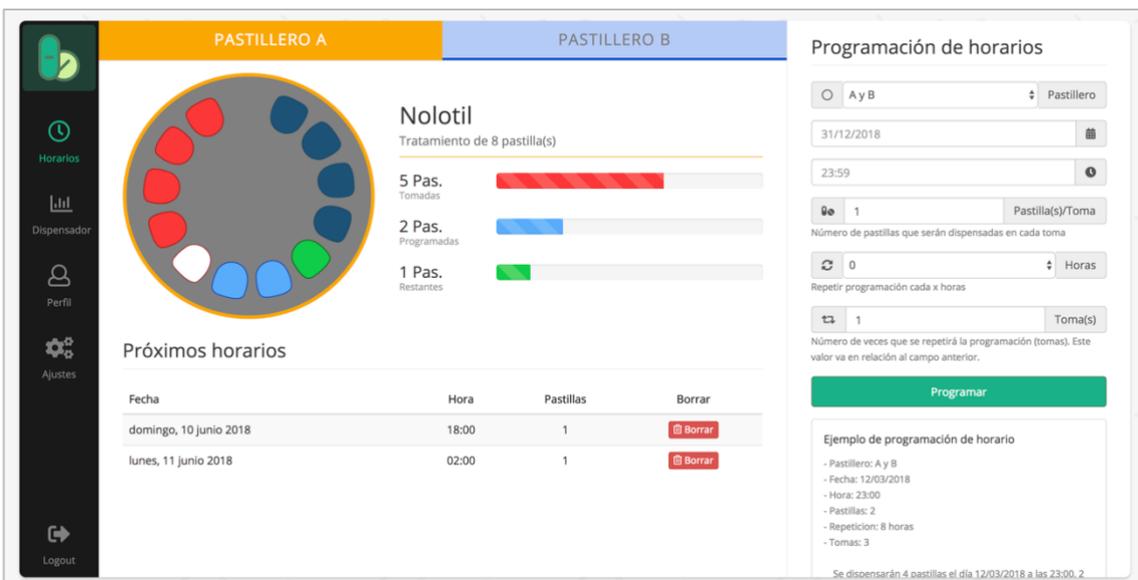


Figura 4.39. Diseño de la interfaz – Horarios

Cuando un usuario inicia sesión en la aplicación, la ventana de inicio es la mostrada en la imagen anterior. En ella puede verse un resumen del estado actual de cada una de las dos *ruletas* del dispensador (la parte superior permite cambiar del pastillero A al B), próximos horarios programados y un formulario para programar nuevos horarios de toma de medicación.

En el formulario de *programación de horarios* se debe seleccionar el pastillero o *ruleta* de programación, fecha y hora a la que deben dispensarse las pastillas, número de pastillas que serán dispensadas y opcionalmente, puede seleccionarse cada cuánto se debe repetir la programación y durante cuantas tomas. De esta manera se crearán tantos horarios como se haya introducido en el número de veces de repetición y que se repetirán según las horas marcadas en el dicho campo.

4.9.1.4 Interfaz de usuario – Dispensador

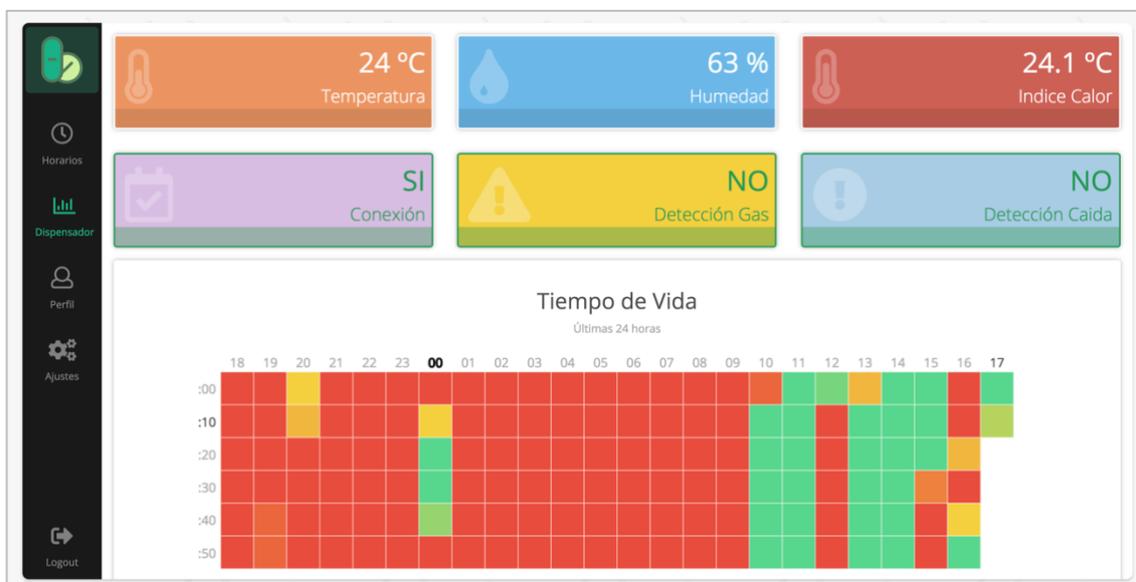


Figura 4.40. Diseño de la interfaz – Dispensador

En la venta de *Dispensador* se muestra un resumen y datos en tiempo real del dispensador y sus inmediateces. En la parte superior se encuentran valores actuales de temperatura, humedad, índice de calor, si el dispensador se encuentra conectado y si se ha detectado gas o caída del dispensador.

Tras esto, se muestra una visualización que recoge el *tiempo de vida* del dispensador, o lo que es lo mismo, peticiones del dispensador al servidor. Cada cuadro representa 10 minutos de las últimas 24 horas. En color rojo, el dispensador no realizó peticiones al servidor debido a que no disponía de conexión, mientras que en color verde se indica que el dispensador realizó el mayor número de peticiones posibles en esos 10 minutos.

El número máximo de peticiones del dispensador al servidor cada 10 minutos es de 10. Esto es así debido a que el dispensador ha sido programado para realizar una petición de horarios al servidor cada minuto.

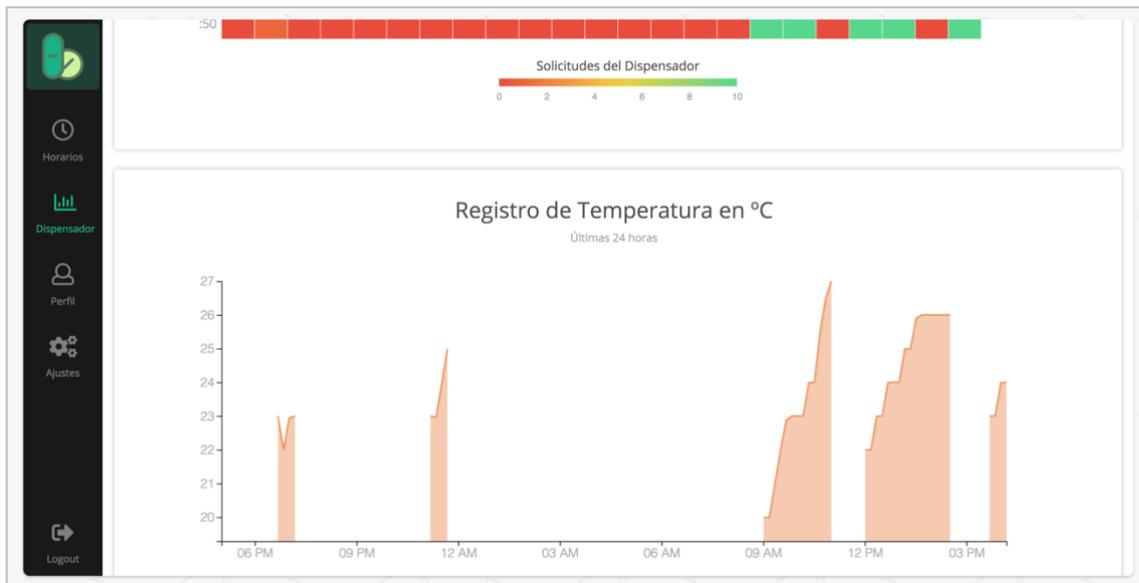


Figura 4.41. Diseño de la interfaz – Registro temperatura

Después de la visualización del *tiempo de vida*, existen otras dos que muestran la variación de la *temperatura* y de la *humedad* durante las últimas 24 horas. Las secciones sin datos representan que el dispensador no estuvo conectado durante esas horas.

Las secciones sin datos de las visualizaciones de la variación de la *temperatura* y *humedad* coinciden con las secciones en color rojo del tiempo de vida. Y, por el contrario, las secciones con datos de estas dos visualizaciones son las horas de las secciones en color verde del tiempo de vida.

Para la creación de estas visualizaciones del *tiempo de vida* del dispensador y de las variaciones de la *temperatura* y humedad de las últimas 24 horas, se ha utilizado la librería de desarrollo de infogramas [D3.js](https://d3js.org/).

4.9.1.5 Interfaz de usuario – Perfil

Figura 4.42. Diseño de la interfaz – Perfil

Los ajustes del *perfil* del usuario iniciado en sesión se encuentran disponibles tanto para el usuario cliente como para el administrador. En esta ventana pueden modificarse todos los datos utilizados en el registro salvo el serial del dispensador.

4.9.1.6 Interfaz de usuario – Ajustes

En la sección de *ajustes* de la aplicación se podrán configurar todos los ajustes relativos al tratamiento de los pastilleros, toma de medicación, dispensador, notificaciones y configuración de notificaciones.

4.9.1.6.1 Tratamiento

Figura 4.43. Diseño de la interfaz – Ajustes – Tratamiento

En los ajustes del *tratamiento* pueden configurarse para cada uno de los dispensadores o pastilleros el nombre del medicamento y el número de pastillas de las que se compone el tratamiento. El número de pastillas de una *ruleta* solo podrá ser modificado si no dispone de horarios programados para la misma y si el pastillero se encuentra en la posición de inicio, o lo que es lo mismo, no ha dispensado aún ninguna pastilla.

4.9.1.6.2 Toma de medicación

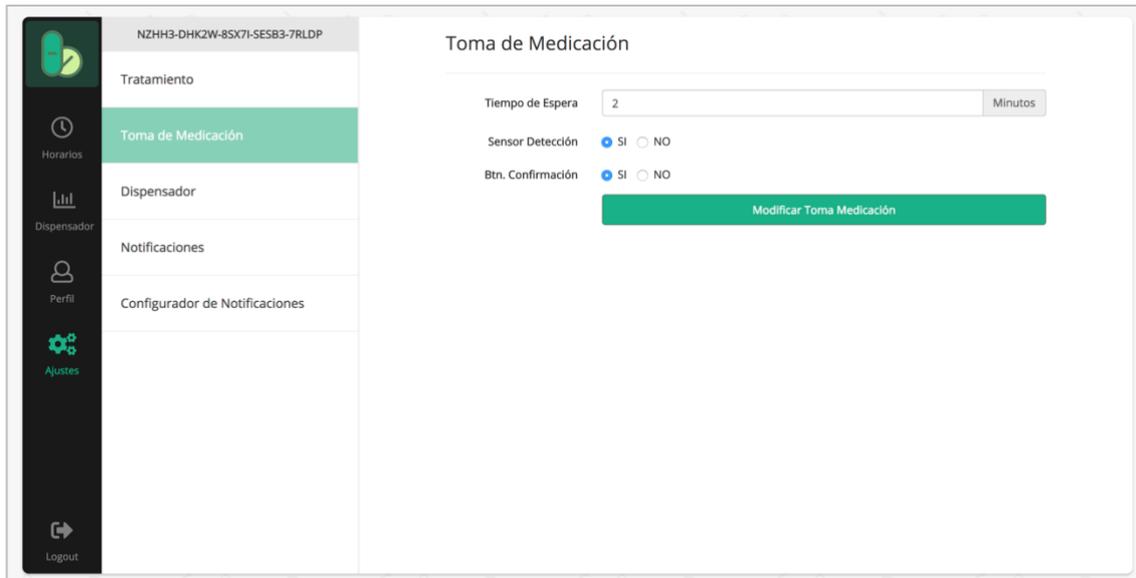


Figura 4.44. Diseño de la interfaz – Ajustes - Toma de medicación

En la *toma de medicación* puede configurarse el tiempo de espera en minutos para que la medicación sea tomada (después de ese tiempo, se considera que la medicación no ha sido tomada y se procede a notificar a los usuarios), y, si para considerar tomada la medicación, el sensor IR debe detectar la recogida de las pastillas y el botón verde de confirmación debe ser pulsado (son valores a través de los cuales se considerará tomada la medicación o no).

4.9.1.6.3 Dispensador

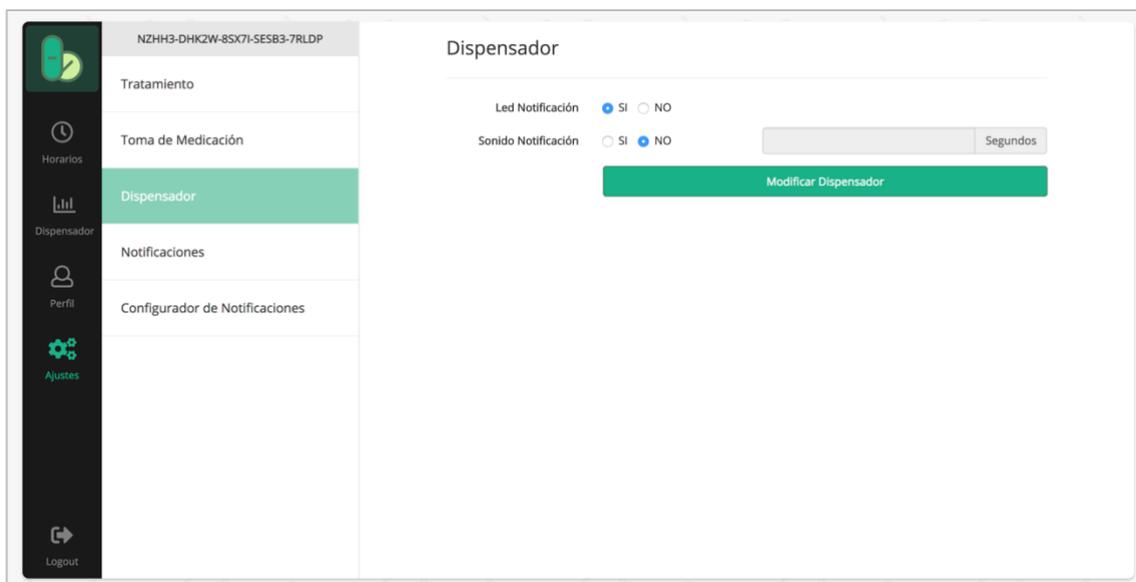


Figura 4.45. Diseño de la interfaz – Ajustes – Dispensador

En los ajustes del *dispensador* puede configurarse como se realizará la notificación de toma de medicación en el dispensador. Mediante el LED de notificación y/o emitiendo sonido. En el caso de selección la opción de sonido de notificación se puede especificar los segundos que permanecerá el dispensador emitiendo el sonido.

4.9.1.6.4 Notificaciones

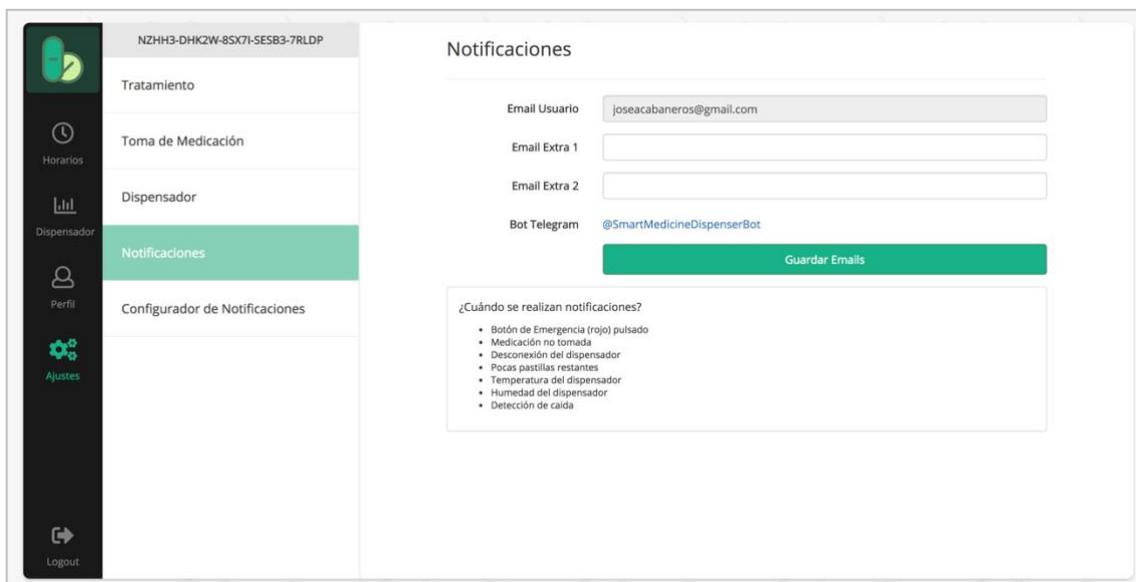


Figura 4.46. Diseño de la interfaz – Ajustes – Notificaciones

En los ajustes de *notificaciones* podrán añadirse hasta dos emails extra de notificación. En esta sección aparecerán los correos electrónicos de los usuarios registrados con dicho serial

y que serán utilizados para la notificación de alarmas. También se encuentra un enlace al bot de Telegram para suscribirse a las notificaciones del dispensador vía mensaje de Telegram.

4.9.1.6.5 Configurador de notificaciones

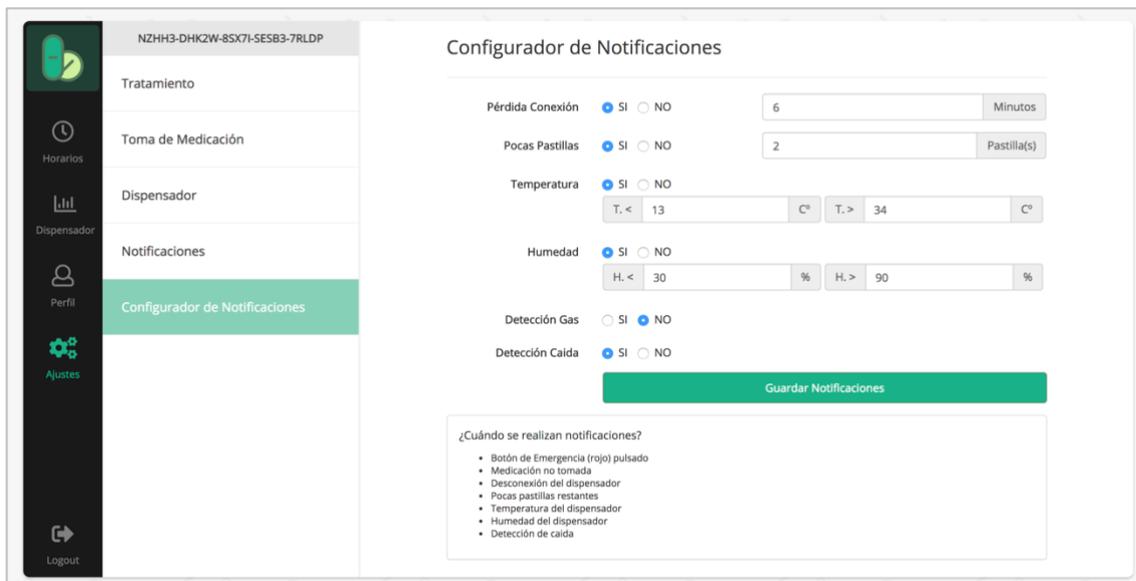


Figura 4.47. Diseño de la interfaz – Ajustes – Configurador de notificaciones

El *configurador de notificaciones* permite modelar las notificaciones que se realizarán a los correos electrónicos registrados y al bot de Telegram. Estas configuraciones son:

- **Pérdida de conexión:** permite seleccionar si se desea notificación cuando el dispensador pierda la conexión. En el caso de requerir notificación, puede configurarse el número de minutos que debe permanecer desconectado para enviar la notificación.
- **Pocas pastillas:** permite seleccionar si se desea notificación cuando a los pastilleros del dispensador les quede un número mínimo de pastillas en su interior. En el caso de requerir notificación, puede configurarse el número de pastillas restantes para enviar la notificación.
- **Temperatura:** permite seleccionar si se desea notificación cuando la temperatura ambiental del dispensador se encuentre por encima o por debajo de los márgenes establecidos en dicha configuración.
- **Humedad:** permite seleccionar si se desea notificación cuando la humedad ambiental del dispensador se encuentre por encima o por debajo de los márgenes fijados en dicha configuración.
- **Detección gas:** permite seleccionar si se desea notificación cuando el dispensador detecte gas en las inmediaciones.
- **Detección caída:** permite seleccionar si se desea notificación cuando el dispensador detecte una caída o golpe sobre el mismo.

Por último, en esta sección de la configuración se encuentra un panel de resumen que se actualiza según la configuración realizada y que informa de los casos en los que se realizarán notificaciones a los correos electrónicos proporcionados y al canal de Telegram.

4.9.1.7 Interfaz de usuario – Administración – Serials

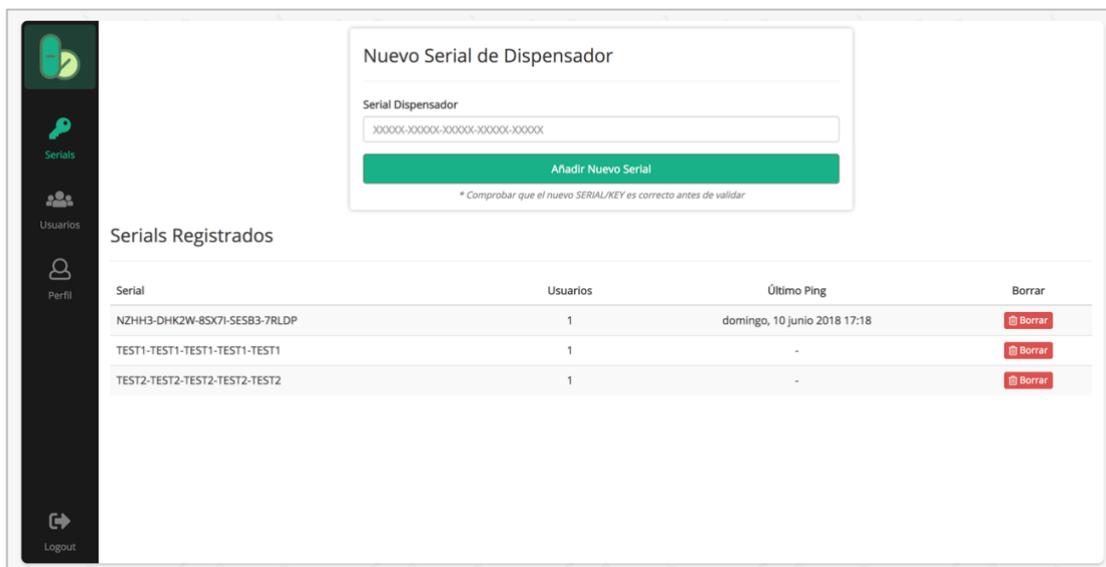


Figura 4.48. Diseño de la interfaz – Administración – Serials

En la sección *serials* de la parte de administración pueden registrarse nuevos seriales de dispensadores, consultar los seriales ya registrados y eliminarlos. La eliminación de seriales solo puede realizarse si no existen usuarios registrados con el serial que se pretende eliminar.

4.9.1.8 Interfaz de usuario – Administración – Usuarios

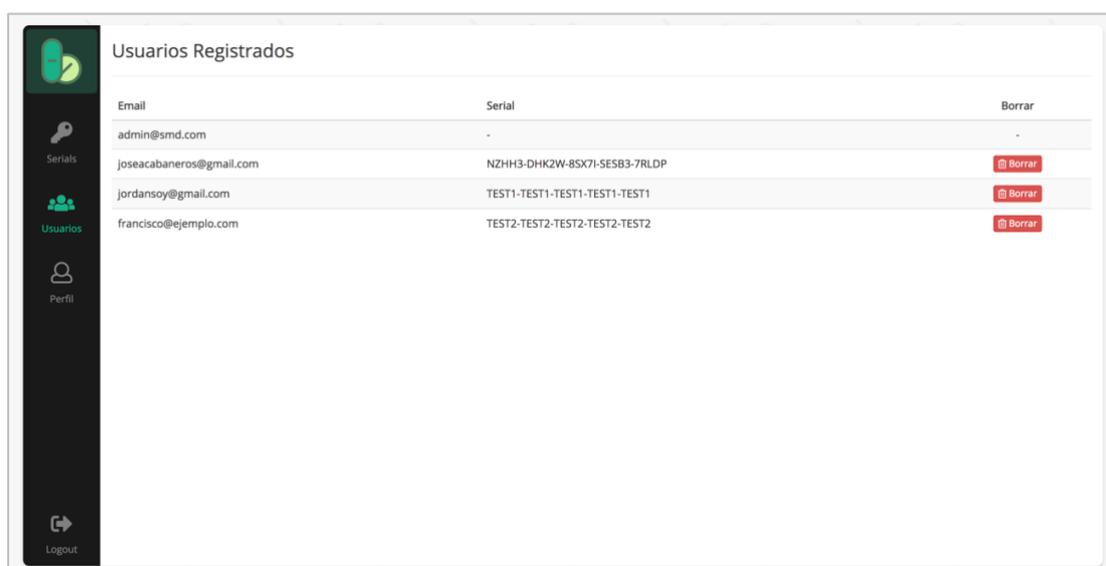


Figura 4.49. Diseño de la interfaz – Administración – Usuarios

Y, por último, la sección de *usuarios registrados* en el sistema del panel de administración permite consultar los usuarios y con qué número de serie fueron registrados. Además, el administrador también podrá eliminarlos.

4.9.2 Diseño responsive

La creación de una aplicación web capaz de adaptarse a la pantalla del dispositivo en la que se visualice es uno de los requisitos clave de este proyecto. El número de pantallas distintas donde se podría visualizar la aplicación es prácticamente ilimitado. Por ello, se persigue la creación de un diseño generalizado que sea capaz de adaptarse a cualquier dispositivo y sobre todo sea usable.



Figura 4.50. Diseño de la interfaz - Horarios - Smartphone vertical



Figura 4.51. Diseño de la interfaz - Dispensador - Smartphone vertical

En las imágenes anteriores se puede comprobar el aspecto que tendría la aplicación vista en modo vertical desde un *smartphone*. A la izquierda se encuentra la página de *horarios* que, a diferencia de la vista de escritorio, dispone de un botón flotante que permite abrir el formulario de programación de horarios. En la parte derecha se visualiza la página de

dispensador renderizando las visualizaciones de D3 y reorganizando las tarjetas que muestran datos en tiempo real.



Figura 4.52. Diseño de la interfaz - Horarios - Smartphone horizontal



Figura 4.53. Diseño de la interfaz - Dispensador - Smartphone horizontal

En ambas imágenes se puede observar el aspecto que presenta la aplicación desde un *smartphone* en modo horizontal. A diferencia del modo vertical, se puede comprobar cómo cambia la distribución de los componentes en ambas páginas de la aplicación.

4.10 Protocolo de Comunicación entre el Dispensador y el Servidor – API

La comunicación entre el microcontrolador del dispensador y el servidor se realiza por medio de peticiones desde el dispensador a la API del servidor. En algunas ocasiones, estas peticiones se realizan con el método GET y en otras con el método POST.

En esta sección se detallará cómo y cuándo el dispensador realiza peticiones a la API del servidor. También se describirá el proceso seguido para procesar cada una de las peticiones en el servidor hasta que se envía la respuesta de vuelta al microcontrolador.

4.10.1 Peticiones GET – Solicitud de horarios

El microcontrolador (por medio del módulo [ESP8266](#)) realiza peticiones GET a la API del servidor únicamente para solicitar horarios de programación. Estas peticiones son realizadas cada minuto por programación del microcontrolador. Además, las peticiones son aprovechadas para enviar parámetros ambientales y de control del dispensador por medio de la propia *query string*.

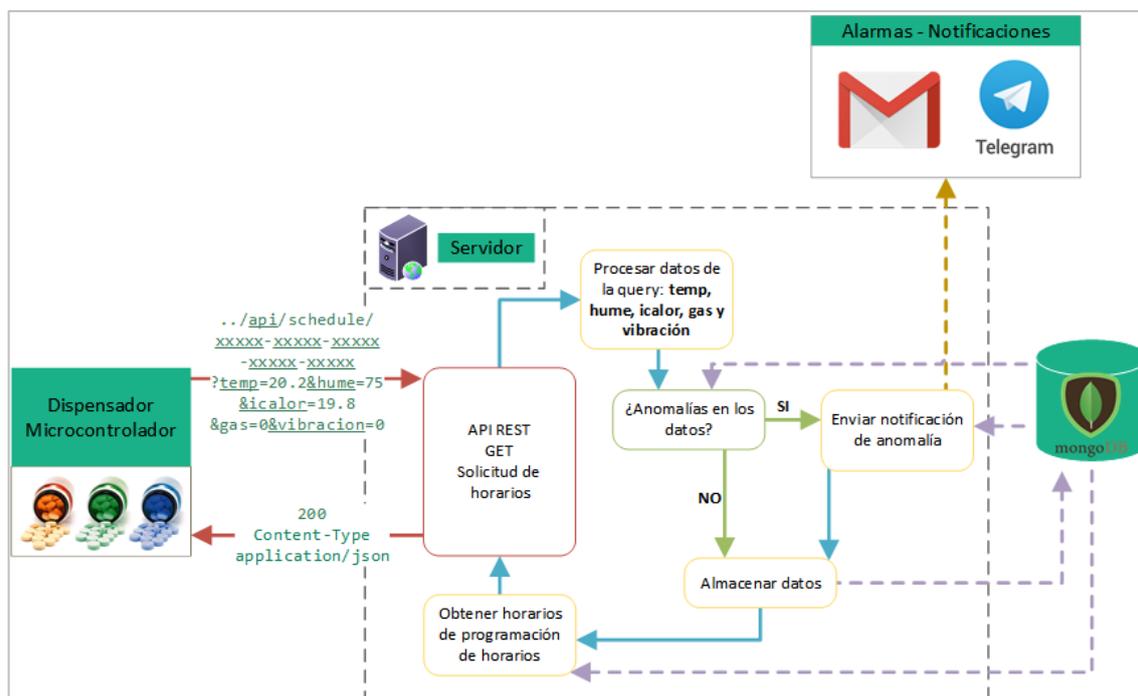


Figura 4.54. Protocolo de comunicación GET - Solicitud de horarios

En la imagen anterior se describe el flujo de ejecución realizado desde que el microcontrolador realiza la petición GET a la API hasta que recibe la respuesta en formato JSON con los horarios de medicación que requieren dispensar medicamentos en ese instante.

El proceso se inicia con la petición GET del dispensador a la API del servidor. Esta petición se realiza por medio de una URL del siguiente tipo:

```
https://localhost/api/schedule/XXXXX-XXXXX-XXXXX-XXXXX-XXXXX?temp=20.2&hume=75&icalor=19.8&gas=0&vibracion=0
```

En ella se especifica el serial del dispensador que solicita horarios de programación y al que pertenecen los datos de los parámetros de la *query*. En los parámetros de la *query* se envían los valores actuales de temperatura (en Celsius), de humedad (en %), el índice de calor (en Celsius) y detección de gas y vibración (1 se ha detectado, 0 no).

Entonces, el servidor recibe la petición y comienza procesando los parámetros de la *query*. Con los parámetros procesados y obteniendo las opciones de configuración del dispensador (establecidas por el usuario a través de la aplicación web en el apartado [configurador de notificaciones](#)) de la base de datos, el servidor busca anomalías en los datos recibidos. En el caso de detectar alguna anomalía, el servidor solicita a la base de datos los correos electrónicos y los IDs de Telegram a los que enviará una notificación de dicha anomalía. Tras esto, se almacenan los datos de los parámetros y se solicitan los horarios de programación de toma de medicamentos. Los horarios solicitados corresponden a la fecha y hora de la petición realizada. Finalmente, el servidor prepara la respuesta en formato JSON con los datos de los horarios que deben ser dispensados en ese instante de tiempo junto a la configuración de los respectivos horarios.

La respuesta por parte del servidor al dispensador contiene siempre un código 200 y un formato JSON. En el caso de que sea necesario dispensar medicamentos, ese JSON contendrá datos. Y en caso contrario, el JSON se enviará vacío.

Si se requiere dispensar pastillas en la fecha y hora de la petición del dispensador, el formato JSON de respuesta presenta el siguiente formato:

```
1. {
2.   "Numero horarios": 2,
3.   "Horarios programados": [
4.     {
5.       "id": "5af0781ade79fa08ac8293d9",
6.       "unixtometoma": 1525716060,
7.       "pastillero": "A",
8.       "pastillas": 1
9.     },
10.    {
11.      "id": "5af0781ade79fa08ac8293da",
12.      "unixtometoma": 1525716060,
13.      "pastillero": "B",
14.      "pastillas": 1
15.    }
16.  ],
17.  "Configuracion horarios": [
18.    {
19.      "tespera": "180",
20.      "ir": "true",
21.      "btn": "false",
22.      "led": "true",
23.      "zumbador": "false",
24.      "tzumbador": "null"
25.    }
26.  ]
27. }
```

Donde *número horarios* es 1 o 2 en función de si requiere dispensar medicamentos de un único pastillero o *ruleta*, o de ambos. *Horarios programados* contiene los datos de cada programación que debe ser dispensada. Cada entrada contiene el ID de la programación, el *UNIX time* de la toma (que corresponderá a la fecha y hora de la petición), pastillero o *ruleta* a la que afecta la programación y el número de pastillas que deben ser dispensadas.

Por último, se envían los parámetros de configuración establecidos por el usuario en la aplicación web en los apartados [toma de medicación](#) y [dispensador](#). Estos datos son: el tiempo de espera en segundos para que la medicación sea tomada, las acciones que debe realizar el usuario para que se considere tomada la medicación (detección IR y/o botón verde de confirmación) y el modo de notificación del dispensador al usuario que debe tomar la medicación (led y sonido de notificación, y en el caso de tener activada la opción de sonido, el número de segundo que permanecerá sonando).

4.10.2 Peticiones POST – Evento en el dispensador

El microcontrolador (por medio del módulo [ESP8266](#)) realiza peticiones POST a la API del servidor para:

- Actualizar acciones del usuario en el dispensador
- Actualizar posiciones de los pastilleros o *ruletas* tras dispensar medicamentos.
- Notificar que una medicación programada no fue tomada
- Notificar que el botón de emergencia fue pulsado.

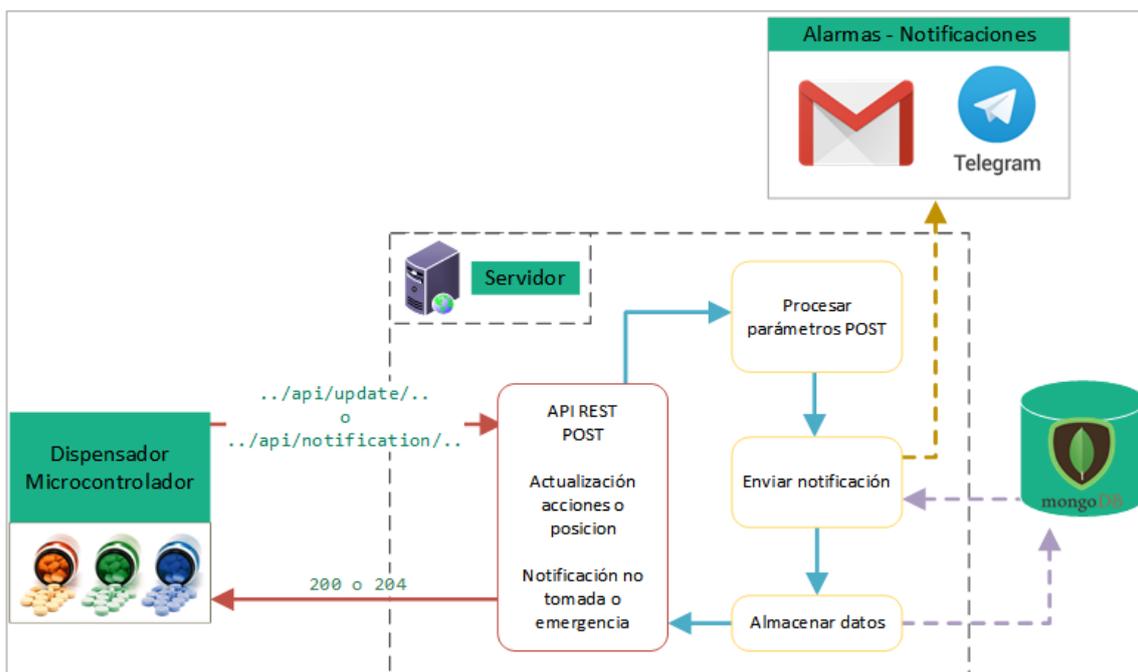


Figura 4.55. Protocolo de comunicación POST – Actualización y notificación

Estas peticiones son realizadas cuando surge en el dispensador alguno de los eventos anteriores. Cada evento tiene su URL en la API del servidor que procesa los datos recibidos

como cuerpo de la petición POST. Las URLs de la API del servidor y los datos que se envían en el cuerpo de la petición son del tipo:

- `https://localhost/api/update/action`
 - Datos: id horario y acción (tomadaBtn o tomadaIR)
- `https://localhost/api/update/position`
 - Datos: serial, pastillero y pastillas dispensadas
- `https://localhost/api/notification/notake`
 - Datos: id horario
- `https://localhost/api/notification/emergency`
 - Datos: serial

En la imagen anterior se muestra el flujo de ejecución seguido desde que el dispensador realiza una petición POST de las anteriores hasta que el servidor responde.

Las peticiones POST están orientadas a realizar algún cambio en la base de datos o como proceso de notificación. El proceso general seguido para procesar las peticiones POST a la API del servidor es el siguiente.

El dispensador realiza la petición POST a la API del servidor mediante alguna de las URLs vistas anteriormente y con los parámetros requeridos. El servidor procesa los parámetros recibidos en el cuerpo de la petición y, en el caso de que se trate de una petición de medicación no tomada, emergencia o pocas pastillas restantes, el servidor solicita los correos electrónicos e IDs de Telegram para enviar las notificaciones pertinentes. Luego, el servidor almacena los datos recibidos en la base de datos y envía la respuesta al dispensador con el código 200 si el procesamiento de la petición se realizó correctamente o 204 si hubo algún problema al procesar la petición.

4.11 Especificación Técnica del Plan de Pruebas

Como ya se especificó en el análisis de sistema, las pruebas que serán realizadas para comprobar el funcionamiento del sistema conectado y su correcta integración entre subsistemas. Asimismo, para garantizar que los usuarios de la aplicación web y del dispensador de medicamentos comprenden su funcionamiento y se familiarizan con su uso, se realizarán varias pruebas de usabilidad con cada uno de ellos.

4.11.1 Pruebas Unitarias y de Integración

A continuación se presentan las pruebas que serán realizadas junto al resultado esperado tras su ejecución. Las pruebas unitarias y de integración se realizarán sobre cada uno de los requisitos que componen la funcionalidad del sistema.

<i>Requisito Funcional RF-01 – Registrar usuarios en el sistema</i>	
Prueba	Resultado Esperado
Registrar usuario	El sistema posee un usuario más
Prueba	Resultado Esperado
Usar serial no registrado en el sistema	El sistema muestra un mensaje de error de serial no disponible
Prueba	Resultado Esperado
Registrar usuario ya existente	El sistema muestra un mensaje de error de email de usuario ya registrado
<i>Requisito Funcional RF-02 – Identificar usuarios cliente</i>	
Prueba	Resultado Esperado
Identificar usuario cliente	El sistema redirecciona al usuario al dashboard cliente de la aplicación
Prueba	Resultado Esperado
Identificar usuario no registrado	El sistema muestra un mensaje de error de credenciales de acceso inválidas
<i>Requisito Funcional RF-03 – Identificar usuarios administrador</i>	
Prueba	Resultado Esperado
Identificar usuario administrado	El sistema redirecciona al usuario al dashboard de administración de la aplicación
Prueba	Resultado Esperado
Identificar usuario administrador no registrado	El sistema muestra un mensaje de error de credenciales de acceso inválidas
<i>Requisito Funcional RF-04 – Consultar estado del dispensador</i>	
Prueba	Resultado Esperado
Consultar estado del pastillero A del dispensador	El usuario comprueba el estado de las pastillas contenidas en el pastillero A del dispensador de forma visual y numérica
Prueba	Resultado Esperado
Consultar estado del pastillero B del dispensador	El usuario comprueba el estado de las pastillas contenidas en el pastillero B del dispensador de forma visual y numérica
<i>Requisito Funcional RF-05 – Programar días y horas a las que deben expulsarse las pastillas</i>	
Prueba	Resultado Esperado
Programar día y hora de toma de medicación	El sistema registro un nuevo horario de programación y se muestra al usuario en próximos horarios programados

Prueba	Resultado Esperado
Programar día y hora de toma de medicación que se solape con otros horarios	El sistema muestra un mensaje en la operación y el horario con el que se solapa.
Prueba	Resultado Esperado
Programar múltiples horarios de toma de medicación	El sistema registra los nuevos horarios y los muestra al usuario en próximos horarios programados.
<u>Requisito Funcional RF-06 – Consultar horarios programados</u>	
Prueba	Resultado Esperado
Consultar próximos horarios programados	El usuario consulta a través de la aplicación la fecha, hora y número de pastillas para los próximos horarios programados
<u>Requisito Funcional RF-07 – Eliminar horarios programados</u>	
Prueba	Resultado Esperado
Eliminar horarios programados	El sistema elimina el horario programado y actualiza la pantalla con los nuevos próximos horarios
<u>Requisito Funcional RF-08 – Mostrar gráficamente el tiempo de vida del dispensador</u>	
Prueba	Resultado Esperado
Mostrar gráficamente el tiempo de vida del dispensador	El usuario visualiza la distribución horaria de las últimas 24 horas del número de conexiones del dispensador con el servidor.
<u>Requisito Funcional RF-09 – Mostrar gráficamente el registro de temperaturas y humedades</u>	
Prueba	Resultado Esperado
Mostrar gráficamente el registro de temperaturas	El usuario visualiza la distribución horaria de las últimas 24 horas del registro de temperaturas del dispensador
Prueba	Resultado Esperado
Mostrar gráficamente el registro de humedad	El usuario visualiza la distribución horaria de las últimas 24 horas del registro de humedad del dispensador
<u>Requisito Funcional RF-10 – Mostrar datos en tiempo real del dispensador</u>	
Prueba	Resultado Esperado
Mostrar datos en tiempo real del dispensador	El usuario visualiza en tiempo real la temperatura, humedad, índice de calor, estado de conexión y estado de detección de gas y golpe del dispensador
<u>Requisito Funcional RF-11 – Modificar datos personales de registro</u>	
Prueba	Resultado Esperado
Modificar datos personales del usuario	El sistema actualiza los datos personales del usuario
Prueba	Resultado Esperado
Modificar email de usuario por uno ya registrado	El sistema muestra un mensaje de error de email ya registrado en el sistema
<u>Requisito Funcional RF-12 – Especificar que pastillas han sido introducidas en el dispensador</u>	
Prueba	Resultado Esperado
Modificar que pastillas han sido introducidas en el dispensador	El sistema registra el nombre y número de pastillas del tratamiento del pastillero A y B.
Prueba	Resultado Esperado
Modificar tratamiento de un pastillero ya iniciado	El sistema muestra un mensaje de error de imposibilidad en el cambio de tratamiento de un pastillero si dispone de horarios programados y/o no se encuentra en la posición de inicio.
<u>Requisito Funcional RF-13 – Configurar tiempo de espera que determina que no se ha tomado la medicación</u>	

Prueba	Resultado Esperado
Configurar tiempo de espera de toma de medicación	El sistema modifica el tiempo de espera requerido para tomar la medicación del dispensador.
<u>Requisito Funcional RF-14 – Configurar el modo en el que se considera tomada la medicación</u>	
Prueba	Resultado Esperado
Activar y desactivar la opción de detección de toma	El sistema modifica la configuración del dispensador con la detección o no detección de la retirada de las pastillas.
<u>Requisito Funcional RF-15 – Configurar la emisión de notificaciones de toma de medicación en el dispensador</u>	
Prueba	Resultado Esperado
Activar y desactivar la opción de botón de confirmación	El sistema modifica la configuración del dispensador con el requerimiento o no de confirmar la toma de la medicación
<u>Requisito Funcional RF-16 – Añadir o eliminar correos electrónicos de notificación de alarmas</u>	
Prueba	Resultado Esperado
Añadir un nuevo email extra de notificación	El sistema registra el nuevo email de notificación para el dispensador.
Prueba	Resultado Esperado
Eliminar email extra de notificación	El sistema elimina el email de notificación para el dispensador.
<u>Requisito Funcional RF-17 – Suscribirse al canal de Telegram de notificaciones</u>	
Prueba	Resultado Esperado
Suscribirse al canal de Telegram	El sistema almacena el ID de Telegram del usuario suscrito a las notificaciones del dispensador
<u>Requisito Funcional RF-18 – Configurar notificaciones de alarmas a los emails y canal de Telegram</u>	
<u>Requisito Funcional RF-18.1 – Activar o desactivar notificación de pérdida de conexión</u>	
Prueba	Resultado Esperado
Activar y desactivar la notificación de pérdida de conexión	El sistema modifica la configuración de notificaciones del dispensador para enviarlas cuando se pierde la conexión (activada) o no (desactivada)
Prueba	Resultado Esperado
Modificar el número de minutos	El sistema modifica la configuración de las notificaciones del dispensador para enviarlas cuando se pierda la conexión del dispensador durante los minutos especificados
<u>Requisito Funcional RF-18.2 – Activar o desactivar notificación de pocas pastillas restantes</u>	
Prueba	Resultado Esperado
Activar y desactivar la notificación de pocas pastillas restantes	El sistema modifica la configuración de notificaciones del dispensador para enviarlas cuando queden pocas pastillas en uno de los pastilleros (activada) o no (desactivada)
Prueba	Resultado Esperado
Modificar el número de pastillas	El sistema modifica la configuración de las notificaciones del dispensador para enviarlas cuando queden menos pastillas de las especificadas
<u>Requisito Funcional RF-18.3 – Activar o desactivar notificación de anomalía en la temperatura</u>	
Prueba	Resultado Esperado
Activar y desactivar la notificación de anomalía en la temperatura	El sistema modifica la configuración de notificaciones del dispensador para enviarlas cuando se detecte una anomalía en la temperatura(activada) o no (desactivada)

Prueba	Resultado Esperado
Modificar la temperatura para que se considere anomalía	El sistema modifica la configuración de las notificaciones del dispensador para enviarlas cuando la temperatura sea inferior o superior a los valores de anomalía establecidos
<i>Requisito Funcional RF-18.4 – Activar o desactivar notificación de anomalía en la humedad</i>	
Prueba	Resultado Esperado
Activar y desactivar la notificación de anomalía en la humedad	El sistema modifica la configuración de notificaciones del dispensador para enviarlas cuando se detecte una anomalía en la humedad(activada) o no (desactivada)
Prueba	Resultado Esperado
Modificar la humedad para que se considere anomalía	El sistema modifica la configuración de las notificaciones del dispensador para enviarlas cuando la humedad sea inferior o superior a los valores de anomalía establecidos
<i>Requisito Funcional RF-18.5 – Activar o desactivar notificación de detección de gas</i>	
Prueba	Resultado Esperado
Activar y desactivar la notificación de detección de gas	El sistema modifica la configuración de notificaciones del dispensador para enviarlas cuando se detecte gas(activada) o no (desactivada)
<i>Requisito Funcional RF-18.6 – Activar o desactivar notificación de golpe del dispensador</i>	
Prueba	Resultado Esperado
Activar y desactivar la notificación de detección de golpe en el dispensador	El sistema modifica la configuración de notificaciones del dispensador para enviarlas cuando se detecte un golpe(activada) o no (desactivada)
<i>Requisito Funcional RF-19 – Añadir nuevos seriales al sistema</i>	
Prueba	Resultado Esperado
Añadir un nuevo serial al sistema	El sistema registra un nuevo serial de dispensador
Prueba	Resultado Esperado
Añadir un serial ya registrado	El sistema muestra un mensaje de error de serial ya registrado
<i>Requisito Funcional RF-20 – Consultar seriales registrados en el sistema</i>	
Prueba	Resultado Esperado
Consultar seriales registrados	El usuario visualiza los seriales registrados, número de usuario registrados con cada serial y la fecha y hora de la última conexión del dispensador
<i>Requisito Funcional RF-21 – Eliminar seriales del sistema</i>	
Prueba	Resultado Esperado
Eliminar serial registrado	El sistema elimina el serial
Prueba	Resultado Esperado
Eliminar serial con usuario registrados	El sistema muestra un mensaje de error en la eliminación del serial por disponer de usuarios registrados
<i>Requisito Funcional RF-22 – Consultar usuarios registrado en el sistema</i>	
Prueba	Resultado Esperado
Consultar usuario registrados	El usuario visualiza el email de los usuarios registrados y el serial de su dispensador
<i>Requisito Funcional RF-23 – Eliminar usuarios del sistema</i>	
Prueba	Resultado Esperado
Eliminar usuario registrado	El sistema elimina el usuario
<i>Requisito Funcional RF-24 – Recoger y tomar medicación</i>	
Prueba	Resultado Esperado
Recoger medicación del	El sistema almacena el evento de recogida de las pastillas

dispensador	dispensadas de un horario de programación
<u>Requisito Funcional RF-25 – Confirmar medicación tomada</u>	
Prueba	Resultado Esperado
Confirmar la toma de la medicación	El sistema almacena el evento de confirmación de toma de las pastillas dispensadas de un horario de programación
<u>Requisito Funcional RF-26 – Emitir notificación de emergencia</u>	
Prueba	Resultado Esperado
Emitir notificación de emergencia	El sistema envía notificaciones a los emails del dispensador y canal de Telegram de situación de emergencia
<u>Requisito Funcional RF-27 – Expulsar pastillas</u>	
Prueba	Resultado Esperado
Dispensar pastillas	El dispensador expulsa pastillas y el sistema almacena el nuevo estado de los pastilleros tras expulsarlas
<u>Requisito Funcional RF-28 – Registrar estado de dispensador</u>	
Prueba	Resultado Esperado
Registrar nuevo estado de los pastilleros del dispensador	El sistema almacena el nuevo estado de los pastilleros del dispensador una vez a expulsado las pastillas de un horario de programación
<u>Requisito Funcional RF-29 – Registrar que el dispensador está conectado a internet</u>	
Prueba	Resultado Esperado
Registrar conexión del dispensador	El sistema registra el estado de conexión del dispensador.
<u>Requisito Funcional RF-30 – Almacenar datos ambientales de temperatura y humedad</u>	
Prueba	Resultado Esperado
Almacenar datos ambientales de temperatura y humedad	El sistema registra datos ambientales del dispensador cada minuto
<u>Requisito Funcional RF-31 – Almacenar detección de gas o golpe del dispensador</u>	
Prueba	Resultado Esperado
Almacenar estado de detección de gas o golpe	El sistema registra el estado de detección de gas y caída del dispensador cada minuto
<u>Requisito Funcional RF-32 – Enviar notificación por email y canal de Telegram cuando no se tome la medicación</u>	
Prueba	Resultado Esperado
Enviar notificaciones a los emails y canal de Telegram de medicación no tomada	El sistema envía email a los usuarios del dispensador y canal de Telegram de cuando la medicación de un horario de programación no haya sido tomada.
<u>Requisito Funcional RF-33 – Enviar notificación por email y canal de Telegram cuando se produzca una anomalía en el dispensador</u>	
Prueba	Resultado Esperado
Enviar notificaciones a los emails y canal de Telegram de anomalía en el dispensador	El sistema envía email a los usuarios del dispensador y canal de Telegram de cuando se detecte una anomalía en los datos del dispensador
<u>Requisito Funcional RF-34 – Enviar notificación por email y canal de Telegram cuando se pulse el botón de emergencia del dispensador</u>	
Prueba	Resultado Esperado
Enviar notificaciones a los emails y canal de Telegram de emergencia	El sistema envía email a los usuarios del dispensador y canal de Telegram de cuando se pulse el botón de emergencia del dispensador
<u>Requisito Funcional RF-35 – Gestionar horarios de toma de medicación</u>	
Prueba	Resultado Esperado

Gestionar horarios de toma de medicación	El dispensador controla el tiempo establecido para tomar la medicación dispensada
<i>Requisito Funcional RF-36 – Emitir notificaciones sonoras y luminosas</i>	
Prueba	Resultado Esperado
Emitir notificaciones sonoras y luminosas	El dispensador emite notificaciones sonoras y/o luminosas cuando se requieran tomar pastillas dispensadas por un horario de toma programado
<i>Requisito Funcional RF-37 – Configurar dispensador</i>	
Prueba	Resultado Esperado
Configurar dispensador	El dispensador se configura con los datos de los ajustes que el usuario ha realizado en la aplicación web

4.11.2 Pruebas de Usabilidad

Las pruebas de usabilidad son fundamentales para corregir y mejorar ciertos aspectos del sistema conectado. Serán llevadas a cabo mediante la observación de cómo interactúan los usuarios con la aplicación y el dispensador y cómo de fácil o difícil les resulta usarlos.

4.11.2.1 Pruebas de usabilidad sobre la Aplicación Web

Para las pruebas de usabilidad de la aplicación web se usarán tres tipos distintos de perfiles de usuarios. Usuarios de mediana edad sin demasiado conocimientos informáticos ni experiencia en Internet, usuario de mediana edad con experiencia en informática y usuarios jóvenes con habilidad en el uso de aplicaciones móviles.

A todos estos usuarios se les ordenará realizar una serie de pasos de navegación por la aplicación con los que observar su forma de interacción con la misma y se les formularán preguntas a lo largo de la navegación. La rutina de navegación y preguntas será:

- 1) Registrarse en el sistema
- 2) Iniciar sesión con el usuario registrado
- 3) ¿Comprende lo que le muestra la pantalla de inicio?
- 4) Programar un único horario de toma de medicación
- 5) Programar múltiples horarios de toma de medicación
- 6) ¿Le ha resultado sencillo programar múltiples horarios?
- 7) Navegar a la pantalla del dispensador
- 8) ¿Comprende los datos y las visualizaciones que se le muestran?
- 9) Navegar a la pantalla de ajustes
- 10) Configurar tratamiento del pastillero B
- 11) ¿Conoce cuál es el objetivo de modificar el tratamiento de un pastillero?
- 12) Añadir un nuevo email de notificación
- 13) Configurar las notificaciones para que se envíen cuando queden menos de 3 pastillas
- 14) ¿Comprende todas las opciones del configurador de notificaciones?
- 15) Iniciar sesión en la aplicación con un smartphone
- 16) Programar un horario de medicación
- 17) ¿Le resulta sencillo el uso de la versión móvil?

18) ¿Añadiría o cambiaría algo en la aplicación?

4.11.2.2 Pruebas de usabilidad sobre el Dispensador de Medicamentos

Para las pruebas de usabilidad de dispensador de medicamentos se contará con dos tipos distintos de perfiles de usuarios: usuarios de mediana edad sin conocimientos tecnológicos y usuarios de avanzada edad.

A todos estos usuarios se les ordenará realizar una serie de pasos en el dispensador con los que observar su forma de interacción con la misma y se les formularán preguntas a lo largo de los mismos. Los pasos y preguntas que serán realizados son:

- 1) Recoger pastillas dispensadas
- 2) ¿El dispensador le ha notificado clara y correctamente la toma de la medicación?
- 3) Confirmar toma de medicación
- 4) Emitir situación de emergencia
- 5) ¿Recuerda los pasos que hemos seguidos para tomar la medicación?
- 6) ¿Cambiaría algo del dispensador?

Capítulo 5. Implementación del Sistema

5.1 Estándares y Normas Seguidos

Para el desarrollo de la aplicación web, se han seguido los estándares comunes de cualquier aplicación; HTML5 y CSS3. Siguiendo estos estándares, se consigue un enriquecimiento del marcado HTML y CSS. Esto permite escalar los contenidos de la web de manera más sencilla y etiquetar los contenidos con significado semántico.

El sistema distribuido ha sido desarrollado en Node.js y por ello, toda la implementación referente a la capa del servidor ha sido desarrollada en JavaScript. Para el desarrollo del código JavaScript, se ha seguido la especificación de desarrollo ECMAScript 5 que permite definir tipos dinámicos y soporta algunas características de la programación orientada a objetos. Además, permite realizar acciones comunes en el lenguaje JavaScript de forma mucho más sencilla.

Para la comunicación entre el microcontrolador del dispensador y el servidor Node.js, se ha utilizado el estilo de arquitectura de software para sistemas REST. Esta arquitectura permite intercambiar información a través de internet haciendo uso del protocolo HTTP. Las respuestas a las peticiones del microcontrolador del dispensador son enviadas en formato JSON con los datos de la respuesta. JSON es un formato de texto ligero para el intercambio de datos, perfecto para su uso en la transferencia de datos a través de Internet.

En cuanto a las normas seguidas para el desarrollo; el código del microcontrolador ha sido organizado en dos ficheros independientes, uno para la programación del módulo WiFi del controlador y otro para el módulo encargado de las comunicaciones con los componentes electrónicos, motores y sensores.

La aplicación web ha sido estructurada en módulos que abarcan un conjunto de funcionalidades concretas. Así, los módulos definidos son: API de conexión e intercambio de datos con el microcontrolador, conexión, acceso y comunicación con la base de datos, funcionalidades por rutas de la aplicación web, módulo de utilidades, y carpeta contenedora de todas las páginas y plantillas HTML de la aplicación web.

Por último, en lo referente al desarrollo de código tanto del microcontrolador como del sistema distribuido se ha procurado externalizar todas aquellas funcionalidades que requerían de su uso en diferentes partes del sistema con el objetivo de no repetir código. También, se han seguido todas aquellas prácticas de *buen código* para el desarrollo de software, refactorizando sistemáticamente a lo largo de todo el desarrollo y se han comentado todas aquellas funcionalidades, componentes y elementos importantes o con una complejidad no trivial.

5.2 Lenguajes de Programación

Los lenguajes de programación utilizados para la implementación del sistema conectado han sido dos:

- **JavaScript** junto con el entorno **Node.js**

Node.js es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor basado en el lenguaje de programación ECMAScript, asíncrono, con I/O de datos en una arquitectura orientada a eventos y basado en el motor V8 de Google.

Fue creado con el enfoque de ser útil en la creación de programas de red altamente escalables, como servidores web. Fue creado por Ryan Dahl en 2009 y su evolución está apadrinada por la empresa Joyent.

Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas, aunque existe una forma de JavaScript del lado del servidor (server-side JavaScript o SSJS).

El entorno de ejecución Node.js junto al lenguajes JavaScript han sido utilizado para desarrollar toda la lógica capa del servidor del sistema. Además, el lenguaje JavaScript también ha sido utilizado en el lado del cliente de la aplicación para ciertos componentes dinámicos y mejora en la experiencia de usuario de la aplicación.

- **C++**

Es un lenguaje de programación diseñado a mediados de los años 80. La intención de su creación fue extender el lenguaje de programación C con mecanismos que permiten la manipulación de objetos. Por ello, C++ es un lenguaje híbrido además de un lenguaje de programación multiparadigma.

C++ es el lenguaje usado para la implementación de toda la lógica del microcontrolador, tanto del módulo encargado de los componentes físicos como del módulo WiFi.

5.3 Herramientas y Programas Usados para el Desarrollo

En esta sección se describen todos los programas y herramientas utilizados para el desarrollo del sistema conectado. También se incluyen los módulos o componentes externos requeridos en el desarrollo.

5.3.1 Eclipse junto al plugin NodeEclipse

El entorno de desarrollo *Eclipse* ha sido utilizado para el desarrollo de la capa del servidor. Para ello, se ha utilizado el plugin *NodeEclipse* que permite extender la funcionalidad de entorno para el desarrollo de proyectos Node.js.



Figura 5.1. Logo Eclipse Oxygen

La versión de *Eclipse* utilizada ha sido Oxygen 4.7.1 y la versión del plugin *NodeEclipse* la 2.0.

5.3.2 Arduino Software IDE

El entorno de desarrollo *Arduino Software IDE* ha sido utilizado para el desarrollo de los módulos del microcontrolador. Para la implementación del módulo WiFi del microcontrolador se ha utilizado la librería externa *SoftwareSerial* que permite abstraer la programación de dicho módulo para su desarrollo.



Figura 5.2. Logo Arduino

La versión utilizada del entorno *Arduino Software IDE* ha sido la 1.8.5 y la versión de la librería *SoftwareSerial* la 1.0.0.

5.3.3 Control de versiones Git

El software de control de versiones *Git* ha sido utilizado durante todo el desarrollo del proyecto para el mantenimiento de versiones de desarrollo y control de cambios tanto de la implementación del lado del servidor como del microcontrolador.



Figura 5.3. Logo Git

5.3.4 GitHub

GitHub ha sido la plataforma de desarrollo colaborativo utilizada para almacenar todas las versiones de desarrollo y cambios del sistema conectado. Ha sido utilizado junto a la herramienta *Git* para hacer efectivos los cambios.

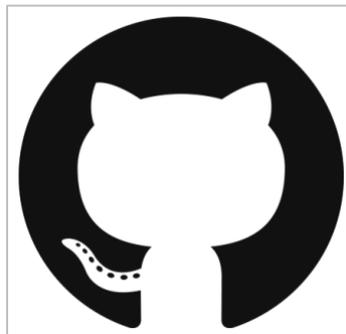


Figura 5.4. Logo GitHub

5.3.5 MongoDB en local

La base de datos *MongoDB* ha sido instalada en local para testear todas las funcionalidades desarrolladas en el sistema. La versión de *MongoDB* utilizada para el desarrollo del proyecto ha sido la 3.6.



Figura 5.5. Logo MongoDB

5.3.6 RoboMongo

RoboMongo es una interfaz gráfica que permite gestionar instancias de la base de datos MongoDB de manera gráfica y sencilla. Permite conectarse tanto a base de datos mongo desplegadas en local como en la nube.



Figura 5.6. Logo RoboMongo

Este software ha sido utilizado durante todo el desarrollo del proyecto para gestionar los datos almacenados en la instancia de MongoDB utilizada. La versión utilizada ha sido la 1.2.1.

5.3.7 Postman

Es un programa que permite realizar llamadas a APIs desplegadas en un servidor externo. Permite configurar parámetros en la llamada y visualizar los datos de la respuesta.



Figura 5.7. Logo Postman

Postman ha sido utilizado para ejecutar pruebas sobre la API REST desarrollada para la comunicación entre el microcontrolador del dispensador y el servidor.

5.3.8 Heroku

Es la plataforma de servicio de computación en la nube utilizada para desplegar la parte del servidor del sistema en Internet. Permite desplegar aplicaciones en Node.js en Internet de forma sencilla, definiendo para ello una URL donde se mantendrá desplegada la aplicación.



Figura 5.8. Logo Heroku

Para subir la aplicación a Heroku, se ha utilizado la herramienta de control de versiones *Git*.

5.3.9 mLab

mLab es una plataforma que permite crear instancias de la base de datos *MongoDB* en la nube. Esta plataforma ha sido utilizada para migrar la base de datos MongoDB de local a la nube y de esta manera poder hacerla accesible desde la aplicación desplegada en *Heroku*.



Figura 5.9. Logo mLab

5.3.10 Tinkercad

Es un editor de componentes 3D online que permite diseñar piezas para ser impresas en 3D de forma sencilla. *Tinkercad* ha sido utilizado para diseñar todos los componentes del dispensador para su posterior impresión.



Figura 5.10. Logo Tinkercad

5.3.11 Cura

Cura es un programa que permite importar piezas diseñadas en 3D y configurar los parámetros de estas para su impresión. Convierte los archivos de diseño de las piezas en ficheros con un formato interpretable por las impresoras 3D.



Figura 5.11. Logo Cura

Este programa ha sido utilizado para configurar y generar el archivo de impresión de todas las piezas que componen el dispensador de medicamentos.

5.4 Creación del Sistema

Esta sección se comentarán los problemas más relevantes surgidos durante el desarrollo del proyecto y se describirán detalles de implementación, componentes y otros aspectos relevantes del sistema distribuido, la aplicación web y el dispensador de medicamentos.

5.4.1 Dificultades Superadas

A continuación, se comentan algunos de los problemas más destacados ocasionados durante el desarrollo del proyecto. En estos problemas se incluye cualquier obstáculo superado durante la implementación de los subsistemas que componen el sistema conectado.

5.4.1.1 *Modo de funcionamiento del microcontrolador*

El microcontrolador que finalmente se eligió para el desarrollo del proyecto está formado por dos módulos que deben programarse obligatoriamente por separado. Además, se trata de un microcontrolador de reciente aparición y con prácticamente nula documentación y la poca que existe no se encuentra en inglés.

Aun así, se decidió usar este microcontrolador, ya que era el que mejor se ajustaba a las necesidades del sistema y pensando que la programación no distaría mucho de un Arduino. Cuando empezaron las pruebas con el microcontrolador, se comprobó que era más difícil de programar de que parecía en un principio. Disponía de un *switch* físico con hasta ocho configuraciones diferentes para su programación.

Tras muchas pruebas, se consiguió entender el funcionamiento por módulos del microcontrolador. También ayudó encontrar un tutorial detallado de este microcontrolador en ruso, que fue necesario traducir.

5.4.1.2 *Ampliación y aprendizaje de tecnologías de programación*

Para el desarrollo de la arquitectura del microcontrolador y especialmente, aspectos de comunicación a través de Internet, fue necesario ampliar los conocimientos sobre la programación del microcontrolador. También fue necesario aprender a utilizar la librería REST incluida en la programación del controlador para lanzar peticiones a la API del servidor y procesar su respuesta en formato JSON (por medio de otra librería).

Para la creación de infogramas y visualizaciones de la aplicación web con la librería D3.js, fue necesario el aprendizaje previo del uso de esta librería. D3.js requiere de unos conocimientos mínimos de funcionamiento que deben ser aprendidos antes de desarrollar cualquier visualización con esta librería. Por ello, fue necesario el aprendizaje previo de su uso y una ampliación de los conocimientos para la creación específica de los gráficos y visualizaciones contenidos en la aplicación.

5.4.1.3 Gestión de fechas y horarios

Dado que una de las funcionalidades principales de este sistema es la programación de horarios, resulta indispensable establecer una norma en el formato de fechas. Estos horarios deberían ser transmitidos al microcontrolador del dispensador para que controlara la toma de la medición según al tiempo de toma fijado por el usuario en la aplicación web.

Se decidió almacenar las fechas en formato *UNIX time* que facilitaría la comunicación y gestión de horario entre el dispensador y el servidor. Durante parte del desarrollo, se obtenía la hora en formato UTC y se sumaba una hora según la zona horaria de España.

El problema surgió cuando se produjo el cambio de horario y la parte del sistema desarrollada hasta el momento dejó de funcionar. Finalmente, se decidió hacer los cambios necesarios tanto en la implementación del microcontrolador como el sistema distribuido para que fueran capaces, mediante servicios NTP disponibles en Internet, de autoconfigurar la zona horaria de forma transparente.

De esta manera podría utilizarse el sistema conectado en cualquier fecha del año sin requerir ningún cambio en la implementación del mismo.

5.4.1.4 Diseño de los pastilleros del dispensador

El diseño de piezas en 3D puede realizarse muy meticulosamente y estableciendo medidas exactas. Pero la realidad es que a la hora de imprimirlas esa exactitud no suele transformarse en realidad, produciendo piezas con variaciones, en los mejores casos, de 0.1mm. Utilizar gamas superiores de impresoras 3D podrían paliar estos defectos.

El motor paso a paso elegido para dispensar pastillas tiene una forma muy peculiar y para nada regular. Esto dificultó el diseño de los pastilleros y su posterior incorporación, siendo finalmente necesario adaptar a mano cada uno de los pastilleros una vez impresos para ajustar las dimensiones del mismo y que el motor pudiera girar libremente el selector para dispensar pastillas de forma controlada.

5.4.2 Paquetes externos utilizados en la implementación del sistema distribuido Node.js

Los paquetes NPM externos que han sido instalados para ampliar las funcionalidades de Node.js requeridas para este proyecto. Estos son:

- **body-parser:** hace de middleware que permite analizar los datos de las solicitudes entrantes (`request.body`). Es utilizado para recoger los datos de los formularios una vez realizada la petición POST contra el servidor.
- **express:** es el paquete NPM por excelencia en Node.js que sirve de infraestructura y que proporciona un conjunto de características que facilitan la implementación de la arquitectura de la aplicación.

- **express-session**: sirve para controlar las sesiones de los usuarios iniciados en sesión, almacenar datos en cookies y controlar la caducidad de las sesiones por inactividad.
- **moment**: librería de soporte para la manipulación, validación y formateo de fechas y horas. También permite parsear fechas según la locación.
- **mongoose**: este paquete facilita la conexión y comunicación con la base de datos MongoDB, haciéndolo una tarea muy sencilla y escalable.
- **node-telegram-bot-api**: para el sistema de alarmas de notificaciones de anomalías del dispensador en el canal de Telegram, es necesario utilizar este paquete que permite identificar el canal y enviar notificaciones a cada uno de los usuarios suscritos de forma independiente a cada uno de los seriales de dispensadores con los que se han suscrito.
- **nodemailer**: permite identificar una cuenta de un servidor de correo (para este proyecto usaremos Gmail) y enviar correos de notificaciones de anomalías en el dispensador a cada uno de los correos electrónicos registrados.
- **request**: utilizado para simplificar las llamadas HTTP y HTTPS.
- **serve-favicon**: es un middleware para permite servir el favicon de la aplicación desde un directorio estático.
- **swig**: es un motor de plantillas JavaScript extensible utilizado para renderizar las páginas de la aplicación antes de ser servidas al cliente. También es utilizado en la creación de plantillas HTML que permiten identificar bloques de código recurrentes y secciones donde se añadirán datos específicos del cliente iniciado en sesión y que serán insertados por el servidor en la renderización de la página.

Las conexiones al servidor de correo y al servicio de mensajería Telegram se realizan por medio de credenciales de acceso y a través de los paquetes NPM de extensión de funcionalidad correspondientes expuestos anteriormente.

5.4.3 Implementación del Microcontrolador

El microcontrolador usado para el proyecto dispone de dos módulos controladores independientes, el controlador MEGA 2560 y el ESP8266. El módulo MEGA 2560 se encarga de las comunicaciones con todos los dispositivos conectados al microcontrolador, mientras que el módulo ESP8266 se limita a las comunicaciones a través de internet con la API del servidor.

Al tratarse de dos módulos independientes, su programación también se realiza de forma de independiente. Por ello, es necesario crear dos programas distintos para programar cada uno de los controladores. El microcontrolador dispone de un pequeño *switch* físico que permite seleccionar el módulo que se desea programar.

La comunicación necesaria entre ambos módulos se realiza por el propio puerto serie del microcontrolador. En esta sección se detallarán que datos se intercambian entre los dos módulos para el correcto funcionamiento del dispensador y que resumen las funcionalidades del microcontrolador.

Los comandos enviados desde el módulo MEGA 2560 al controlador ESP8266 son los siguientes:

```

1. //COMANDOS DE EJECUCION EN ESP8266
2. const String motorPastillasDispensadasA = "[MOTORDISPENSADO-A]";
3. const String motorPastillasDispensadasB = "[MOTORDISPENSADO-B]";
4. const String botonConfirmacionPulsado = "[PULSABTN-1]";
5. const String botonEmergenciaPulsado = "[PULSABTN-2]";
6. const String sensorIrDetectado = "[IRDETECCION-1]";
7. const String tempHumeCalor = "[TEMPHUMCALOR]";
8. const String sensorGasDetectado = "[GASDETECCION-1]";
9. const String sensorVibDetectado = "[VIBDETECCION-1]";
    
```

Los comandos de las líneas 2 y 3 permiten notificar que el motor paso a paso ha llegado a su posición y ha dispensado las pastillas correspondientes a la toma de la medicación. Las líneas 4 y 5 notifican que se ha pulsado el botón de emergencia o el de confirmación. La 6 que el sensor IR ha detectado la recogida de la medicación. La 7 envía los valores de temperatura, humedad e índice de calor recogidos. Finalmente, las líneas 8 y 9 notifican la presencia de gas en el ambiente o una caída o golpe del dispensador, respectivamente.

En cuanto a los comandos enviados desde el módulo ESP8266 wifi al controlador MEGA 2560 son:

```

1. //COMANDOS DE EJECUCION EN ATmega2560
2. //Comando completo "[MPAS-A-3]" donde A es el pastillero y 3 la cantidad
3. //de pastillas a tomar (numero de pasos del respectivo pastillero)
4. const String moverPastilleroA = "[MPAS-A]";
5. const String moverPastilleroB = "[MPAS-B]";
6. const String wifiOk = "[WIFIOK1]"; //Comando de confirmacion de conexion WiFi
7. //Comando para solicitar la temperatura y la humedad del dispensador
8. const String solicitarTempHum = "[SOLICTEMHUM-1]";
9. //Comando completo "[LEDNOT-0]" donde 0 es que debe apagarse y 1 encenderse
10. const String ledNotificacion = "[LEDNOT-]";
11. //Comando completo "[ZUMNOT-0]" donde 0 es que debe apagarse y 1 encenderse
12. const String zumabdorNotificacion = "[ZUMNOT-]";
    
```

Los comandos de las líneas 4 y 5 ordenan dispensar las pastillas correspondientes al segundo parámetro de la *ruleta* o dispensador del primer parámetro. La línea 6 notifica que se ha establecido conexión al router WiFi con éxito. La 8 solicita datos de temperatura y humedad actuales al controlador MEGA. La 10 ordena encender o apagar el led de notificación y la 12 emitir o dejar de emitir sonido al zumbador.

Cabe destacar que la gran parte de la lógica del microcontrolador recae sobre el módulo ESP8266, debido a que es el encargado de comunicarse con la API de servidor, *parsear* las respuestas, llevar un control del tiempo para la toma de una medicación, etc. Mientras que el módulo MEGA 2560 haría las veces esclavo esperando las órdenes del ESP8266.

5.4.3.1 Librerías de desarrollo

Puesto que la programación de los módulos MEGA 2560 y ESP8266 WiFi se realiza independientemente para cada uno de ellos, las librerías externas de las que hacen uso para completar la funcionalidad del sistema también son independientes a cada controlador.

Las librerías externas usadas en el controlador MEGA 2560 son:

- **AccelStepper.h:** permite controlar los motores paso a paso de una manera sencilla además de configurar otros parámetros como la velocidad de giro o pasos por vuelta.
- **DHT.h:** se trata de una librería necesaria para controlar el módulo de temperatura y humedad. Permite obtener la temperatura y humedad actual en varias unidades y de forma sencilla

Las librerías utilizadas en el controlador ESP8266 WiFi son más debido a que este módulo es el centro de control del microcontrolador. Estas librerías son:

- **ESP8266WiFi.h:** librería indispensable en este módulo que permite controlar todos los aspectos generales del mismo.
- **RestCliente.h:** permite realizar llamadas GET y POST a la API del servidor y enviar parámetro en la petición.
- **ArduinoJson.h:** librería utilizada para *parsear* los datos obtenidos de la respuesta a peticiones de la API y que facilita obtener la información de la respuesta.
- **NTPClient.h:** librería que permite consultar la fecha y hora actual en cualquier huso horario y que es utilizada para llevar el control del tiempo de toma de medicación transcurrido desde que se dispensaron las pastillas.
- **WiFiUdp.h:** es una librería para hacer uso del protocolo de datagramas de usuarios (UDP o *User Datagram Protocol*) requerida por la librería NTPClient.h para su funcionamiento.

5.4.4 Detalles de construcción del dispensador de medicamentos

En esta sección se comentarán los detalles de construcción del dispensador de medicamentos, su diseño final y otros detalles de modelado e impresión que resulten de relevancia para comprender como ha sido construido.

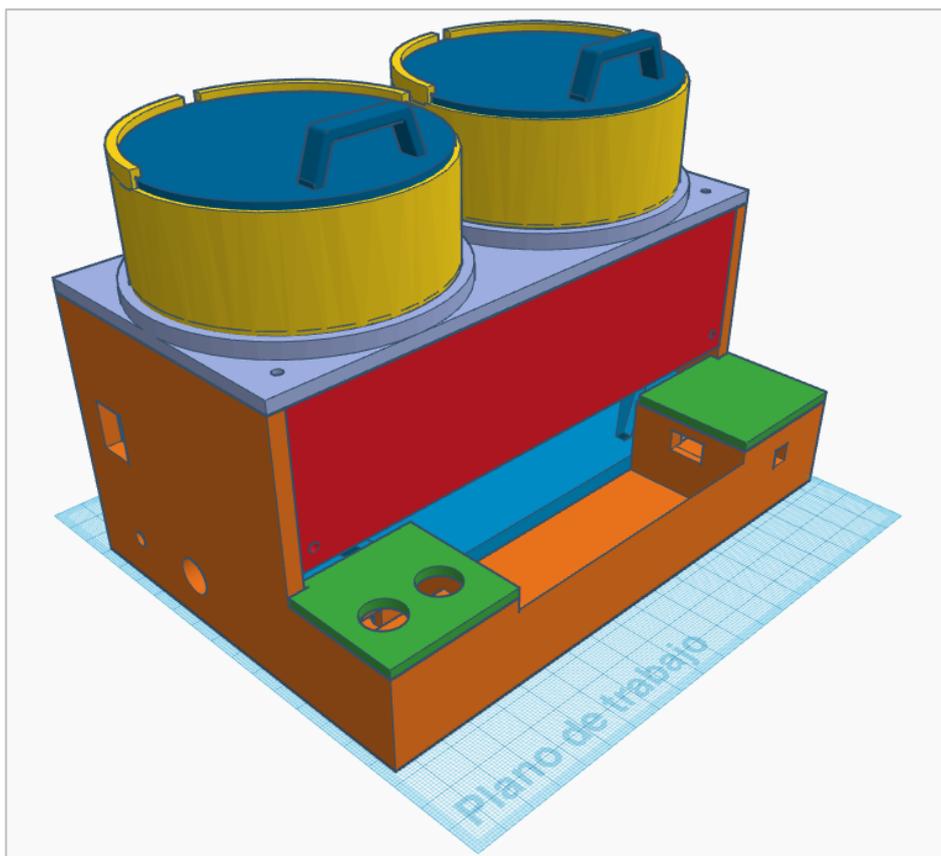


Figura 5.12. Diseño final del dispensador de medicamentos

La imagen anterior muestra el diseño final del dispensador de medicamentos diseñado para ser impreso en 3D con la herramienta [Tinkercad](#). Este diseño es una evolución del primer [boceto](#) creado en el análisis y que cuenta con todas las secciones y compartimentos para alojar todos los componentes electrónicos, motores y sensores necesarios.

El prototipo cuenta con 19 piezas diferentes que han sido impresas por separado y unidas mediante tornillos con el objetivo de no fijar ningún componente electrónico al dispensador, permitiendo así su sustitución en caso necesario.

Para la impresión de todas las piezas se han requerido alrededor de 72 horas de impresión y unos 700g de plástico PLA²².

La gran mayoría de componentes electrónicos, microcontrolador y cableado se encuentran en la parte trasera del dispensador y debajo de la rampa que guía las pastillas hacia su lugar de recogida. Los componentes accionables por el usuario y de detección han sido posicionados en la parte frontal del dispensador. Adicionalmente, los motores que permiten dispensar las pastillas se encuentran acoplados a cada uno de los pastilleros en la parte superior de estos.

²² El ácido poliláctico o poliácido láctico (PLA) es un polímero constituido por moléculas de ácido láctico, con propiedades semejantes a las del tereftalato de polietileno (PET) que se utiliza para hacer envases, pero que además es biodegradable.

Para rellenar los pastilleros, basta con retirar las tapas de cada uno de ellos e insertar las pastillas una a una en cada una de las 12 posiciones de las que dispone. No es necesario insertar 12 pastillas, sino que el número de pastillas insertadas en cada uno de los pastilleros irá acorde a la configuración del tratamiento que el usuario ha realizado en la aplicación web.

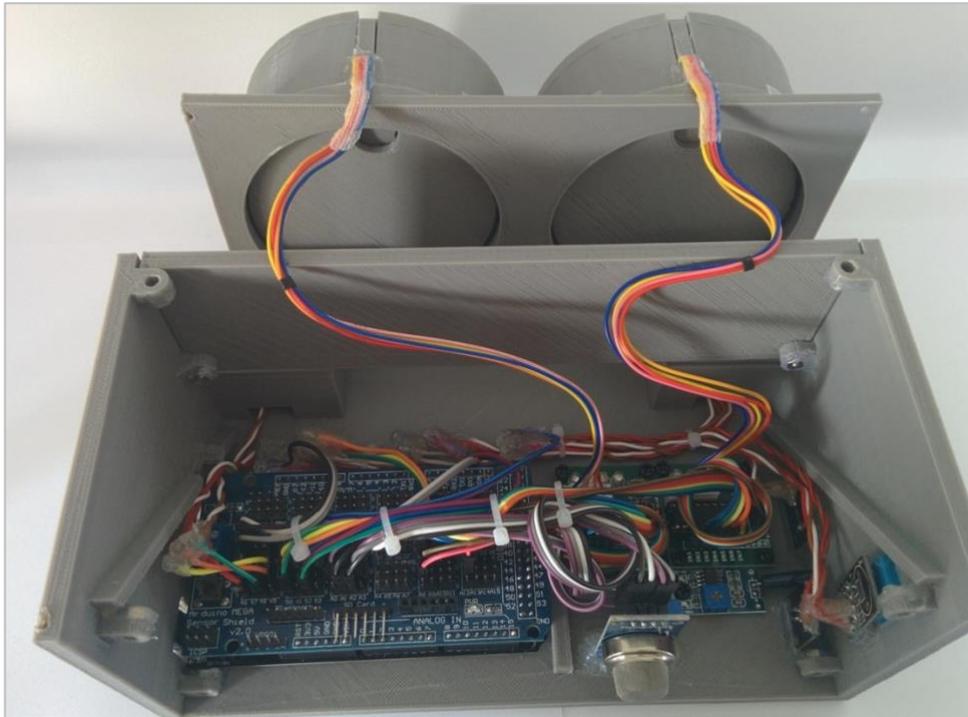


Figura 5.13. Diseño final. Componentes del dispensador

En la imagen anterior pueden visualizarse los componentes electrónicos distribuidos en el interior del dispensador y sus conexiones. Los motores paso a paso se encuentran posicionados en el interior de cada uno de los dos pastilleros. Además, botones, el LED de notificación y el sensor de detección IR, se encuentran situados en la parte frontal del dispensador.

Capítulo 6. Desarrollo de las Pruebas

6.1 Pruebas Unitarias

A continuación, se presentan las pruebas unitarias y de integración realizadas junto al resultado obtenido tras su ejecución. Las pruebas serán realizadas sobre los requisitos del sistema como ya se detalló en la [especificación técnica del plan de pruebas](#).

<i>Requisito Funcional RF-01 – Registrar usuarios en el sistema</i>	
Prueba	Resultado Esperado
Registrar usuario	El sistema posee un usuario más
	Resultado Obtenido El sistema efectivamente posee un usuario más
Prueba	Resultado Esperado
Usar serial no registrado en el sistema	El sistema muestra un mensaje de error de serial no disponible
	Resultado Obtenido El sistema muestra el mensaje de error correctamente
Prueba	Resultado Esperado
Registrar usuario ya existente	El sistema muestra un mensaje de error de email de usuario ya registrado
	Resultado Obtenido El sistema muestra el mensaje de error correctamente
<i>Requisito Funcional RF-02 – Identificar usuarios cliente</i>	
Prueba	Resultado Esperado
Identificar usuario cliente	El sistema redirecciona al usuario al dashboard cliente de la aplicación
	Resultado Obtenido El sistema redirecciona efectivamente al usuario al dashboard del cliente
Prueba	Resultado Esperado
Identificar usuario no registrado	El sistema muestra un mensaje de error de credenciales de acceso inválidas
	Resultado Obtenido El sistema muestra el mensaje de error correctamente
<i>Requisito Funcional RF-03 – Identificar usuarios administrador</i>	
Prueba	Resultado Esperado
Identificar usuario administrado	El sistema redirecciona al usuario al dashboard de administración de la aplicación
	Resultado Obtenido El sistema redirecciona efectivamente al usuario al dashboard de administración
Prueba	Resultado Esperado
Identificar usuario administrador no registrado	El sistema muestra un mensaje de error de credenciales de acceso inválidas
	Resultado Obtenido El sistema muestra el mensaje de error correctamente
<i>Requisito Funcional RF-04 – Consultar estado del dispensador</i>	

Prueba	Resultado Esperado
Consultar estado del pastillero A del dispensador	El usuario comprueba el estado de las pastillas contenidas en el pastillero A del dispensador de forma visual y numérica
	Resultado Obtenido
	El usuario visualiza efectivamente el estado de las pastillas contenidas en el pastillero A
Prueba	Resultado Esperado
Consultar estado del pastillero B del dispensador	El usuario comprueba el estado de las pastillas contenidas en el pastillero B del dispensador de forma visual y numérica
	Resultado Obtenido
	El usuario visualiza efectivamente el estado de las pastillas contenidas en el pastillero B
<i>Requisito Funcional RF-05 – Programar días y horas a las que deben expulsarse las pastillas</i>	
Prueba	Resultado Esperado
Programar día y hora de toma de medicación	El sistema registro un nuevo horario de programación y se muestra al usuario en próximos horarios programados
	Resultado Obtenido
	El sistema registra efectivamente el nuevo horario y aparece en próximos horarios programados
Prueba	Resultado Esperado
Programar día y hora de toma de medicación que se solape con otros horarios	El sistema muestra un mensaje en la operación y el horario con el que se solapa
	Resultado Obtenido
	El sistema muestra el mensaje de error correctamente
Prueba	Resultado Esperado
Programar múltiples horarios de toma de medicación	El sistema registra los nuevos horarios y los muestra al usuario en próximos horarios programados
	Resultado Obtenido
	El sistema registra efectivamente los horarios y aparecen en próximos horarios programados
<i>Requisito Funcional RF-06 – Consultar horarios programados</i>	
Prueba	Resultado Esperado
Consultar próximos horarios programados	El usuario consulta a través de la aplicación la fecha, hora y número de pastillas para los próximos horarios programados
	Resultado Obtenido
	El usuario visualiza efectivamente los detalles de los próximos horarios programados
<i>Requisito Funcional RF-07 – Eliminar horarios programados</i>	
Prueba	Resultado Esperado
Eliminar horarios programados	El sistema elimina el horario programado y actualiza la pantalla con los nuevos próximos horarios
	Resultado Obtenido
	El sistema elimina efectivamente el horario y desaparece de próximos horarios programados
<i>Requisito Funcional RF-08 – Mostrar gráficamente el tiempo de vida del dispensador</i>	
Prueba	Resultado Esperado
Mostrar gráficamente el tiempo de vida del dispensador	El usuario visualiza la distribución horaria de las últimas 24 horas del número de conexiones del dispensador con el servidor
	Resultado Obtenido
	El usuario visualiza efectivamente el gráfico de tiempo de

	vida del dispensador
<i>Requisito Funcional RF-09 – Mostrar gráficamente el registro de temperaturas y humedades</i>	
Prueba	Resultado Esperado
Mostrar gráficamente el registro de temperaturas	El usuario visualiza la distribución horaria de las últimas 24 horas del registro de temperaturas del dispensador
	Resultado Obtenido
	El usuario visualiza efectivamente el gráfico de distribución horaria de la temperatura
Prueba	Resultado Esperado
Mostrar gráficamente el registro de humedad	El usuario visualiza la distribución horaria de las últimas 24 horas del registro de humedad del dispensador
	Resultado Obtenido
	El usuario visualiza efectivamente el gráfico de distribución horaria de la humedad
<i>Requisito Funcional RF-10 – Mostrar datos en tiempo real del dispensador</i>	
Prueba	Resultado Esperado
Mostrar datos en tiempo real del dispensador	El usuario visualiza en tiempo real la temperatura, humedad, índice de calor, estado de conexión y estado de detección de gas y golpe del dispensador
	Resultado Obtenido
	El usuario visualiza efectivamente los datos en tiempo real del dispensador mostrados
<i>Requisito Funcional RF-11 – Modificar datos personales de registro</i>	
Prueba	Resultado Esperado
Modificar datos personales del usuario	El sistema actualiza los datos personales del usuario
	Resultado Obtenido
	El sistema modifica efectivamente los datos personales del usuario
Prueba	Resultado Esperado
Modificar email de usuario por uno ya registrado	El sistema muestra un mensaje de error de email ya registrado en el sistema
	Resultado Obtenido
	El sistema muestra el mensaje de error correctamente
<i>Requisito Funcional RF-12 – Especificar que pastillas han sido introducidas en el dispensador</i>	
Prueba	Resultado Esperado
Modificar que pastillas han sido introducidas en el dispensador	El sistema registra el nombre y número de pastillas del tratamiento del pastillero A y B
	Resultado Obtenido
	El sistema registra efectivamente el tratamiento de los pastilleros A y B del dispensador
Prueba	Resultado Esperado
Modificar tratamiento de un pastillero ya iniciado	El sistema muestra un mensaje de error de imposibilidad en el cambio de tratamiento de un pastillero si dispone de horarios programados y/o no se encuentra en la posición de inicio
	Resultado Obtenido
	El sistema muestra el mensaje de error correctamente
<i>Requisito Funcional RF-13 – Configurar tiempo de espera que determina que no se ha tomado la medicación</i>	
Prueba	Resultado Esperado
Configurar tiempo de espera	El sistema modifica el tiempo de espera requerido para tomar

de toma de medicación	la medicación del dispensador
	Resultado Obtenido
	El sistema modifica efectivamente el tiempo de espera de toma de medicación del dispensador
<u>Requisito Funcional RF-14 – Configurar el modo en el que se considera tomada la medicación</u>	
Prueba	Resultado Esperado
Activar y desactivar la opción de detección de toma	El sistema modifica la configuración del dispensador con la detección o no detección de la retirada de las pastillas
	Resultado Obtenido
	El sistema modifica efectivamente la configuración de detección de toma de medicación del dispensador
<u>Requisito Funcional RF-15 – Configurar la emisión de notificaciones de toma de medicación en el dispensador</u>	
Prueba	Resultado Esperado
Activar y desactivar la opción de botón de confirmación	El sistema modifica la configuración del dispensador con el requerimiento o no de confirmar la toma de la medicación
	Resultado Obtenido
	El sistema modifica efectivamente la configuración de confirmación de toma de medicación del dispensador
<u>Requisito Funcional RF-16 – Añadir o eliminar correos electrónicos de notificación de alarmas</u>	
Prueba	Resultado Esperado
Añadir un nuevo email extra de notificación	El sistema registra el nuevo email de notificación para el dispensador
	Resultado Obtenido
	El sistema almacena efectivamente el nuevo email de notificación del dispensador
Prueba	Resultado Esperado
Eliminar email extra de notificación	El sistema elimina el email de notificación para el dispensador
	Resultado Obtenido
	El sistema elimina el efectivamente el email de notificación
<u>Requisito Funcional RF-17 – Suscribirse al canal de Telegram de notificaciones</u>	
Prueba	Resultado Esperado
Suscribirse al canal de Telegram	El sistema almacena el ID de Telegram del usuario suscrito a las notificaciones del dispensador
	Resultado Obtenido
	El sistema registra efectivamente el ID de usuario de Telegram para el dispensador
<u>Requisito Funcional RF-18 – Configurar notificaciones de alarmas a los emails y canal de Telegram</u>	
<u>Requisito Funcional RF-18.1 – Activar o desactivar notificación de pérdida de conexión</u>	
Prueba	Resultado Esperado
Activar y desactivar la notificación de pérdida de conexión	El sistema modifica la configuración de notificaciones del dispensador para enviarlas cuando se pierde la conexión (activada) o no (desactivada)
	Resultado Obtenido
	El sistema modifica efectivamente la configuración de notificación de pérdida de conexión del dispensador
Prueba	Resultado Esperado
Modificar el número de	El sistema modifica la configuración de las notificaciones del

minutos	dispensador para enviarlas cuando se pierda la conexión del dispensador durante los minutos especificados
	Resultado Obtenido
	El sistema modifica efectivamente los minutos que debe permanecer desconectado el dispensador para emitir notificaciones
Requisito Funcional RF-18.2 – Activar o desactivar notificación de pocas pastillas restantes	
Prueba	Resultado Esperado
Activar y desactivar la notificación de pocas pastillas restantes	El sistema modifica la configuración de notificaciones del dispensador para enviarlas cuando queden pocas pastillas en uno de los pastilleros (activada) o no (desactivada)
	Resultado Obtenido
	El sistema modifica correctamente la configuración de notificación de pocas pastillas restantes
Prueba	Resultado Esperado
Modificar el número de pastillas	El sistema modifica la configuración de las notificaciones del dispensador para enviarlas cuando queden menos pastillas de las especificadas
	Resultado Obtenido
	El sistema modifica correctamente las pastillas restantes en los pastilleros del dispensador para emitir notificaciones
Requisito Funcional RF-18.3 – Activar o desactivar notificación de anomalía en la temperatura	
Prueba	Resultado Esperado
Activar y desactivar la notificación de anomalía en la temperatura	El sistema modifica la configuración de notificaciones del dispensador para enviarlas cuando se detecte una anomalía en la temperatura(activada) o no (desactivada)
	Resultado Obtenido
	El sistema modifica efectivamente la configuración de notificación de anomalía en la temperatura del dispensador
Prueba	Resultado Esperado
Modificar la temperatura para que se considere anomalía	El sistema modifica la configuración de las notificaciones del dispensador para enviarlas cuando la temperatura sea inferior o superior a los valores de anomalía establecidos
	Resultado Obtenido
	El sistema modifica correctamente los valores inferior y superior de temperatura
Requisito Funcional RF-18.4 – Activar o desactivar notificación de anomalía en la humedad	
Prueba	Resultado Esperado
Activar y desactivar la notificación de anomalía en la humedad	El sistema modifica la configuración de notificaciones del dispensador para enviarlas cuando se detecte una anomalía en la humedad(activada) o no (desactivada)
	Resultado Obtenido
	El sistema modifica efectivamente la configuración de notificación de anomalía en la humedad del dispensador
Prueba	Resultado Esperado
Modificar la humedad para que se considere anomalía	El sistema modifica la configuración de las notificaciones del dispensador para enviarlas cuando la humedad sea inferior o superior a los valores de anomalía establecidos
	Resultado Obtenido
	El sistema modifica correctamente los valores inferior y superior de temperatura

	superior de humedad
<i>Requisito Funcional RF-18.5 – Activar o desactivar notificación de detección de gas</i>	
Prueba	Resultado Esperado
Activar y desactivar la notificación de detección de gas	El sistema modifica la configuración de notificaciones del dispensador para enviarlas cuando se detecte gas(activada) o no (desactivada)
	Resultado Obtenido
	El sistema modifica efectivamente la configuración de notificación de detección de gas
<i>Requisito Funcional RF-18.6 – Activar o desactivar notificación de golpe del dispensador</i>	
Prueba	Resultado Esperado
Activar y desactivar la notificación de detección de golpe en el dispensador	El sistema modifica la configuración de notificaciones del dispensador para enviarlas cuando se detecte un golpe(activada) o no (desactivada)
	Resultado Obtenido
	El sistema modifica efectivamente la configuración de notificación de detección de golpe en el dispensador
<i>Requisito Funcional RF-19 – Añadir nuevos seriales al sistema</i>	
Prueba	Resultado Esperado
Añadir un nuevo serial al sistema	El sistema registra un nuevo serial de dispensador
	Resultado Obtenido
	El sistema registra efectivamente el nuevo serial
Prueba	Resultado Esperado
Añadir un serial ya registrado	El sistema muestra un mensaje de error de serial ya registrado
	Resultado Obtenido
	El sistema muestra el mensaje de error correctamente
<i>Requisito Funcional RF-20 – Consultar seriales registrados en el sistema</i>	
Prueba	Resultado Esperado
Consultar seriales registrados	El usuario visualiza los seriales registrados, número de usuario registrados con cada serial y la fecha y hora de la ultima conexión del dispensador
	Resultado Obtenido
	El usuario visualiza efectivamente los detalles de los seriales registrados en el sistema
<i>Requisito Funcional RF-21 – Eliminar seriales del sistema</i>	
Prueba	Resultado Esperado
Eliminar serial registrado	El sistema elimina el serial
	Resultado Obtenido
	El sistema elimina efectivamente el serial registrado
Prueba	Resultado Esperado
Eliminar serial con usuario registrados	El sistema muestra un mensaje de error en la eliminación del serial por disponer de usuarios registrados
	Resultado Obtenido
	El sistema muestra el mensaje de error correctamente
<i>Requisito Funcional RF-22 – Consultar usuarios registrado en el sistema</i>	
Prueba	Resultado Esperado
Consultar usuario registrados	El usuario visualiza el email de los usuarios registrados y el serial de su dispensador
	Resultado Obtenido

	El usuario visualiza correctamente los detalles de los usuarios registrados
<u>Requisito Funcional RF-23 – Eliminar usuarios del sistema</u>	
Prueba	Resultado Esperado
Eliminar usuario registrado	El sistema elimina el usuario
	Resultado Obtenido
	El sistema elimina efectivamente el usuario registrado
<u>Requisito Funcional RF-24 – Recoger y tomar medicación</u>	
Prueba	Resultado Esperado
Recoger medicación del dispensador	El sistema almacena el evento de recogida de las pastillas dispensadas de un horario de programación
	Resultado Obtenido
	El sistema almacena efectivamente el evento de recogida de las pastillas para el horario
<u>Requisito Funcional RF-25 – Confirmar medicación tomada</u>	
Prueba	Resultado Esperado
Confirmar la toma de la medicación	El sistema almacena el evento de confirmación de toma de las pastillas dispensadas de un horario de programación
	Resultado Obtenido
	El sistema almacena efectivamente el evento de confirmación de toma de la medicación para el horario
<u>Requisito Funcional RF-26 – Emitir notificación de emergencia</u>	
Prueba	Resultado Esperado
Emitir notificación de emergencia	El sistema envía notificaciones a los emails del dispensador y canal de Telegram de situación de emergencia
	Resultado Obtenido
	El sistema envía correctamente notificaciones por email y canal de Telegram de la situación de emergencia
<u>Requisito Funcional RF-27 – Expulsar pastillas</u>	
Prueba	Resultado Esperado
Dispensar pastillas	El dispensador expulsa pastillas y el sistema almacena el nuevo estado de los pastilleros tras expulsarlas
	Resultado Obtenido
	El dispensador expulsa las pastillas de un horario efectivamente
<u>Requisito Funcional RF-28 – Registrar estado de dispensador</u>	
Prueba	Resultado Esperado
Registrar nuevo estado de los pastilleros del dispensador	El sistema almacena el nuevo estado de los pastilleros del dispensador una vez a expulsado las pastillas de un horario de programación
	Resultado Obtenido
	El sistema almacena el nuevo estado de los pastilleros del dispensador correctamente
<u>Requisito Funcional RF-29 – Registrar que el dispensador está conectado a internet</u>	
Prueba	Resultado Esperado
Registrar conexión del dispensador	El sistema registra el estado de conexión del dispensador.
	Resultado Obtenido
	El sistema registra efectivamente el estado de conexión del dispensador
<u>Requisito Funcional RF-30 – Almacenar datos ambientales de temperatura y humedad</u>	

Prueba	Resultado Esperado
Almacenar datos ambientales de temperatura y humedad	El sistema registra datos ambientales del dispensador cada minuto
	Resultado Obtenido El sistema registra correctamente los datos ambientales del dispensador cada minuto
<i>Requisito Funcional RF-31 – Almacenar detección de gas o golpe del dispensador</i>	
Prueba	Resultado Esperado
Almacenar estado de detección de gas o golpe	El sistema registra el estado de detección de gas y caída del dispensador cada minuto
	Resultado Obtenido El sistema registra efectivamente el estado de detección de gas y golpe del dispensador cada minuto
<i>Requisito Funcional RF-32 – Enviar notificación por email y canal de Telegram cuando no se tome la medicación</i>	
Prueba	Resultado Esperado
Enviar notificaciones a los emails y canal de Telegram de medicación no tomada	El sistema envía email a los usuarios del dispensador y canal de Telegram de cuando la medicación de un horario de programación no haya sido tomada
	Resultado Obtenido El sistema envía efectivamente emails y mensajes al canal de Telegram cuando la medicación no ha sido tomada
<i>Requisito Funcional RF-33 – Enviar notificación por email y canal de Telegram cuando se produzca una anomalía en el dispensador</i>	
Prueba	Resultado Esperado
Enviar notificaciones a los emails y canal de Telegram de anomalía en el dispensador	El sistema envía email a los usuarios del dispensador y canal de Telegram de cuando se detecte una anomalía en los datos del dispensador
	Resultado Obtenido El sistema envía efectivamente emails y mensajes al canal de Telegram de la anomalía del dispensador
<i>Requisito Funcional RF-34 – Enviar notificación por email y canal de Telegram cuando se pulse el botón de emergencia del dispensador</i>	
Prueba	Resultado Esperado
Enviar notificaciones a los emails y canal de Telegram de emergencia	El sistema envía email a los usuarios del dispensador y canal de Telegram de cuando se pulse el botón de emergencia del dispensador
	Resultado Obtenido El sistema envía efectivamente la notificación de emergencia mediante emails y mensajes al canal de Telegram
<i>Requisito Funcional RF-35 – Gestionar horarios de toma de medicación</i>	
Prueba	Resultado Esperado
Gestionar horarios de toma de medicación	El dispensador controla el tiempo establecido para tomar la medicación dispensada
	Resultado Obtenido El dispensador controla efectivamente el tiempo de toma para la medicación dispensada
<i>Requisito Funcional RF-36 – Emitir notificaciones sonoras y luminosas</i>	
Prueba	Resultado Esperado
Emitir notificaciones sonoras y luminosas	El dispensador emite notificaciones sonoras y/o luminosas cuando se requieran tomar pastillas dispensadas por un

	horario de toma programado
	Resultado Obtenido
	El dispensador notifica efectivamente el requerimiento de toma de medicación
<i>Requisito Funcional RF-37 – Configurar dispensador</i>	
Prueba	Resultado Esperado
Configurar dispensador	El dispensador se configura con los datos de los ajustes que el usuario ha realizado en la aplicación web
	Resultado Obtenido
	El dispensador se configura efectivamente con los datos de configuración recibidos del servidor

6.2 Pruebas de Usabilidad

En esta sección se ejecutará la rutina establecida en la [especificación técnica de requisitos](#) sobre los perfiles de usuario mencionados para observar cómo interactúan con la aplicación web y con el dispensador de medicamentos y anotar las opiniones al respecto.

Las pruebas usabilidad fueron llevadas a cabo durante todo el proceso de desarrollo del proyecto con el objetivo de ir mejorando la calidad de interacción de los usuarios con la aplicación web y con el dispensador hasta llegar a un punto de satisfacción de la forma de uso de ambos elementos por parte de los usuarios.

La realización de estas pruebas a distintos perfiles de usuario a lo largo del desarrollo permitió mejorar varios aspectos de la aplicación y del dispensador, sobre todo en el desarrollo temprano de la aplicación y del prototipado del dispensador.

Finalmente, con la aplicación web y el dispensador de medicamentos terminados y con todos los cambios destinados a mejorar la usabilidad de ambos componentes, se realizó una última batería de pruebas a cada uno de los perfiles de usuario analizados de la que se obtuvieron los siguientes resultados.

En las tablas presentadas a continuación, debe tenerse en cuenta que los pasos (**marcados en negrita**) fueron evaluados por el propio autor de las pruebas, mientras que las preguntas fueron respondidas por los usuarios de la aplicación web y del dispensador de medicamentos.

Pruebas de usabilidad sobre la <i>Aplicación Web</i>					
Paso o pregunta	Perfil de Usuario	Respuesta			
		Siempre (OK)	Frecuentemente	Ocasionalmente	Nunca (NO)
1. Registrarse en el sistema	Sin con. informáticos	X			
	Con con. informáticos	X			
	Con habilidad. en apps	X			
2. Iniciar sesión con el usuario registrado	Sin con. informáticos	X			
	Con con. informáticos	X			
	Con habilidad. en apps	X			
3. ¿Comprende lo que le muestra la pantalla de inicio?	Sin con. informáticos		X		
	Con con. informáticos	X			
	Con habilidad. en apps	X			
4. Programar un único horario de toma de medicación	Sin con. informáticos	X			
	Con con. informáticos	X			
	Con habilidad. en apps	X			
5. Programar múltiples horarios de toma de medicación	Sin con. informáticos		X		
	Con con. informáticos	X			
	Con habilidad. en apps	X			
6. ¿Le ha resultado sencillo programar múltiples horarios?	Sin con. informáticos	X			
	Con con. informáticos	X			
	Con habilidad. en apps	X			
7. Navegar a la pantalla	Sin con. informáticos	X			

del dispensador	Con con. informáticos	X			
	Con habilidad. en apps	X			
8. ¿Comprende los datos y las visualizaciones que se le muestran?	Sin con. informáticos		X		
	Con con. informáticos	X			
9. Navegar a la pantalla de ajustes	Con habilidad. en apps	X			
	Sin con. informáticos	X			
	Con con. informáticos	X			
10. Configurar tratamiento del pastillero B	Con habilidad. en apps	X			
	Sin con. informáticos	X			
	Con con. informáticos	X			
11. ¿Conoce cuál es el objetivo de modificar el tratamiento de un pastillero?	Con habilidad. en apps		X		
	Sin con. informáticos		X		
	Con con. informáticos	X			
12. Añadir un nuevo email de notificación	Con habilidad. en apps	X			
	Sin con. informáticos	X			
	Con con. informáticos	X			
13. Configurar las notificaciones para que se envíen cuando queden menos de 3 pastillas	Con habilidad. en apps	X			
	Sin con. informáticos	X			
	Con con. informáticos	X			
14. ¿Comprende todas las opciones del configurador de notificaciones?	Con habilidad. en apps		X		
	Sin con. informáticos		X		
	Con con. informáticos		X		
15. Iniciar sesión en la aplicación con un smartphone	Con habilidad. en apps	X			
	Sin con. informáticos	X			
	Con con. informáticos	X			
16. Programar un horario de medicación	Con habilidad. en apps	X			
	Sin con. informáticos		X		
	Con con. informáticos	X			
17. ¿Le resulta sencillo el uso de la versión móvil?	Con habilidad. en apps	X			
	Sin con. informáticos	X			
	Con con. informáticos	X			
18. ¿Añadiría o cambiaría algo en la aplicación?	Con habilidad. en apps				X
	Sin con. informáticos				X
	Con con. informáticos				X

Pruebas de usabilidad sobre el <i>Dispensador de Medicamentos</i>					
Paso o pregunta	Perfil de Usuario	Respuesta			
		Siempre (OK)	Frecuentemente	Ocasionalmente	Nunca (NO)
1. Recoger pastillas dispensadas	Sin con. informáticos	X			
	Edad avanzada	X			
2. ¿El dispensador le ha notificado clara y correctamente la toma de la medicación?	Sin con. informáticos	X			
	Edad avanzada	X			
3. Confirmar toma de medicación	Sin con. informáticos	X			
	Edad avanzada		X		

4. Emitir situación de emergencia	Sin con. informáticos	X			
	Edad avanzada	X			
5. ¿Recuerda los pasos que hemos seguidos para tomar la medicación?	Sin con. informáticos	X			
	Edad avanzada		X		
6. ¿Cambiaría algo del dispensador?	Sin con. informáticos				X
	Edad avanzada				X

Capítulo 7. Manuales del Sistema

7.1 Manual de Instalación

En este manual del sistema se describe el conjunto de pasos y herramientas necesarias para instalar el controlador del microcontrolador y el sistema distribuido, aplicación web y base de datos con los que lograr la puesta en marcha del sistema automático.

7.1.1 Instalación del microcontrolador del dispensador

El primer paso será descargar e instalar el software IDE de Arduino de <https://www.arduino.cc/en/main/software>.

Con el IDE de Arduino instalado, el siguiente paso instalar la librería *SoftwareSerial* desde el gestor de librerías del propio IDE de desarrollo.

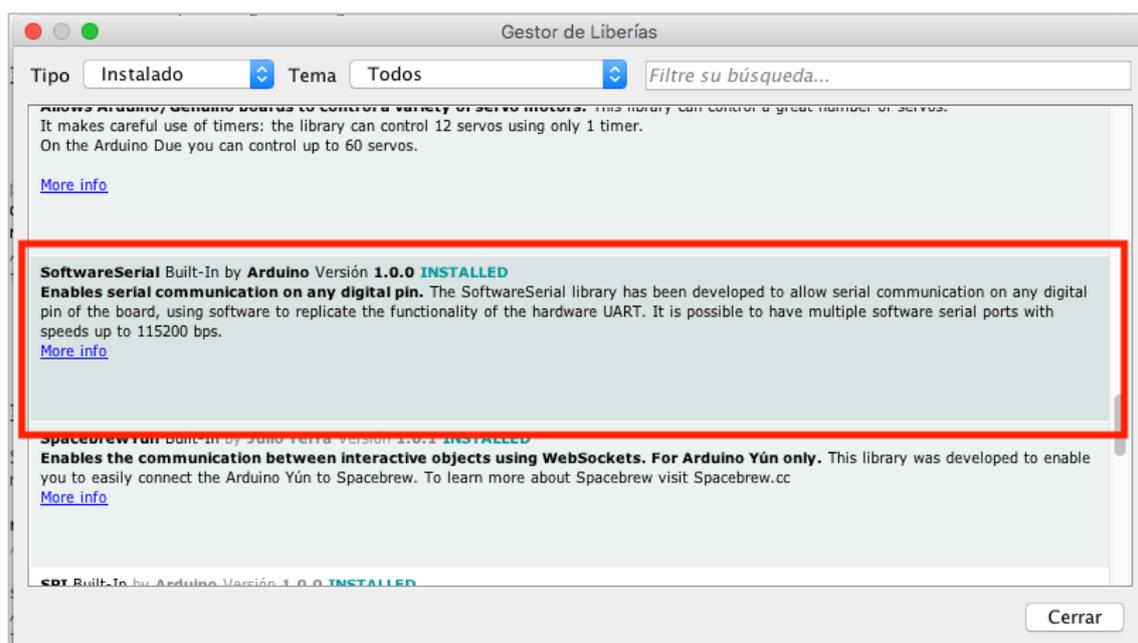


Figura 7.1. Instalación del microcontrolador del dispensador – *SoftwareSerial*

Como ya se comentó a lo largo del documento, la programación del microcontrolador se realiza en dos partes y a través de dos ficheros de desarrollo diferentes. Por un lado, la programación del módulo ATmega2560 y por otra, la programación del módulo ESP8266.

Para la programación de cada uno de los módulos del microcontrolador, es necesario configurar el *switch* físico del microcontrolador que se muestra en la siguiente imagen.

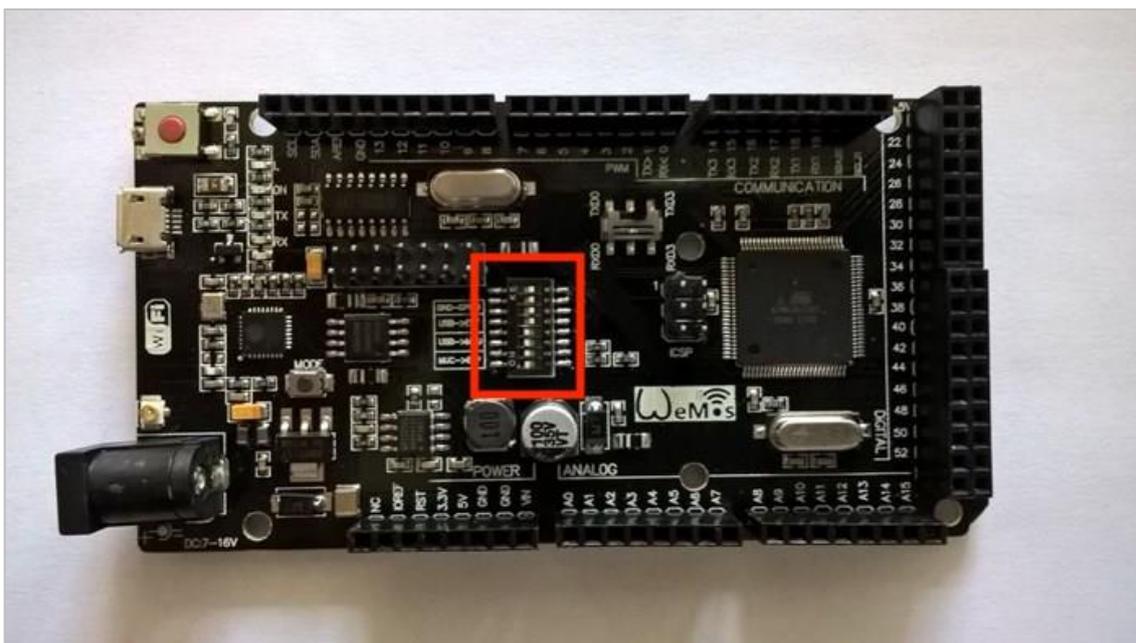


Figura 7.2. Instalación del microcontrolador del dispensador – Switch del microcontrolador

En primer lugar, se programará el módulo wifi ESP8266. Para ello, será necesario posicionar el *switch* del dispensador activando los pines 5-6-7 dejando el resto desactivados.

Una vez configurado el *switch*, se conecta el microcontrolador al ordenador para subir la programación de dicho módulo. La implementación de este módulo se encuentra en un archivo llamado *SmartMedicineDispenser-ESP8266.ino*.

```

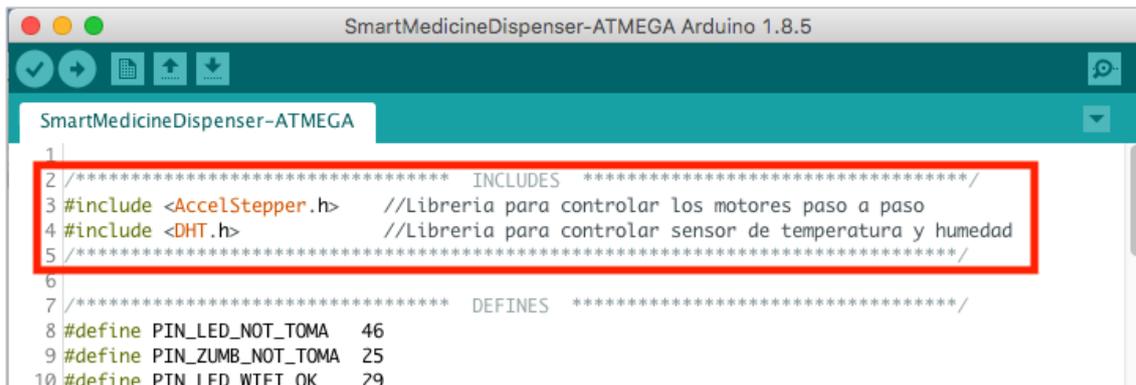
SmartMedicineDispenser-ESP8266 Arduino 1.8.5
SmartMedicineDispenser-ESP8266
1
2 /***** INCLUDES *****/
3 #include <ESP8266WiFi.h> //WiFi ESP8266
4 #include "RestClient.h" //Peticiones REST contra el servidor NodeJS
5 #include <ArduinoJson.h> //Parser JSON
6 #include <NTPClient.h> //Network Time Protocol Cliente. Obtener fecha y hora
7 #include <WiFiUdp.h> //User Datagram Protocol. Necesario para NTPClient
8 /*****
9
10 /***** CONSTANTES *****/
11 //Serial Caja de Medicamentos
12 const String serial = "NZHH3-DHK2W-8SX7I-SES83-7RLDP";
13 const int numPastilleros = 2;
14
15 //Datos de conexion al WiFi
16 /*
17 const char* ssid = "JAZZTEL_U9rS";
18 const char* password = "ku7j7br5phzx";
19 */
20 const char* ssid = "AndroidAP";
21 const char* password = "abcd1234";
22
23 //Datos de conexion a la API REST
24 const char* host = "smart-medicine-dispenser.herokuapp.com";
    
```

Figura 7.3. Instalación del microcontrolador del dispensador – Módulo ESP8266

Cabe destacar que para la programación de este módulo es necesario tener instaladas las librerías externas de las que hace uso la implementación y que se muestran en la imagen anterior.

Con el módulo ESP8266 programado, el siguiente paso es programar el módulo ATmega2560. Para ello es necesario activar los pines del *switch* del microcontrolador 1-2-3-4, dejando el resto desactivados.

Conectamos de nuevo el microcontrolador al ordenador y abrimos el archivo llamado *SmartMedicineDispenser-ATMEGA.ino*.



```

SmartMedicineDispenser-ATMEGA Arduino 1.8.5
SmartMedicineDispenser-ATMEGA
1
2 /***** INCLUDES *****/
3 #include <AccelStepper.h> //Libreria para controlar los motores paso a paso
4 #include <DHT.h> //Libreria para controlar sensor de temperatura y humedad
5 /*****
6
7 /***** DEFINES *****/
8 #define PIN_LED_NOT_TOMA 46
9 #define PIN_ZUMB_NOT_TOMA 25
10 #define PIN_LED_WIFI_OK 29
    
```

Figura 7.4. Instalación del microcontrolador del dispensador – Módulo ATmega2560

Para la programación de este este módulo únicamente es necesario tener instaladas las librerías del sensor de temperatura y humedad (*DHT*) y la librería (*AccelStepper*) para el control de los motores paso a paso.

Siguiendo estos pasos, ya tendríamos el microcontrolador del dispensador programado y listo para ser usado.

7.1.2 Instalación del sistema distribuido, aplicación web y base de datos.

Para la instalación del sistema distribuido y el despliegue de la aplicación web, es necesario tener iniciada una instancia de la base de datos MongoDB. Para ello es necesario instalar MongoDB disponible en <https://www.mongodb.com/download-center?jmp=nav#community>.

Con la base de dato instalada, basta con configurar el archivo *iniciarDB.bat* del proyecto Node.js con la ruta en la que se ha instalado MongoDB y ejecutar el archivo.

```

1 cd C:\Program Files\MongoDB\Server\3.2\bin
2 mongod --dbpath C:\data\db

```

Figura 7.5. Instalación del sistema distribuido, aplicación web y base de datos – Iniciar MongoDB

Con esto ya tendríamos la base de datos desplegada en local. El siguiente paso es desplegar el proyecto Node.js que contiene el propio sistema y la aplicación web.

Para ello es necesario tener instalado Node.js, disponible en <https://nodejs.org/en/>. Con Node.js instalado, ejecutamos `npm install` sobre el directorio del proyecto. Esto instalará todas las dependencias necesarias para el despliegue.

Y por último ejecutamos `node app.js` en el directorio del proyecto Node.js para desplegar la aplicación.

```

ERUER>node app.js
Servidor activo

```

Figura 7.6. Instalación del sistema distribuido, aplicación web y base de datos – Despliegue de la aplicación

Siguiendo estos pasos, ya tendríamos la aplicación desplegada en local. Es importante saber que, para empezar a utilizar la aplicación, es necesario añadir el usuario administrador a la base de datos MongoDB para poder iniciar sesión como administrador y poder registrar números de serie del dispensador con los que poder registrar usuarios.

```

1.  {
2.    "serial" : "-",
3.    "nombre" : "Administrador",
4.    "apellido" : "Administrador",
5.    "telefono" : "-",
6.    "tipo" : "ADMIN",
7.    "email" : "admin@smd.com",
8.    "password" : "4bfbafe08be32f50ad5cf35700a74c33fab1db34d786a1053828c70aa33fd75b
9.  }

```

El código anterior debe ser añadido a la colección *usuarios* de MongoDB para poder iniciar sesión en la aplicación como usuario administrador. La contraseña encriptada de este usuario corresponde a *Admin1234*. Una vez iniciada la sesión en la aplicación podemos cambiar la contraseña por otra más segura.

7.2 Manual de Ejecución

La aplicación web se encuentra desplegada y con su instancia de MongoDB iniciada en la nube <https://smart-medicine-dispenser.herokuapp.com>

Con la programación realizada sobre cada uno de los módulos del microcontrolador y la instalación de la base de datos MongoDB (con el usuario administrador añadido) y Node.js, solo quedaría iniciar cada uno de los componentes.

Primero iniciaremos la instancia de la base de datos MongoDB. Para ello, simplemente debemos ejecutar el fichero `iniciarDB.bat` contenido en la carpeta del proyecto Node.js.

```

C:\Windows\system32\cmd.exe
C:\Users\U0234549\Desktop\WorkspaceEclipseOxigen\SmartMedicineDispenser-NODEJS_SERVER>cd C:\Program Files\MongoDB\Server\3.2\bin
C:\Program Files\MongoDB\Server\3.2\bin>mongod --dbpath C:\data\db
2018-06-26T22:54:44.238+0200 I CONTROL [main] NOTICE: 64-bit host=U0234549-PC
is installed, no need to zero-out data files
2018-06-26T22:54:44.237+0200 I CONTROL [initandlisten] MongoDB starting : pid=9528 port=27017 dbpath=C:\data\db 64-bit
2018-06-26T22:54:44.238+0200 I CONTROL [initandlisten] targetMinOS: Windows 7/Windows Server 2008 R2
2018-06-26T22:54:44.238+0200 I CONTROL [initandlisten] db version v3.2.10
2018-06-26T22:54:44.238+0200 I CONTROL [initandlisten] git version: 79d9b3ab5ce20f51c272b4411202710a082d0317
2018-06-26T22:54:44.238+0200 I CONTROL [initandlisten] OpenSSL version: OpenSSL
    
```

Figura 7.7. Ejecución del sistema – MongoDB

Este fichero desplegará automáticamente la instancia de la base de datos en local a la que tendrá acceso el proyecto Node.js que contiene el sistema.

Una vez iniciada la base de datos, el siguiente paso es desplegar el proyecto Node.js del sistema. Para ello basta con ejecutar `node app` sobre la raíz del proyecto.

```

Administrador: C:\Windows\system32\cmd.exe - node app
C:\Users\U0234549\Desktop\WorkspaceEclipseOxigen\SmartMedicineDispenser-NODEJS_SERVER>node app
Servidor activo
    
```

Figura 7.8. Ejecución del sistema – Node.js

Una vez desplegada la aplicación aparecerá un mensaje de *Servidor activo* que confirma el despliegue correcto de la aplicación.

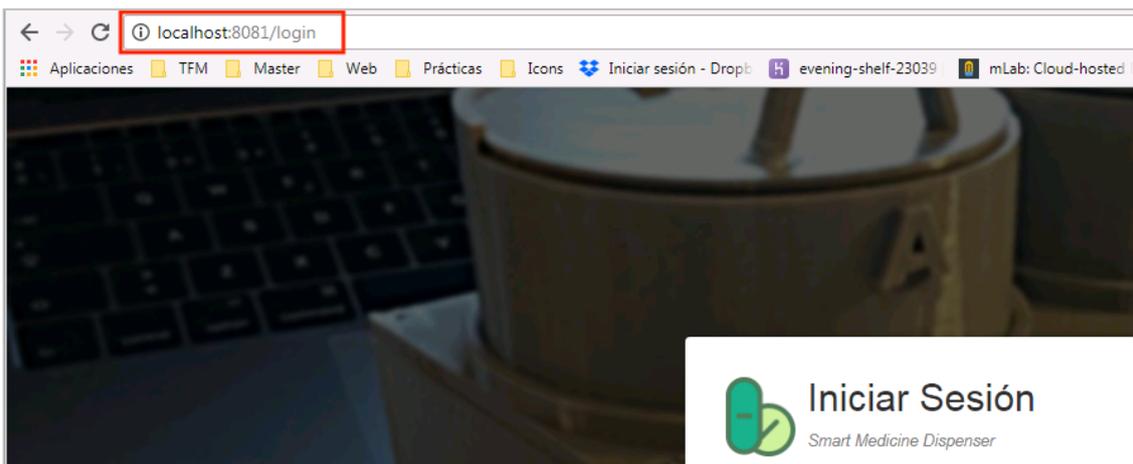


Figura 7.9. Ejecución del sistema – Despliegue en local

La aplicación pasará a estar disponible a través del puerto 8081 de localhost.

Ahora será necesario registrar un nuevo usuario con el serial del dispensador. En serial introducido debe ser el mismo que el programado en fichero *SmartMedicineDispenser-ESP8266.ino*.

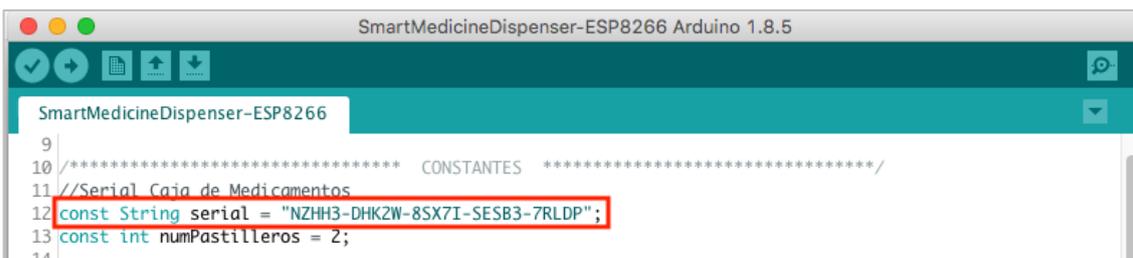


Figura 7.10. Ejecución del sistema - Serial del dispensador

Una vez registrado el usuario, iniciamos sesión con el email y la contraseña utilizados para el registro y nos dirigimos a la página *Dispensador*.

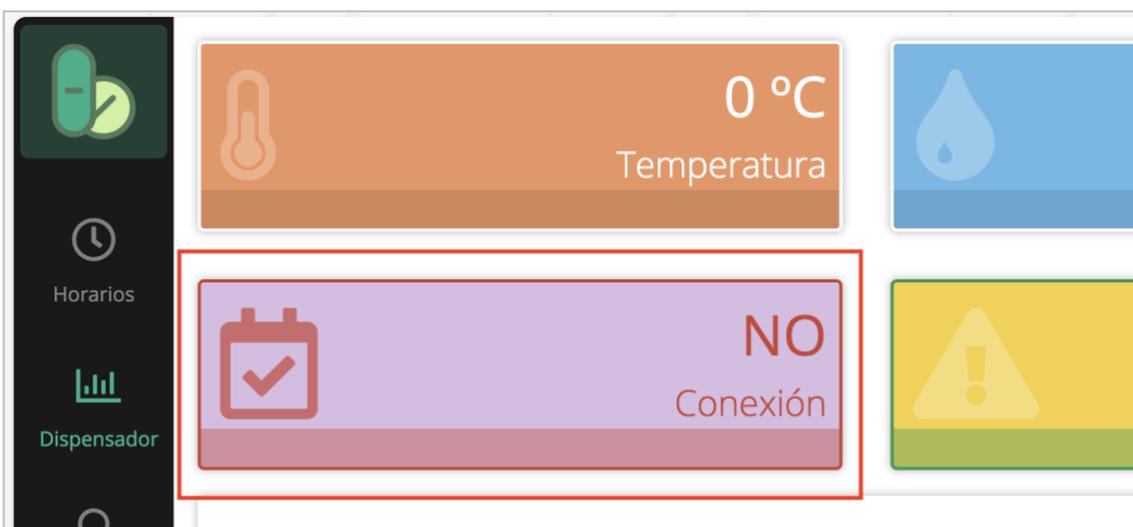


Figura 7.11. Ejecución del sistema – Desconexión del dispensador

En la pantalla de *Dispensador* podemos comprobar que el dispensador se encuentra desconectado. La imagen anterior muestra el lugar de la pantalla donde se muestra que el dispensador de medicamentos se encuentra desconectado.

El siguiente paso es conectar el microcontrolador del dispensador programado a la corriente y esperar a que se encienda el led de notificación de conexión. El lugar en el que se encuentra el LED en el dispensador se detalla en el [manual de usuario](#).

Una vez el led de notificación de conexión se ha encendido, podemos regresar a la pantalla de *Dispensador* y comprobar que ahora se encuentra conectado al sistema.

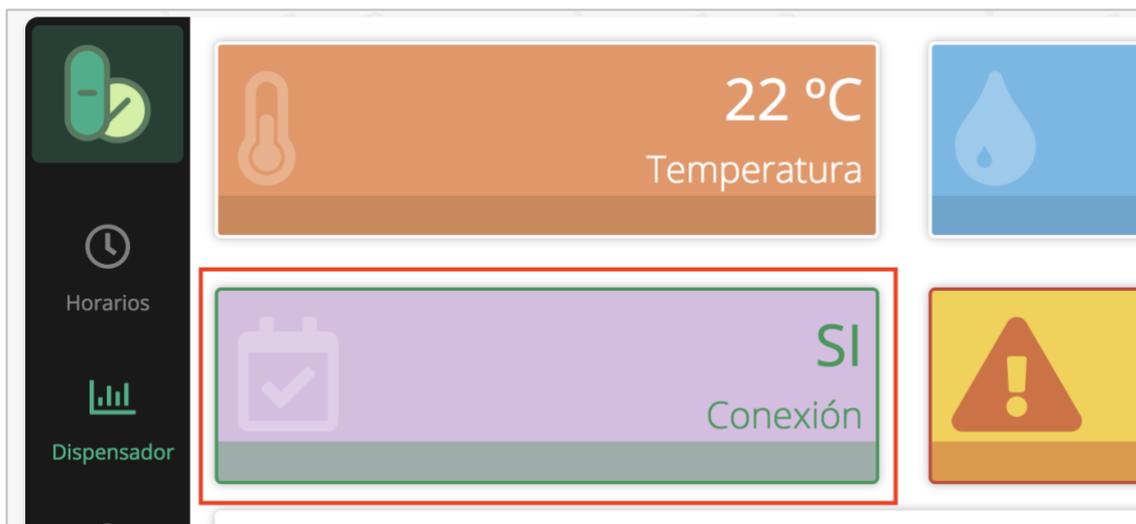


Figura 7.12. Ejecución del sistema – Conexión del dispensador

Siguiendo todos estos pasos ya tendríamos el sistema conectado funcionando y listo para su uso. Ahora podemos probar a programar un horario de toma de medicación próximo y comprobar que el selector del pastillero programado gira a la hora indicada.

7.3 Manual de Usuario

El manual de usuario del sistema servirá como referencia de consulta del funcionamiento de la aplicación web y del dispensador de medicamentos. En él se detallarán todos los componentes y funcionalidades disponibles de ambos componentes del sistema.

7.3.1 Manual de la aplicación web

En el manual de usuario de la aplicación web se detallan las funcionalidades de cada una de las pantallas y la forma de navegación entre las mismas. Esta sección se divide en dos: el manual del usuario cliente de la aplicación y el manual del administrador del sistema.

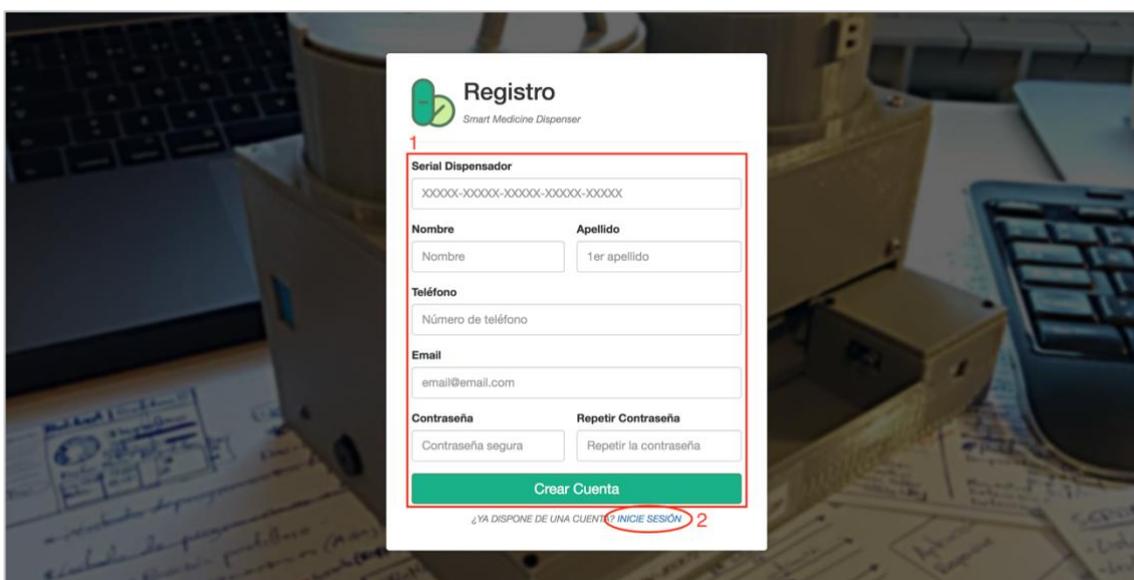


Figura 7.13. Manual de la aplicación web – Registro

En la imagen anterior se muestra la pantalla de registro de la aplicación web. Las funcionalidades de esta pantalla son:

1. **Formulario de registro:** en el formulario, el usuario debe introducir el serial de su dispensador (código alfanumérico situado en el frontal del dispensador), nombre, apellido, número de teléfono, correo electrónico y contraseña.
2. Enlace de acceso a la pantalla de **inicio de sesión**.

El email introducido para el registro servirá para iniciar sesión en la aplicación junto a la contraseña introducida y como email de notificaciones de alarmas del dispensador.

El código de serial será comprobado contra el sistema, mostrando un mensaje de error en el caso de no existir. Cabe recordar que solo se permite el registro de hasta 2 usuarios como máximo por serial de dispensador, mostrando un mensaje de error en el registro en el caso de que ya existan dos usuarios registrados con el serial introducido.

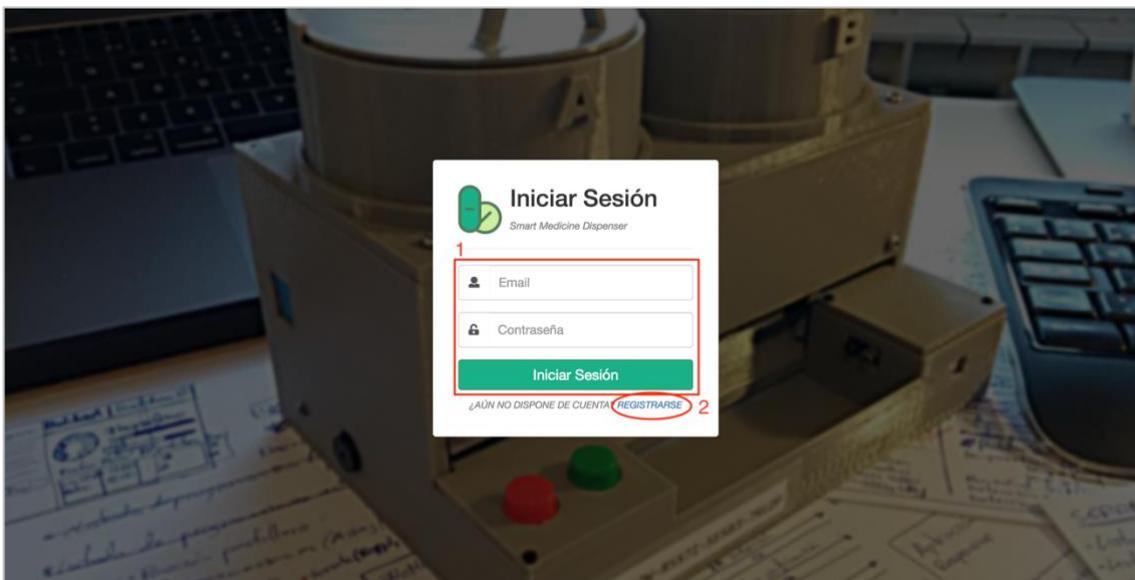


Figura 7.14. Manual de la aplicación web – Inicio de Sesión

La pantalla de inicio de la aplicación muestra:

1. **Formulario para identificarse en el sistema:** el usuario debe introducir el email y la contraseña usados para registrarse en la aplicación.
2. Enlace de acceso a la pantalla de **registro**.

El usuario que puede identificarse en la pantalla puede ser de dos tipos:

- **Usuario cliente:** es el usuario que se registró en el sistema introduciendo para ello el serial de su dispensador.
- **Usuario administrador:** usuario contenido en el sistema y marcado con privilegios de administración.

En función del rol del usuario identificado, el sistema redirigirá al usuario a la pantalla de inicio del *dashboard* cliente o al *dashboard* de administración. En las siguientes secciones se divide el manual en función del rol del usuario identificado.

7.3.1.1 Manual de la aplicación del usuario cliente

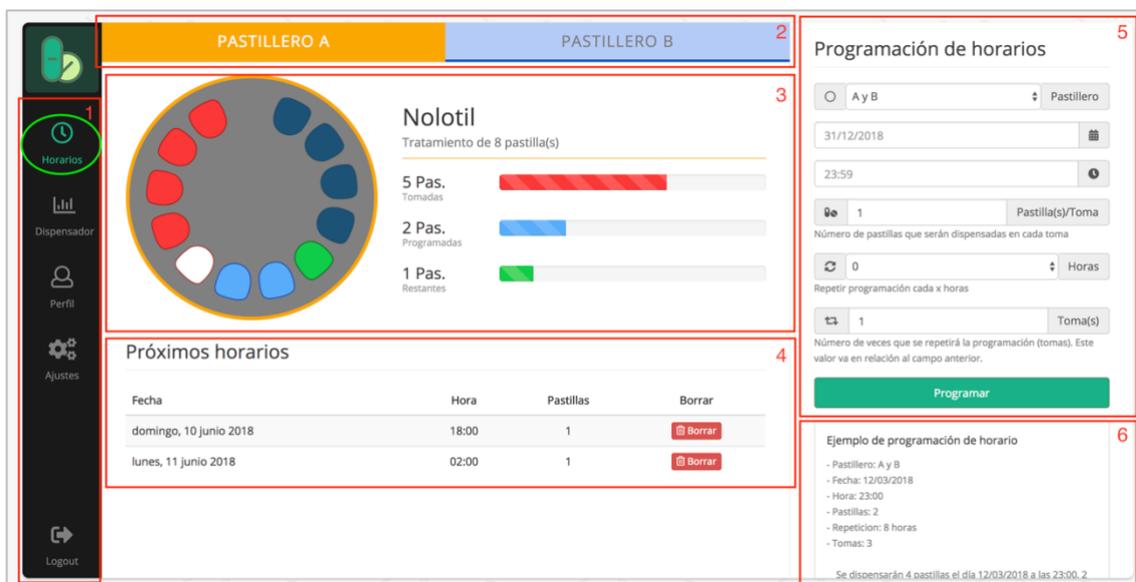


Figura 7.15. Manual de la aplicación web – Gestión de horarios y dispensador

La imagen anterior muestra la pantalla de inicio al identificarse en el sistema como usuario cliente. Esta pantalla se compone de los siguientes elementos:

1. **Barra de navegación:** en verde, la página de la aplicación seleccionada.
2. **Selector de pastillero:** el estado del pastillero (3) y los próximos horarios (4) van en relación al pastillero seleccionado.
3. **Estado del pastillero** seleccionado: en rojo se muestran las posiciones dispensadas, en blanco la posición actual del pastillero, en azul claro las posiciones programadas, en verde las restantes y en azul oscuro las no disponibles. El número de pastillas de las que dispone va directamente relacionado al número de pastillas del tratamiento seleccionado para el pastillero.
4. **Próximos horarios programados:** listado con los detalles de los próximos horarios programados de toma de medicación con la opción de su eliminación.
5. **Formulario de programación de horarios:** a través de este formulario se pueden programar uno o múltiples horarios, para un pastillero o para ambos.
6. **Ejemplo de programación de horarios:** donde se muestra un ejemplo de programación completo para la opción más complicada, programar múltiples horarios.



Figura 7.16. Manual de la aplicación web – Gestión de horarios y dispensador – Móvil

La imagen anterior muestra la pantalla de *horarios* desde un dispositivo móvil. La barra de navegación (1) se posiciona en la parte inferior del dispositivo y el formulario de programación de horarios se encuentra accesible a través del botón (2).



Figura 7.17. Manual de la aplicación web – Dispensador

Pulsando en la sección *Dispensador* de la barra de navegación se accede a la siguiente información:

1. **Datos en tiempo real del dispensador:** muestra la temperatura, la humedad, el índice de calor, el estado de conexión del dispensador y de detección de gas y golpes. Los tres últimos datos se bordean en verde o en rojo en función de si es correcto el valor o presenta algún problema, respectivamente.
2. **Distribución horaria del tiempo de vida del dispensador:** muestra la distribución horaria de las últimas 24 horas de las conexiones del dispensador agrupadas por bloques de 10 minutos. El verde denota conexiones correctas y el rojo desconexión

del dispensador. En función del número de conexiones, el color del bloque correspondiente será un color intermedio entre el rojo y el verde.

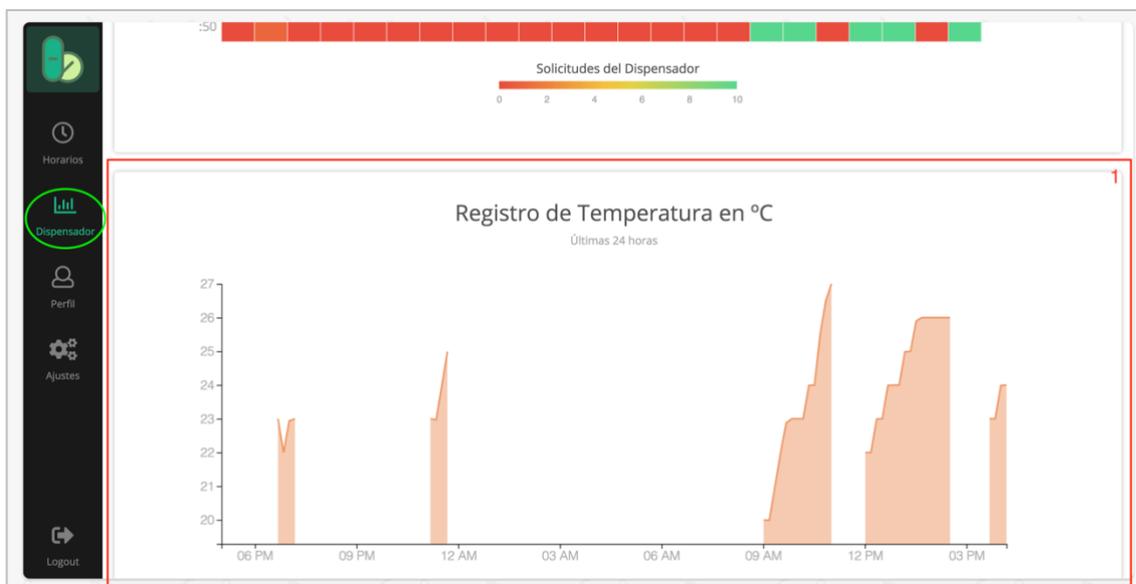


Figura 7.18. Manual de la aplicación web – Dispensador – Temperatura

La imagen anterior muestra la segunda parte de la pantalla *Dispensador* (*accesible por scroll*). Donde se visualiza la distribución horaria de las últimas 24 horas de la temperatura del dispensador. Esta visualización va directamente relacionada al gráfico del tiempo de vida de conexiones del dispensador.

The figure shows the 'Datos Personales' (Personal Data) form in the web application. The sidebar is the same as in Figure 7.18, but the 'Perfil' icon is highlighted with a green circle. The form contains the following fields: 'Serial' (NZHH3-DHK2W-8SX7I-SESB3-7RLDP), 'Nombre' (Jose Antonio), 'Apellido' (Cabaneros), 'Teléfono' (657459306), 'Email' (joseacabaneros@gmail.com), 'Nueva Contraseña' (Contraseña segura), and 'Repetir Contraseña' (Contraseña segura). A green button at the bottom is labeled 'Actualizar Datos Personales'. A red box highlights the entire form area, and a small '1' is in the top right corner of the form frame.

Figura 7.19. Manual de la aplicación web – Datos personales

Navegando hasta la pantalla de *Perfil*, el usuario puede modificar sus datos personales introducidos durante el registro a través del formulario (1). Pudiendo modificar el nombre, apellido, teléfono, email y contraseña.

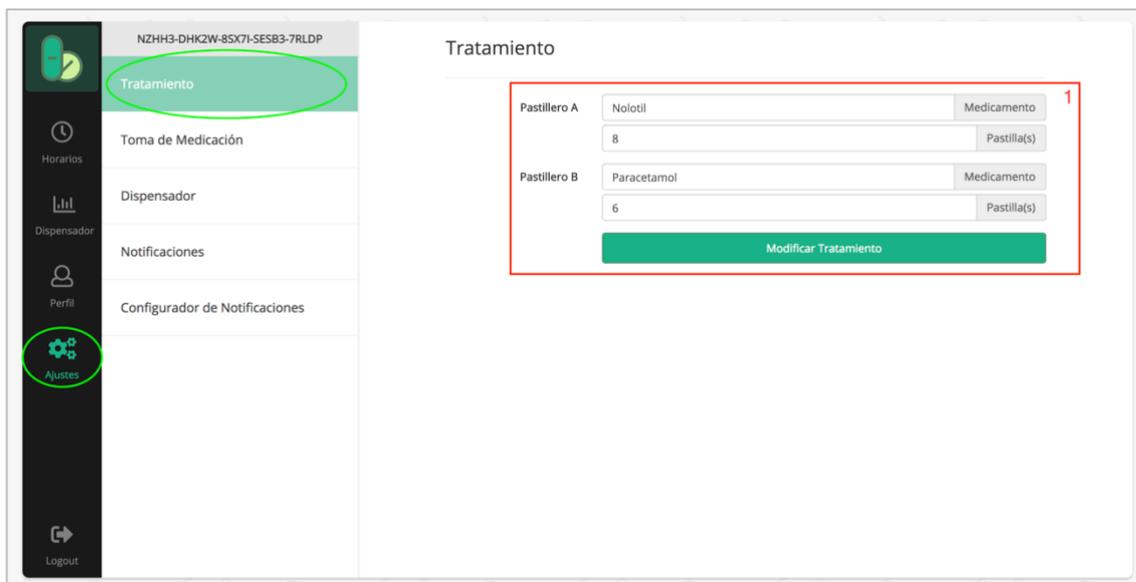


Figura 7.20. Manual de la aplicación web – Ajustes – Tratamiento

En la sección de *Ajustes* disponible en la barra de navegación, el usuario puede seleccionar entre las 5 opciones de configuración disponibles. Seleccionado la opción de configuración de *Tratamiento*, el usuario puede modificar, a través del formulario (1), el nombre y el número de pastillas del tratamiento de cada uno de los pastilleros. Esta acción solo es posible cuando el pastillero correspondiente no disponga de horarios programados y se encuentre en la posición de inicio.

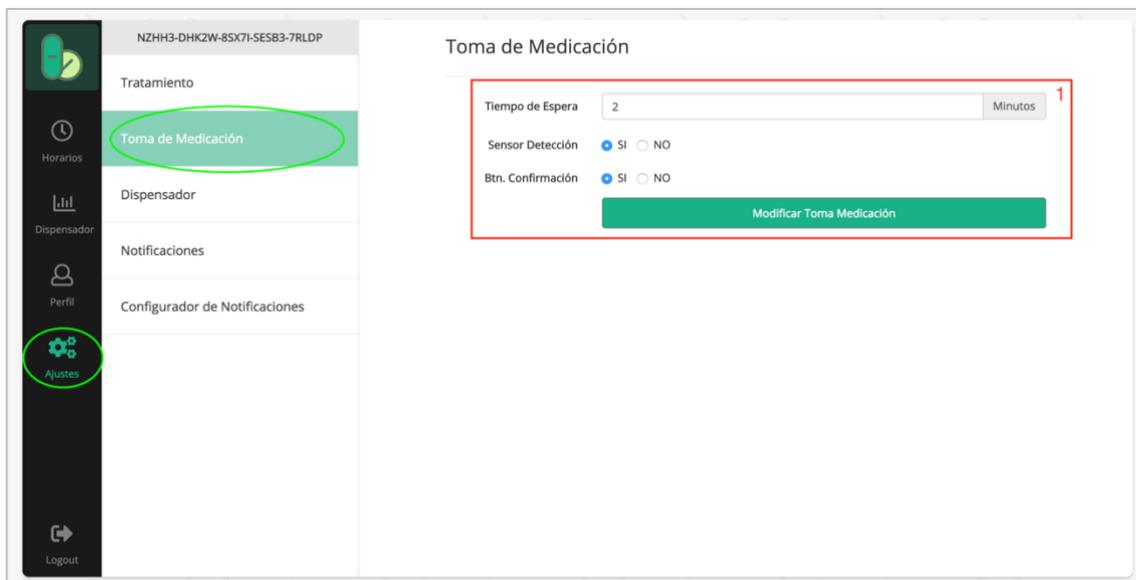


Figura 7.21. Manual de la aplicación web – Ajustes – Toma de medicación

En la opción *Toma de medicación*, el usuario puede modificar a través del formulario (1) como y cuando se considera tomada la medicación dispensada. Esto incluye, el tiempo de espera desde que se dispensan las pastillas hasta que se considera no tomada la medicación y como se debe confirmar la toma en el pastillero. Esto es, a través de la detección de la recogida de las pastillas y/o confirmando la toma a través del botón verde del dispensador.

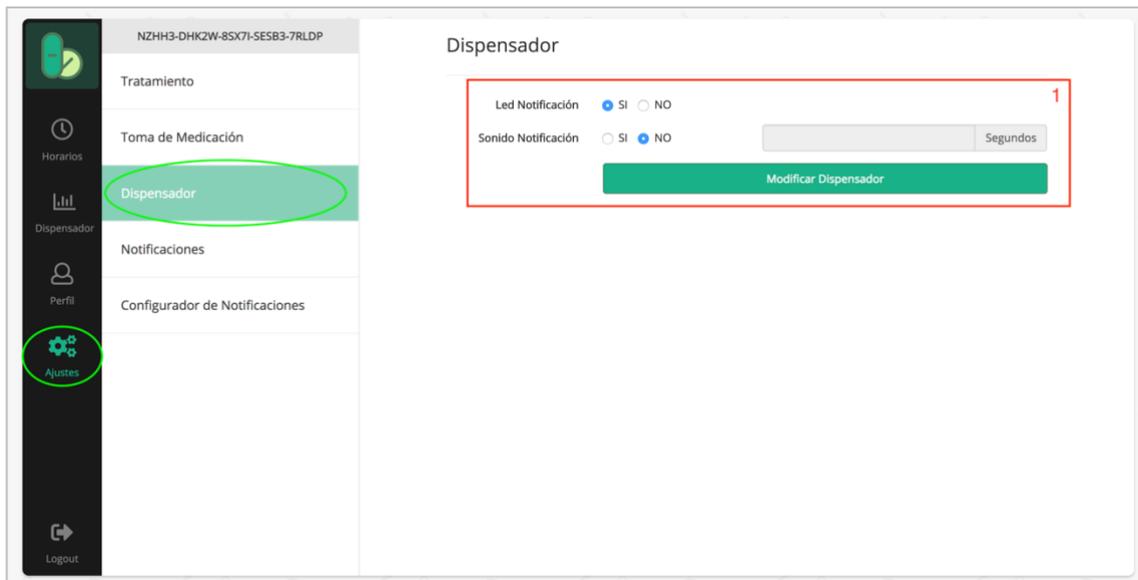


Figura 7.22. Manual de la aplicación web – Ajustes – Dispensador

La opción de configuración *Dispensador* permite configurar la forma en la que se emiten las notificaciones de toma de medicación en el dispensador. Las opciones son mediante el led de notificación y/o emitiendo sonido (1). La emisión del sonido de notificación permite configurar el número de segundos que permanecerá la notificación sonora.

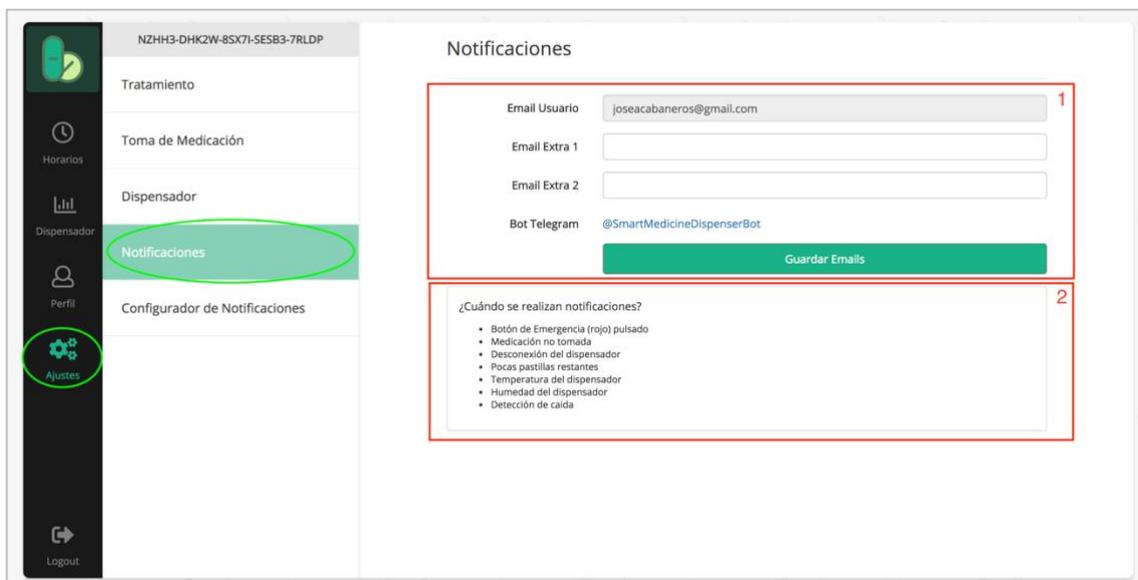


Figura 7.23. Manual de la aplicación web – Ajustes – Notificaciones

La configuración de *Notificaciones* muestra los emails de los usuarios registrados con el serial del dispensador (hasta dos emails correspondientes a dos usuarios registrados). El usuario puede añadir hasta dos emails extra de notificación de alarmas (1). También se muestra un enlace al bot de Telegram del sistema que permite al usuario suscribirse a los mensajes de su dispensador siguiendo los pasos indicados por el bot. Por último, se muestra un panel de resumen con las notificaciones (2) que se realizarán a los emails indicados en esta sección y a los usuarios suscritos al canal de Telegram.

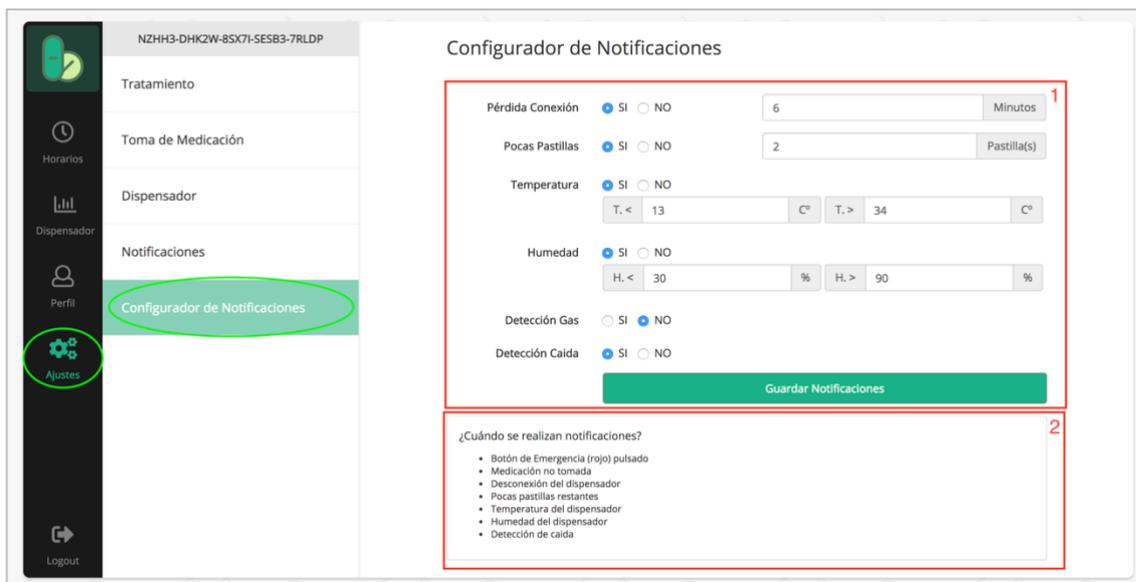


Figura 7.24. Manual de la aplicación web – Ajustes - Toma de medicación

La opción *Configurador de Notificaciones* permite especificar cuándo se realizarán las notificaciones de alarmas a los emails especificados en la pantalla anterior y a los usuarios suscritos al canal de Telegram.

Las notificaciones que pueden especificarse (1) son:

- **Pérdida de conexión del dispensador:** en el caso de activación de esta notificación, puede especificarse el número de minutos que debe permanecer desconectado el dispensador para que se realicen las notificaciones.
- **Pocas pastillas restantes:** en el caso de activación de esta notificación, puede especificarse el número de pastillas restantes de alguno de los pastilleros a partir del cual se emitirán las notificaciones de pocas pastillas.
- **Anomalía en la temperatura** ambiental del dispensador: en caso de activación de esta notificación, pueden especificarse los valores máximo y mínimo de temperatura a partir de los cuales se enviarán las notificaciones.
- **Anomalía en la humedad** ambiental del dispensador: en caso de activación de esta notificación, pueden especificarse los valores máximo y mínimo de humedad a partir de los cuales se enviarán las notificaciones.
- **Detección de gas** en el ambiente del dispensador.
- **Detección de golpe** o caída del dispensador.

Por último, se muestra un panel de resumen (2) acorde a las notificaciones activadas que permite al usuario ver cuándo se realizarán las notificaciones de alarmas a los emails y al canal de Telegram.

7.3.1.2 Manual de la aplicación del administrador

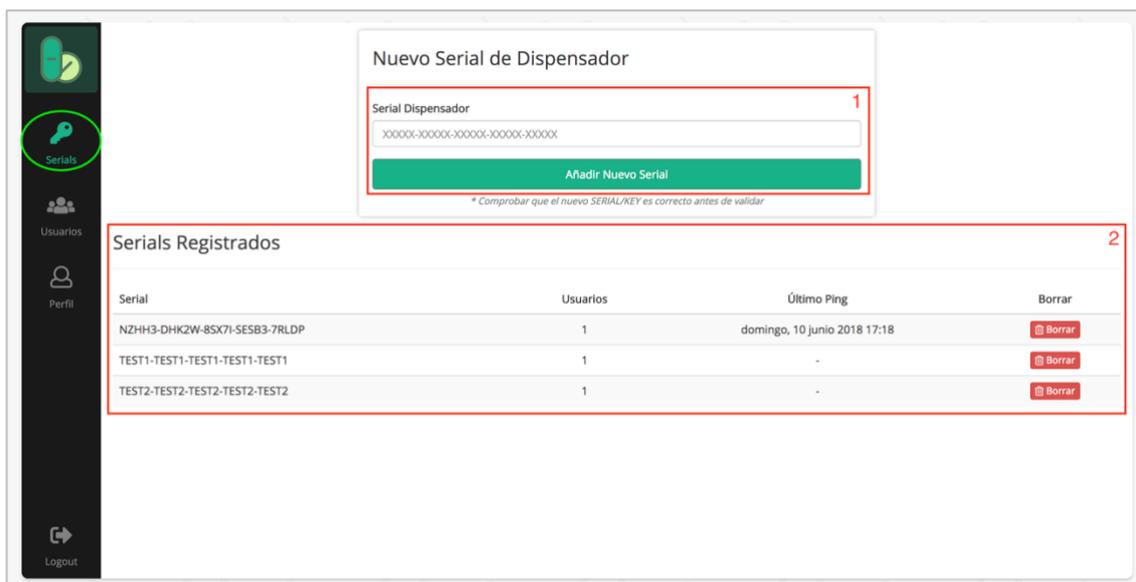


Figura 7.25. Manual de la aplicación web – Serials

La imagen anterior muestra la pantalla de inicio del usuario administrador. Esta es la pantalla de *Serials* en la que el usuario puede añadir nuevos seriales de dispensadores al sistema (1) y consultar los detalles de los ya registrados (2). También se permite la opción de eliminarlos. Cabe destacar que un serial solo puede ser eliminado si no dispone de usuarios registrados con el mismo. En caso contrario se mostrará un mensaje de error en la eliminación.

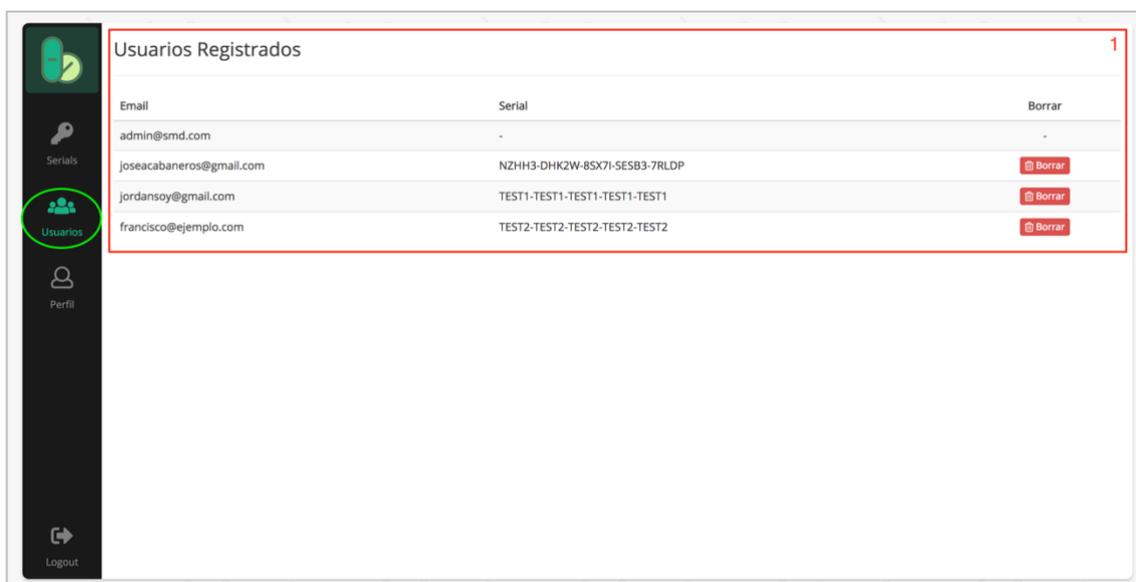


Figura 7.26. Manual de la aplicación web – Usuarios

A través de la pantalla de *Usuarios*, el administrador puede consultar los detalles de los usuarios registrados en el sistema, además de poder eliminarlos (1).

Datos Personales

Serial
-

Nombre
Administrador

Apellido
Administrador

Teléfono
-

Email
admin@smd.com

Nueva Contraseña
Contraseña segura

Repetir Contraseña
Contraseña segura

Actualizar Datos Personales

Figura 7.27. Manual de la aplicación web – Datos personales administrador

En la pantalla de *Perfil*, el usuario administrador puede modificar sus datos personales en el sistema (1). Entre estos datos se encuentran; el nombre, apellido, número de teléfono, email y contraseña de acceso a sistema.

7.3.2 Manual del dispensador de medicamentos

El manual del dispensador de medicamentos servirá como referencia de consulta de los componentes y elementos que contiene el dispensador para abordar las funcionalidades del sistema.

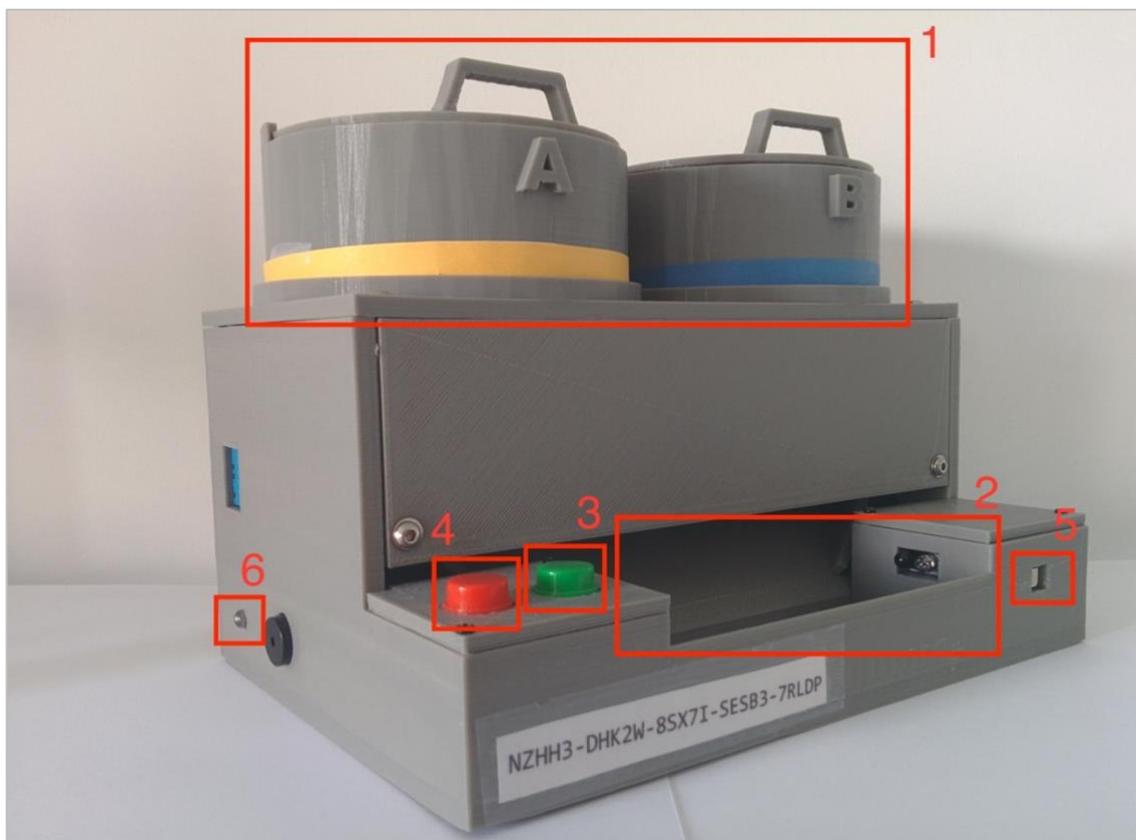


Figura 7.28. Manual del dispensador – Acciones sobre el dispensador

En la imagen anterior se muestra el dispensador de medicamentos. Las acciones sobre los componentes del dispensador que puede realizar el usuario para acometer los objetivos con los que fue creado son:

1. **Pastilleros del dispensador:** retirando la tapa de cada uno de los pastilleros, el usuario accede a las cavidades donde se alojan las pastillas que serán dispensadas. En este punto cabe destacar que cada uno de los pastilleros únicamente debe ser rellenado con un único tipo de medicamento. Una vez rellenos, el usuario debe acceder a la sección de configuración del [Tratamiento](#) para especificar el número de pastillas y el tipo de medicamento que ha introducido en cada uno de los pastilleros.
2. **Lugar de recogida de las pastillas:** en este lugar se dispensarán las pastillas a la fecha y hora de un horario programado y será la posición de donde el usuario las retirará para su toma. Esta cavidad dispone de un sensor que permite saber si las pastillas dispensadas han sido retiradas.
3. **Botón de confirmación:** el botón verde permite al usuario del dispensador confirmar la toma de la medicación.

4. **Botón de emergencia:** el botón rojo permite al usuario enviar notificaciones de emergencia a los emails y canal de Telegram del dispensador.
5. **Led de notificación:** será el LED que notifique al usuario de la toma de la medicación. Este LED permanecerá encendido mientras el usuario no haya realizado las acciones oportunas sobre el dispensador para confirmar la toma (detección de recogida de las pastillas dispensadas y/o confirmación de la toma mediante el botón verde). La confirmación de la toma se especifica en la configuración de la aplicación web [Toma de Medicación](#).
6. **Led de conexión:** este LED permanecerá encendido mientras el dispensador se encuentre conectado a un router WiFi con acceso a internet.

También puede observarse como cada uno de los pastilleros del dispensador se identifica inequívocamente mediante una letra y un color. En la aplicación web se hace referencia a cada uno de los pastilleros a través su letra y color en el dispensador.

Por último, mencionar el serial del dispensador. Puede observarse en la imagen anterior que el serial del dispensador se localiza en la parte frontal inferior.

7.4 Manual del Programador

A pesar de describir en los apartados de esta documentación todos y cada uno de los aspectos de construcción del sistema, se puntualizan de forma resumida ciertos detalles que el programador debe conocer y tener en cuenta a la hora de ampliar o modificar el sistema conectado.

El sistema está compuesto por dos implementaciones diferentes: la implementación del microcontrolador y la del sistema distribuido y aplicación web.

7.4.1 Programación del microcontrolador

El desarrollo de código del microcontrolador se divide en dos ficheros independientes. Un fichero de programación del módulo (ATMega2560) que controla todos los componentes físicos del dispensador y es el encargado de consultar datos sobre los componentes de salida y recibirlos de los componentes de entrada.

El otro fichero sirve de programación para el módulo del wifi (ESP8266) del microcontrolador. Este módulo es el responsable de las peticiones a la API del servidor a través de Internet. Esta última implementación es la que contiene la gran parte de la lógica del microcontrolador. Incluyendo el serial del dispensador, el SSID y contraseña del router WiFi al que debe conectarse y toda la gestión de horarios de toma de medicación.

La comunicación entre ambos módulos se realiza a través del puerto serie. En líneas generales, la comunicación entre ambos módulos se realiza cuando:

- *Del módulo ESP8266 a ATMega2560:* para solicitar datos de los componentes electrónicos de entrada, mover el selector de los pastilleros o realizar alguna acción sobre los componentes de salida.
- *Del módulo ATMega2560 a ESP8266:* para enviar datos recogidos de los componentes de entrada y que previamente ha solicitado el módulo ESP8266 o enviar algún evento ocasionado en el dispensador.

A modo de resumen, el módulo ATMega2560 es el encargado de controlar todos los componentes electrónicos del dispensador conectados al microcontrolador y el módulo ESP8266 se ocupa del control del dispensador, toma de medicamentos y de la comunicación exterior con la API REST del servidor.

Podría decirse que la implementación del microcontrolador sigue una arquitectura maestro-esclavo, donde el módulo ATMega2560 es el esclavo mientras que el módulo ESP8266 actúa como maestro del microcontrolador.

Para modificar o ampliar el código de implementación del microcontrolador es requisito indispensable contar con conocimientos sobre el desarrollo de programas en la plataforma Arduino y haber trabajado anteriormente con programas y componentes más sencillos que permitan conocer las directrices básicas del desarrollo en Arduino.

Por último, el código de desarrollo de ambos módulos del microcontrolador dispone de una gran cantidad de comentarios que detallan la funcionalidad y flujo de ejecución de cada una de las variables y métodos implementados. Esto comentarios serán de gran ayuda para comprender el funcionamiento del dispensador.

7.4.2 Programación del sistema distribuido

La implementación del sistema distribuido se divide en dos funcionalidades principales: la API REST de acceso por parte del microcontrolador del dispensador y la aplicación web de gestión y control.

Para comprender en detalle la implementación, es necesario analizar la estructura del proyecto desarrollado en Node.js.

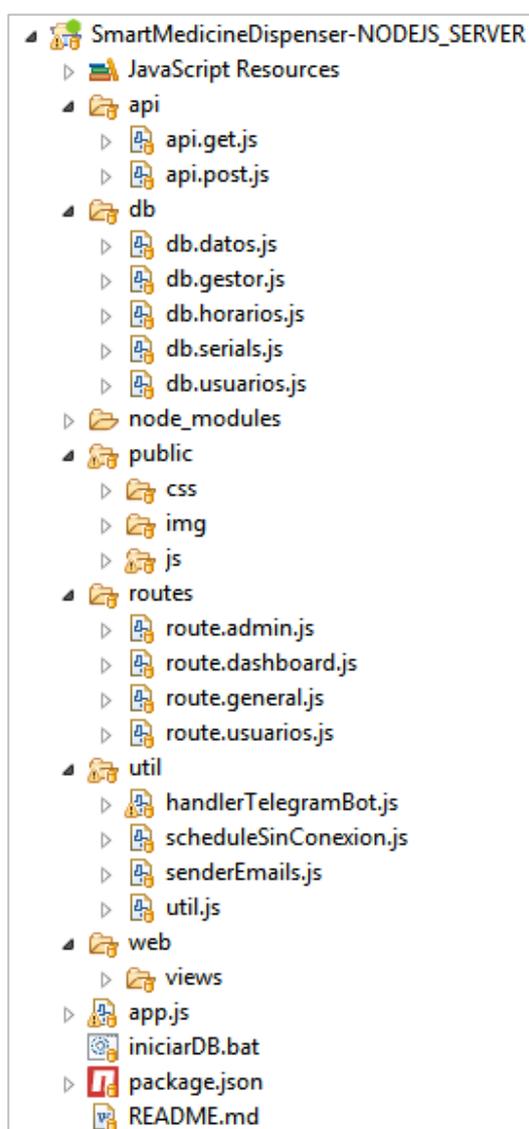


Figura 7.29. Estructura del proyecto Node.js.

El proyecto Node.js de desarrollo del sistema distribuido se organiza en los siguientes módulos:

- **api:** contiene toda la funcionalidad e implementación de la API REST a la que accede el microcontrolador del dispensador. Contiene dos ficheros; uno con la lógica de ejecución a las peticiones GET del dispensador y otro para las peticiones POST.
- **db:** contiene todos los ficheros encargados de la comunicación con la base de datos.
- **public:** los archivos contenidos en esta carpeta serán utilizados para su incorporación en las páginas de la aplicación web para añadir estilos, imágenes y código JS.
- **routes:** en esta carpeta se encuentran todas las rutas de acceso de la aplicación. Tanto peticiones GET de navegación, como peticiones POST de procesamiento de formularios.
- **util:** contiene archivos de utilidad usados en diferentes secciones de la implementación del proyecto.
- **web/views:** esta carpeta contiene todos los ficheros HTML y plantillas usados en el renderizado de las páginas de la aplicación web.

Se recomienda realizar un análisis de cada uno de los módulos que componen el proyecto Node.js antes de ampliar o modificar funcionalidades. El código de implementación dispone de gran cantidad de comentarios que facilitan la comprensión del mismo. También se recomienda encarecidamente seguir la jerarquía de carpetas, módulos, funcionalidades y archivos y el uso de comentarios en el código que faciliten la comprensión de la implementación.

Para ampliar o modificar tanto la implementación del microcontrolador como el desarrollo del sistema distribuido, se recomienda leer y comprender la presente documentación y en especial aquellos capítulos que hacen referencia a los detalles de cada uno de los subsistemas. También se recomienda consultar las [referencias bibliográficas](#) que ayudarán al programador a entender cómo se han implementado ciertas funcionalidades del sistema.

Capítulo 8. Conclusiones y Ampliaciones

8.1 Conclusiones y lecciones aprendidas

El proyecto comenzó con un estudio de viabilidad donde se valoraron los puntos fuertes, complejidad y posibles deficiencias junto al público al que iría dirigido el producto final. Los subsistemas que compondrían el proyecto estaban claros desde un principio: un pastillero capaz de conectarse a internet, un servidor y su base de datos, todo ello debidamente gestionado a través de una aplicación web de gestión del pastillero. Lo que no estaba del todo claro era cómo desarrollar cada uno de los subsistemas.

Las primeras reuniones consistieron en una tormenta de ideas de todo aquello que sería capaz de hacer el sistema conectado. Con prácticamente todos los requisitos a alto nivel definidos, se comenzó a pensar en el diseño del pastillero. El primer enfoque que se quiso dar fue el de un pastillero convencional con cavidades donde se depositarían las pastillas. Más tarde, ese enfoque fue descartado principalmente por el potencial público hacia el que iba dirigido el proyecto, el de las personas de avanzada edad. Se trataba de un enfoque que no encajaba con la máxima de *“no me hagas pensar”*. Después se barajó la posibilidad de construir un tipo de dispensador con pequeños tornillos sin fin que dispensarían uno a uno los medicamentos. Este enfoque no terminó de convencer y finalmente se optó por el enfoque presentado: una *ruleta* con cavidades y soporte móvil que sería capaz de dispensar una a una las pastillas.

En cuanto al controlador del dispensador, se decidió usar un microcontrolador nuevo en el mercado con un método de programación un tanto peculiar. Todo apuntaba a que sería la parte más compleja del proyecto. Los inicios en la programación de este nuevo microcontrolador fueron complicados debido a la escasa documentación y a la complejidad de su programación por módulos. Después de muchas pruebas con todo tipo de componentes y conexiones con servidores de prueba, se logró alcanzar el funcionamiento esperado de este microcontrolador. A partir de ese punto, la programación del controlador se reduciría a ampliar la funcionalidad del mismo, ya que la arquitectura de programación ya había sido definida durante las pruebas previas.

El diseño del dispensador fue realizado simultáneamente a la construcción del sistema electrónico de todos los componentes, sensores y microcontrolador, programación del controlador, implementación del servidor y el desarrollo de la aplicación web. Poco a poco fueron imprimiéndose en 3D cada una de las partes y completándose funcionalidades del sistema conectado.

Se realizaron varias revisiones del desarrollo del proyecto para analizar dificultades encontradas y un seguimiento continuo de la evolución del mismo. Siempre se intentaron fijar

plazos e hitos intermedios en los que se debería tener cierta funcionalidad implementada para evitar desviaciones en la planificación del desarrollo del proyecto.

Finalmente, y tras varios meses de trabajo, se ha conseguido, no solo desarrollar todas las funcionalidades que habían sido fijadas antes del inicio del proyecto, sino que durante el desarrollo del mismo fueron añadiéndose más y más hasta conseguir un sistema conectado verdaderamente completo y funcional, fácil de usar y que puede ser de gran utilidad en el día a día de las personas a las que se dirige.

Con todo esto, el balance final del proyecto no podría ser más satisfactorio. Se ha desarrollado un proyecto multidisciplinar que incluye diseño e impresión 3D, componentes hardware con su correspondiente programación, un servidor y una base de datos de soporte a todo el sistema y una aplicación web funcional, usable y adaptada a todo tipo de dispositivos y pantallas. En definitiva, un sistema inteligente cuyos componentes trabajan de forma sincronizada y cuya batuta lleva el servidor desarrollado.

Para el desarrollo de todas estas disciplinas requeridas en el proyecto se ha profundizado en varias tecnologías ya conocidas, el aprendizaje de otras muchas tanto tecnológicas, disciplinarias como el diseño 3D y dirección, organización y metodologías de desarrollo. Todo ello conforma una agrupación de conocimientos adquiridos y ampliados no solo en el ámbito tecnológico, disciplinas y directrices de ejecución de un proyecto, sino también en forma de capacidad de superación y autorrealización personal.

8.2 Ampliaciones

En este apartado se exponen posibles ampliaciones que podrían realizarse en el sistema conectado. Algunas están orientadas a ampliar las funcionalidades del prototipo de dispensador mientras que otras han surgido como ampliaciones de la funcionalidad desarrollada a lo largo del proyecto.

8.2.1 Pastilleros o ruletas como módulos adaptables del dispensador

El dispensador de medicamentos ha sido diseñado con dos ruletas o pastilleros que permiten dispensar hasta dos tipos de medicamentos diferentes. A pesar de esto, el sistema completo ha sido implementado de tal forma que permite ampliar el número de pastilleros que pueden ser controlados por el sistema de una manera sencilla.

Lo que propone esta ampliación es realizar las adaptaciones necesarias en el diseño del dispensador con el objetivo de soportar la ampliación del número de pastilleros disponibles y por consecuencia, el número de medicamentos diferentes que pueda dispensar.

El dispensador se convertiría en un sistema de módulos adaptables. Cada módulo tendría dos pastilleros y permitiría dispensar dos tipos más de medicamentos diferentes. Estos módulos se añadirían al dispensador de una forma sencilla mediante conectores fijados estratégicamente tanto en el dispensador como en los módulos de pastilleros.

Esta ampliación está enfocada al uso del dispensador en centros como residencias u hospitales, donde el número de medicamentos distintos de los que disponen es considerablemente mayor. Este sistema de módulos haría más sencilla la labor de toma de medicamentos a los pacientes además de automatizar todo proceso.

8.2.2 Pastillas disponibles en cada ruleta para ser dispensadas

Cada una de las *ruletas* o pastilleros permite alojar hasta 12 pastillas en su interior. A pesar de no ser un número excesivamente bajo, en ocasiones podría ser insuficiente dependiendo de las necesidades de cada usuario.

Para esta ampliación sería necesario rediseñar el enfoque de *ruleta* como dispensador para que permita alojar en su interior un gran número de pastillas, probablemente con algún tipo de contenedor donde depositar todos los medicamentos a dispensar del mismo tipo.

El reto estaría en cómo controlar que las pastillas se dispensaran de una en una, debido a que las dimensiones de las pastillas disponibles en el mercado son muy dispares y resultaría imposible crear un único selector que dispensara una única pastilla muy pequeña y no se bloqueara con una muy grande.

Esta ampliación también presenta un inconveniente importante. Las pastillas, una vez fuera de su envoltorio, se degradan en poco tiempo, por lo que no sería muy conveniente rellenar con muchas pastillas el contenedor si van a permanecer en él mucho tiempo.

8.2.3 Sensores y componentes del dispensador

El número de sensores y componentes disponibles en el mercado que podrían resultar de utilidad en el dispensador es verdaderamente amplio. Estos podrían ampliar el número de valores ambientales medidos actualmente además de añadir otras funcionalidades.

Un ejemplo sería la incorporación de un relé que permita controlar algún componente electrónico del hogar como puede ser una televisión. Esta estaría conectada al dispensador y sería este el encargado de controlar la alimentación de la televisión. Cuando sea necesario tomar la mediación, el dispensador podría cortar la corriente de la televisión hasta que la medicación fuera tomada. Obligando al usuario a tomar las pastillas si quiere seguir disfrutando del programa.

Otro ejemplo sería desarrollar un pequeño dispositivo que el usuario pueda llevar cuando salga de casa. Este dispositivo le avisaría mediante sonidos y vibraciones cuando lo toque tomar la medicación y se encuentre fuera de casa. Al llegar de nuevo, el dispositivo notificaría al dispensador de su llegada y dispensaría aquellos medicamentos que no fueron tomados por estar fuera de casa.

8.2.4 Conexión WiFi del dispensador

Actualmente, el SSID y la contraseña del router al que debe conectarse el microcontrolador del dispensador se encuentran incluidos en el propio código de programación.

Lo que propone esta ampliación es incorporar un lector de tarjetas al dispensador. Esta tarjeta dispondría en su interior de un fichero de configuración donde se especificaría el SSID y la contraseña del router al que debe conectarse. El usuario adaptaría el fichero de la tarjeta desde un ordenador con los datos de su router WiFi e insertaría de nuevo la tarjeta en el lector del dispensador. Al iniciar el dispensador, el microcontrolador leería los valores de la tarjeta e inicia el proceso de conexión al router WiFi indicado.

Con esto se conseguiría configurar la conexión WiFi del dispensador sin necesidad de recompilar el código de programación del microcontrolador y subir la nueva versión.

8.2.5 Funcionalidades de la aplicación y del servidor

La mayor parte de las ampliaciones anteriores supondrían un cambio tanto en el diseño como en la funcionalidad de la aplicación web para soportar las nuevas funciones disponibles en el dispensador.

Las funcionalidades de la aplicación web y del servidor también podrían ser ampliadas para soportar un mayor número de configuraciones del dispensador o la incorporación de nuevas visualizaciones en D3 a la sección [dispensador](#).

Y por supuesto, el diseño de la aplicación también podría ser mejorado tanto en términos de usabilidad, estéticos y experiencia de usuario. Añadiendo nuevos componentes, animaciones y diseños. Probablemente haciendo uso de alguno de los frameworks o librerías frontend descritas en el [estudio de alternativas](#).

Capítulo 9. Planificación del Proyecto y Presupuesto

En este capítulo se presenta la planificación seguida para la ejecución del proyecto donde se incluye el listado de tareas, el diagrama de Gantt y un resumen de la planificación. En esta sección del documento también se incluye un presupuesto detallado del proyecto desglosado por los diferentes conceptos que acontecen el mismo.

9.1 Planificación del Proyecto

En la planificación del proyecto se presentan los siguientes resúmenes que conforman dicha planificación:

- **Listado de Tareas:** muestra un listado resumen de las principales tareas que componen la planificación del proyecto
- **Diagrama de Gantt:** presenta de forma gráfica el tiempo de dedicación previsto para las tareas resumen del listado anterior a lo largo del tiempo total previsto para la ejecución del proyecto.
- **Resumen de la Planificación:** recoge, a modo de resumen, los aspectos y valores de las variables más significativas de la planificación.

Es importante destacar que todos los resúmenes presentados en esta sección están orientados a conocer cuáles son las principales tareas que serán desarrolladas en el proyecto y el tiempo de dedicación a cada una de ellas. Por este motivo, se han contraído tareas consideradas menos importantes, dejando solamente las tareas resumen y aquellas consideradas fundamentales para la ejecución del proyecto.

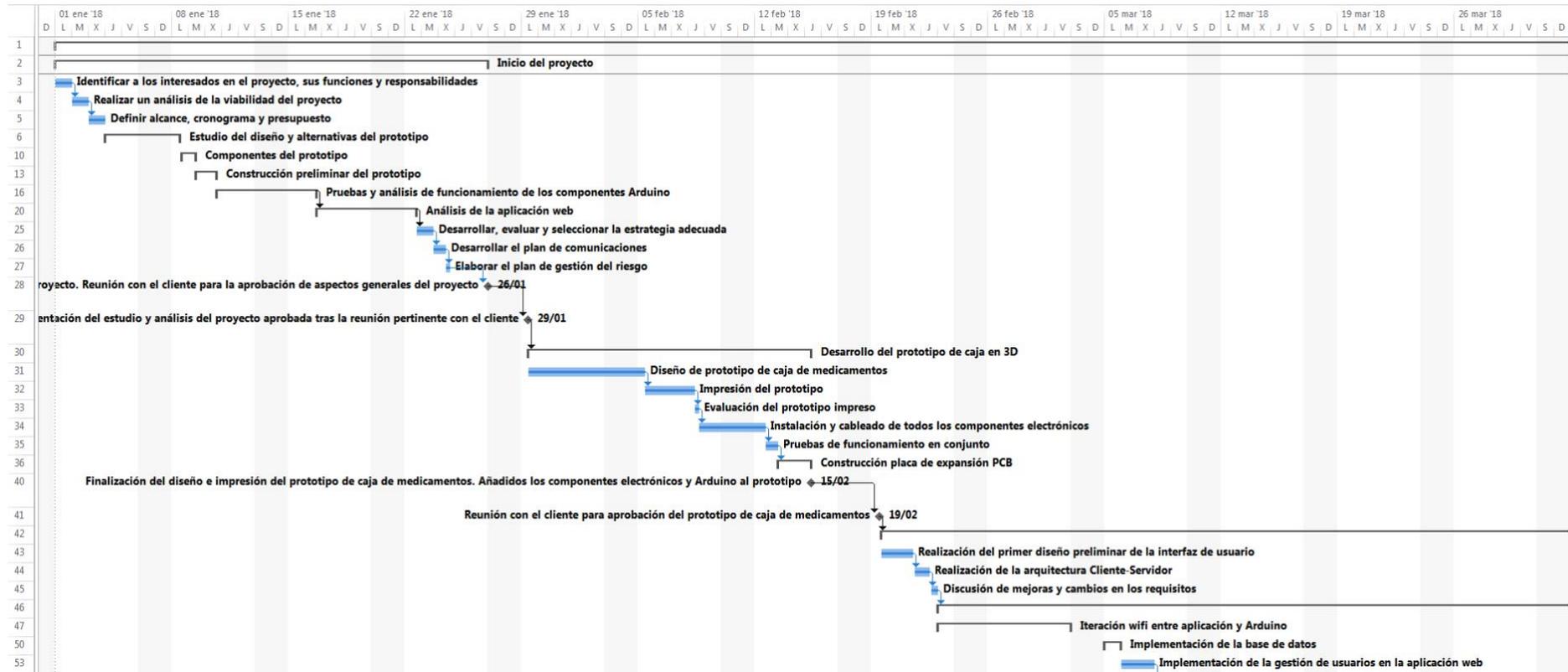
Para conocer con mayor detalle cada una de las tareas que componen la planificación del proyecto, se proporciona un fichero con la planificación completa en formato Microsoft Office Project 2013.

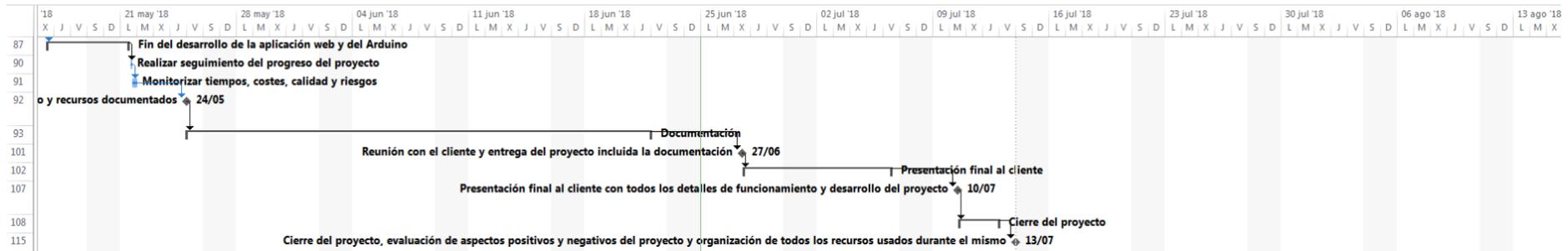
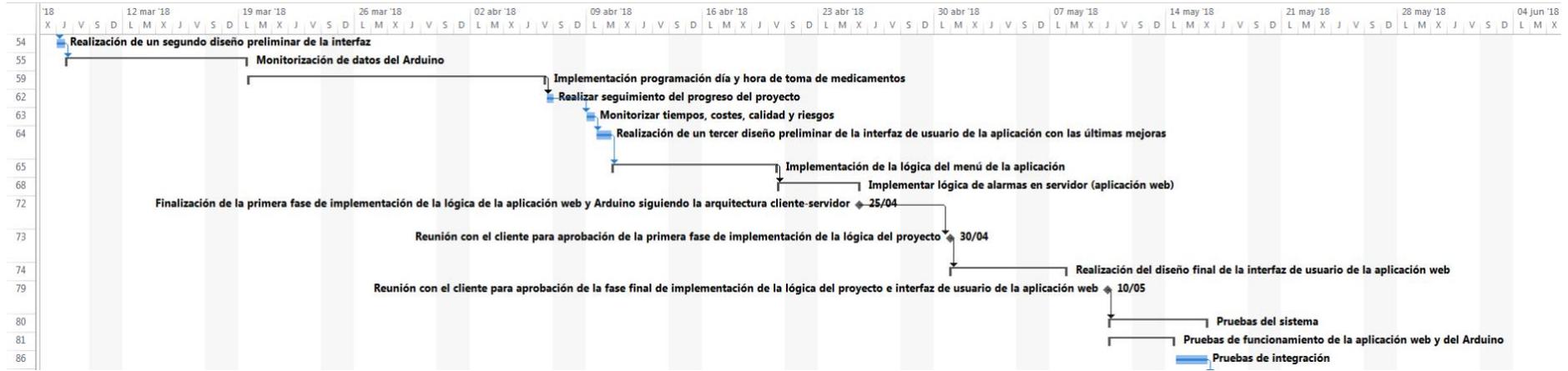
9.1.1 Listado de tareas

		Modo de	Nombre de tarea	Duración	Comienzo	Fin	Predecesora	Nombres de los recursos
1			DISPENSADOR INTELIGENTE DE MEDICAMENTOS CONECTADO Y SISTEMA DE GESTIÓN	140 días	lun 01/01/18	vie 13/07/18		
2	✓		Inicio del proyecto	20 días	lun 01/01/18	vie 26/01/18		
3	✓		Identificar a los interesados en el proyecto, sus funciones y responsabilidades	1 día	lun 01/01/18	lun 01/01/18		Desarrollador - Autor TFM;Ordenador trabajo[1]
4	✓		Realizar un análisis de la viabilidad del proyecto	1 día	mar 02/01/18	mar 02/01/18	3	Desarrollador - Autor TFM;Ordenador trabajo[1]
5	✓		Definir alcance, cronograma y presupuesto	1 día	mié 03/01/18	mié 03/01/18	4	Desarrollador - Autor TFM;Ordenador trabajo[1]
6	✓		▸ Estudio del diseño y alternativas del prototipo	2,5 días	jue 04/01/18	lun 08/01/18		
10	✓		▸ Componentes del prototipo	0,75 días	lun 08/01/18	mar 09/01/18	9	
13	✓		▸ Construcción preliminar del prototipo	1,5 días	mar 09/01/18	mié 10/01/18	12	
16	✓		▸ Pruebas y análisis de funcionamiento de los componentes Arduino	4 días	mié 10/01/18	mar 16/01/18	15	
20	✓		▸ Análisis de la aplicación web	4 días	mar 16/01/18	lun 22/01/18	16	
25	✓		Desarrollar, evaluar y seleccionar la estrategia adecuada	1 día	lun 22/01/18	mar 23/01/18	20	Desarrollador - Autor TFM;Ordenador trabajo[1]
26	✓		Desarrollar el plan de comunicaciones	4 horas	mar 23/01/18	mié 24/01/18	25	Desarrollador - Autor TFM;Ordenador trabajo[1]
27	✓		Elaborar el plan de gestión del riesgo	4 horas	mié 24/01/18	mié 24/01/18	26	Desarrollador - Autor TFM;Ordenador trabajo[1]
28	✓		Estudio y análisis de aspectos generales del proyecto. Reunión con el cliente para la aprobación de aspectos generales del proyecto	2 horas	vie 26/01/18	vie 26/01/18	27FC+2 días	Desarrollador - Autor TFM;Ordenador trabajo[1]
29	✓		Entrega de la documentación del estudio y análisis del proyecto aprobada tras la reunión pertinente con el cliente	1 hora	lun 29/01/18	lun 29/01/18	28	Desarrollador - Autor TFM;Ordenador trabajo[1]
30	✓		Desarrollo del prototipo de caja en 3D	13 días	lun 29/01/18	jue 15/02/18	29	
31	✓		Diseño de prototipo de caja de medicamentos	5 días	lun 29/01/18	lun 05/02/18		Desarrollador - Autor TFM;Ordenador trabajo[2]
32	✓		Impresión del prototipo	3 días	lun 05/02/18	jue 08/02/18	31	Impresora 3D[1];Desarrollador - Autor TFM;Orde
33	✓		Evaluación del prototipo impreso	4 horas	jue 08/02/18	jue 08/02/18	32	Desarrollador - Autor TFM;Ordenador trabajo[1]
34	✓		Instalación y cableado de todos los componentes electrónicos	2 días	jue 08/02/18	lun 12/02/18	33	Desarrollador - Autor TFM;Ordenador trabajo[1]
35	✓		Pruebas de funcionamiento en conjunto	4 horas	lun 12/02/18	mar 13/02/18	34	Desarrollador - Autor TFM;Ordenador trabajo[1]
36	✓		▸ Construcción placa de expansión PCB	2 días	mar 13/02/18	jue 15/02/18	35	
40	✓		Finalización del diseño e impresión del prototipo de caja de medicamentos. Añadidos los componentes electrónicos y Arduino al prototipo	0 horas	jue 15/02/18	jue 15/02/18	39	Desarrollador - Autor TFM
41	✓		Reunión con el cliente para aprobación del prototipo de caja de medicamentos	3 horas	lun 19/02/18	lun 19/02/18	40FC+2 días	Desarrollador - Autor TFM;Ordenador trabajo[1]
42	✓		Desarrollo de la aplicación web y del código microcontrolador	65,5 días	lun 19/02/18	lun 21/05/18	41	
43	✓		Realización del primer diseño preliminar de la interfaz de usuario	2 días	lun 19/02/18	mié 21/02/18		Desarrollador - Autor TFM;Ordenador trabajo[1]
44	✓		Realización de la arquitectura Cliente-Servidor	1 día	mié 21/02/18	jue 22/02/18	43	Desarrollador - Autor TFM;Ordenador trabajo[1]
45	✓		Discusión de mejoras y cambios en los requisitos	4 horas	jue 22/02/18	jue 22/02/18	44	Desarrollador - Autor TFM;Ordenador trabajo[1]
46	✓		Desarrollo de la funcionalidad de la aplicación web y del Arduino	43,5 días	vie 23/02/18	mié 25/04/18	45	
47	✓		▸ Iteración wifi entre aplicación y Arduino	6 días	vie 23/02/18	vie 02/03/18		
50	✓		▸ Implementación de la base de datos	1 día	lun 05/03/18	lun 05/03/18	49	
53	✓		Implementación de la gestión de usuarios en la aplicación web	2 días	mar 06/03/18	mié 07/03/18	52	Desarrollador - Autor TFM;Ordenador trabajo[1]

		Modo de	Nombre de tarea	Duración	Comienzo	Fin	Predecesora	Nombres de los recursos
54	✓		Realización de un segundo diseño preliminar de la interfaz	4 horas	jue 08/03/18	jue 08/03/18	53	Desarrollador - Autor TFM;Ordenador trabajo[1]
55	✓		▸ Monitorización de datos del Arduino	7 días	jue 08/03/18	lun 19/03/18	54	
59	✓		▸ Implementación programación día y hora de toma de medicamentos	14 días	lun 19/03/18	vie 06/04/18	58	
62	✓		Realizar seguimiento del progreso del proyecto	4 horas	vie 06/04/18	vie 06/04/18	59	Desarrollador - Autor TFM;Ordenador trabajo[1]
63	✓		Monitorizar tiempos, costes, calidad y riesgos	4 horas	lun 09/04/18	lun 09/04/18	62	Desarrollador - Autor TFM;Ordenador trabajo[1]
64	✓		Realización de un tercer diseño preliminar de la interfaz de usuario de la aplicación con las últimas mejoras	1 día	lun 09/04/18	mar 10/04/18	63	Desarrollador - Autor TFM;Ordenador trabajo[1]
65	✓		▸ Implementación de la lógica del menú de la aplicación	8 días	mar 10/04/18	vie 20/04/18	64	
68	✓		▸ Implementar lógica de alarmas en servidor (aplicación web)	3 días	vie 20/04/18	mié 25/04/18	65	
72	✓		Finalización de la primera fase de implementación de la lógica de la aplicación web y Arduino siguiendo la arquitectura cliente-servidor	0 horas	mié 25/04/18	mié 25/04/18	71	Desarrollador - Autor TFM
73	✓		Reunión con el cliente para aprobación de la primera fase de implementación de la lógica del proyecto	4 horas	lun 30/04/18	lun 30/04/18	72FC+3 días	Desarrollador - Autor TFM;Ordenador trabajo[1]
74	✓		▸ Realización del diseño final de la interfaz de usuario de la aplicación web	5 días	mar 01/05/18	lun 07/05/18	73	
79	✓		Reunión con el cliente para aprobación de la fase final de implementación de la lógica del proyecto e interfaz de usuario de la aplicación web	4 horas	jue 10/05/18	jue 10/05/18	78FC+2 días	Desarrollador - Autor TFM;Ordenador trabajo[1]
80	✓		▸ Pruebas del sistema	4 días	jue 10/05/18	mié 16/05/18	79	
81	✓		▸ Pruebas de funcionamiento de la aplicación web y del Arduino	2 días	jue 10/05/18	lun 14/05/18		
86	✓		Pruebas de integración	2 días	lun 14/05/18	mié 16/05/18	85	Desarrollador - Autor TFM;Ordenador trabajo[1]
87	✓		▸ Fin del desarrollo de la aplicación web y del Arduino	3 días	mié 16/05/18	lun 21/05/18	86	
90	✓		Realizar seguimiento del progreso del proyecto	2 horas	lun 21/05/18	lun 21/05/18	87	Desarrollador - Autor TFM;Ordenador trabajo[1]
91	✓		Monitorizar tiempos, costes, calidad y riesgos	2 horas	lun 21/05/18	lun 21/05/18	90	Desarrollador - Autor TFM;Ordenador trabajo[1]
92	✓		Finalización de sistema; funcionando (lógica e interfaz), testado y código y recursos documentados	0 días	jue 24/05/18	jue 24/05/18	91FC+3 días	Desarrollador - Autor TFM
93	✓		▸ Documentación	20 días	vie 25/05/18	jue 21/06/18	92	
101	✓		<i>Reunión con el cliente y entrega del proyecto incluida la documentación</i>	4 horas	mié 27/06/18	mié 27/06/18	93FC+3 días	Desarrollador - Autor TFM;Ordenador trabajo[1]
102			▸ Presentación final al cliente	7 días	mié 27/06/18	vie 06/07/18	101	
107			Presentación final al cliente con todos los detalles de funcionamiento y desarrollo del proyecto	1 día	lun 09/07/18	mar 10/07/18	102FC+1 día	Desarrollador - Autor TFM;Ordenador trabajo[1]
108			▸ Cierre del proyecto	2,5 días	mar 10/07/18	jue 12/07/18	107	
115			Cierre del proyecto, evaluación de aspectos positivos y negativos del proyecto y organización de todos los recursos usados durante el mismo	4 horas	vie 13/07/18	vie 13/07/18	108FC+0,5 días	Desarrollador - Autor TFM;Ordenador trabajo[1]

9.1.2 Diagrama de Gantt





9.1.3 Resumen de la planificación

Sobre la base de las consideraciones anteriores, se muestra a continuación una serie de aspectos y propiedades resumen que componen la planificación del proyecto.

Duración del proyecto	128 días laborables 128 días x 8 horas = 1024 horas
Tareas	76
Hitos	13
Principales tareas resumen	<ul style="list-style-type: none"> • Inicio del proyecto – 20 días • Construcción del dispensador – 13 días • Desarrollo de la aplicación web y microcontrolador – 65,5 días • Documentación – 20 días • Presentación – 7 días • Cierre del proyecto – 2,5 días
Inicio del proyecto	01/01/18
Fin del proyecto	13/07/18

Las horas de dedicación media han sido prorrateadas a 8 horas por día laboral, pero realmente las horas dedicadas al desarrollo del proyecto han sido una media de 40 horas semanales incluyendo sábados y domingos.

9.2 Presupuesto del Proyecto

El presupuesto del proyecto será dividido en varias secciones que agrupan conceptos relacionados del mismo. A modo de resumen, se agrupan en:

- Recursos materiales para la construcción del dispensador de medicamentos y hardware utilizado.
- Recursos humanos para la ejecución del proyecto (todos los perfiles disponen del mismo coste por hora debido a que el autor del proyecto ha sido el encargado de recrear los distintos perfiles para la ejecución del proyecto).
- Licencias de software necesario y otros conceptos de gastos.

Código Ítem	Concepto	Cantidad	Precio Unitario	Total Concepto
01	HARDWARE			402,00 €
011	Materiales prototipo			101,00 €
0111	Materiales electrónicos			27,00 €
0111.1	ATMega2560 + WIFI + ESP8266 (32Mb)	1	13,00 €	13,00 €
0111.2	Placa de expansión	1	3,00 €	3,00 €
0111.3	Motor paso a paso 28BYJ-48	2	2,00 €	4,00 €
0111.4	Led	2	0,50 €	1,00 €
0111.5	Sensor de detección IR	1	1,00 €	1,00 €
0111.6	Sensor de temperatura y humedad DH11	1	1,00 €	1,00 €
0111.7	Zumbador	1	1,00 €	1,00 €
0111.8	Sensor de gas MQ-2	1	1,00 €	1,00 €
0111.9	Sensor de vibración	1	1,00 €	1,00 €
0111.10	Botones	2	0,50 €	1,00 €
0112	Materiales de construcción			74,00 €
0112.1	PLA material construcción	1	22,00 €	22,00 €
0112.2	Cableado	1	4,00 €	4,00 €
0112.3	Estaño	1	4,00 €	4,00 €
0112.4	Silicona termofusible	1	3,00 €	3,00 €
0112.5	Muelas abrasivas	1	2,00 €	2,00 €
0112.6	Tornillos autoperforantes m2	1	4,00 €	4,00 €
0112.7	Tornillos y tuercas m2	1	5,00 €	5,00 €
0112.8	Otros: cinta aislante, bridas, brocas, etc.	1	10,00 €	10,00 €
0112.9	Amortización de maquinaria	1	20,00 €	20,00 €
012	Amortización Equipos	3	55,50 €	166,50 €
013	Amortización Impresora 3D	1	45,50 €	45,50 €
014	Servidor y base de datos	1	89,00 €	89,00 €
02	RECURSOS HUMANOS			26.173,00 €
02.1	Director de Proyecto (Horas)	30	50,00 €	1.150,00 €
02.2	Constructor del Prototipo (Horas)	90	25,00 €	2.250,00 €

Dispensador Inteligente de Medicamentos Conectado y Sistema de Gestión

02.3	Analista-Diseñador (Horas)	195	35,00 €	6.825,00 €
02.4	Programador (Horas)	709	22,00 €	15.598,00 €
03	SOFTWARE (amortización de licencias)			366,85 €
03.1	Microsoft Office 2013 + Project + Visio	1	190,95 €	190,95 €
03.2	Windows 10 Home	1	90,00 €	90,00 €
03.3	Adobe Illustrator	1	85,90 €	85,90 €
04	VARIOS			444,50 €
04.1	Material de oficina	1	50,00 €	50,00 €
04.2	Electricidad	5	25,00 €	125,00 €
04.3	Conexión a internet	5	29,90 €	149,50 €
04.4	Desplazamientos	6	20,00 €	120,00 €
SUBTOTAL				27.386,35 €
21% IVA				5.751,13 €
TOTAL				33.137,48 €

Capítulo 10. Referencias Bibliográficas

10.1 Libros y Artículos

[Brown16] Brown, Ethan. “Web Development with Node and Express: Leveraging the JavaScript Stack”. O’Reilly Media. 2016. ISBN 1491949309

[Nayak14] Nayk, Amol. “MongoDB Cookbook”. Packt Publishing. 2014. ISBN 1783987308

[Haviv14] Haviv, Amos Q. “MEAN Web Development”. Packt Publishing. 2014. ISBN 1783983280

[Buyya16] Buyya, Rajkumar; Vahid, Amir. “Internet of Things: Principles and Paradigms”. Morgan Kaufmann. 2016. ISBN 012805395X

[Amariei15] Amariei, Cornel. “Arduino Development Cookbook”. Packt Publishing. 2015. ISBN 1783982942

[Margolis11] Margolis, Michael. “Arduino Cookbook”. O’Reilly Media. 2011. ISBN 1449313876

[Schwartz16] Schwartz, Marco. “Internet of Things with ESP8266”. Packt Publishing. 2016. ISBN 1786468026

10.2 Referencias en Internet

[System Engineering] “Megacontrolador Arduino con ESP8266”
<http://www.sysengineering.ru/blog/электроника-контроллер-arduino-mega-c-esp8266/>

[Arduino] “Interfacing with Hardware”
<https://playground.arduino.cc/Main/InterfacingWithHardware>

[Arduino Json] “JSON library for Arduino” <https://arduinojson.org/>

[RobotDyn] “MEGA+WIFI R3 ATmega2560+ESP8266 Documentation”
<https://robotdyn.com/mega-wifi-r3-atmega2560-esp8266-flash-32mb-usb-ttl-ch340g-micro-usb.html>

[Pieter P.] “A Beginner's Guide to the ESP8266” <https://tttpa.github.io/ESP8266/Chap01%20-%20ESP8266.html>

[42 Bots] “28BYJ-48 Stepper Motor with ULN2003 driver” <http://42bots.com/tutorials/28byj-48-stepper-motor-with-uln2003-driver-and-arduino-uno/>

[Node.js] “Node.js v7.10.1 Documentation” <https://nodejs.org/docs/latest-v7.x/api/>

[NPM] “NPM Documentation” <https://docs.npmjs.com/>

[MongoDB] “MongoDB Docs” <https://docs.mongodb.com/>

[D3] “Data-Driven Documents v5 Documentation” <https://github.com/d3/d3/wiki>

[Swig] “A Node.js and Browser-based JavaScript Template Engine Documentation”
<http://node-swig.github.io/swig-templates/docs/>

[Bootstrap Datpicker] “Bootstrap 3 Datapicker v4 Docs”
<http://eonasdan.github.io/bootstrap-datetimerpicker>

[Telegram] “Telegram APIs” <https://core.telegram.org/api#getting-started>

[Moment.js] “Moment.js Documentation” <https://momentjs.com/docs/>

[Adobe Color] “Adobe Color CC” <https://color.adobe.com/es/create/color-wheel>

[Thingiverse] “Thingiverse pills search”
<https://www.thingiverse.com/search?q=pills&dwh=435b33b1a116acd>

[Tinkercad] “Tinkercad, diseño e impresión 3D” <https://www.tinkercad.com/>

[WebSequenceDiagrams] “Create sequence diagrams online”
<https://www.websequencediagrams.com/>

[yUML] “Create use case diagram” <https://yuml.me/>

Capítulo 11. Apéndices

11.1 Glosario y Diccionario de Datos

En este apartado se encontrarán todas las definiciones de términos ordenados alfabéticamente usados a lo largo de todo el documento que resulten imprescindibles, así como aquellas abreviaturas que resulten interesantes y sean citadas en el mismo:

- **Internet de las Cosas:** *Internet of Things*, abreviado *IoT*, es un concepto que se refiere a la interconexión digital de objetos cotidianos con Internet.
- **Microcontrolador:** Circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria. Está compuesto de varios bloques funcionales, los cuales cumplen una tarea específica.
- **Prototipo de dispensador:** Término usado en el documento para hacer referencia al prototipo de dispensador de medicamentos impreso en 3D.
- **Ruleta o pastillero:** Término utilizado para referirse a cada uno de los dos pastilleros de los que dispone el prototipo. En ocasiones se hace referencia a ellos como dispensadores por independiente, debido a que su funcionalidad es esa, dispensar pastillas contenidas en su interior.
- **Serial:** Secuencia de cinco bloques que contienen cinco caracteres alfanuméricos cada uno de ellos y que identifican inequívocamente un prototipo de dispensador.
- **Sistema conectado:** Composición de todos los subsistemas del proyecto y su comunicación. El sistema inteligente, objeto del proyecto, está formado por el dispensador, el servidor, la base de datos y la aplicación web.
- **Smart Medicine Dispenser:** Término utilizado en la aplicación web para referirse a este proyecto.

11.2 Contenido Entregado en el Archivo adjunto

Directorio	Contenido
<i>./ Directorio raiz</i>	Contiene un fichero leeme.txt explicando esta estructura
<i>./desarrollo</i>	Contiene toda la estructura de directorios del proyecto del desarrollo
<i>./dispensador3D</i>	Contiene todos los ficheros .stl utilizados para la impresión del dispensador. También contiene un fichero de propiedad de impresión
<i>./documentacion</i>	Contiene toda la documentación asociada al proyecto
<i>./documentacion/img</i>	Directorio que contiene las imágenes con mayor número de detalles utilizadas en la documentación
<i>./documentacion/planificacion</i>	Contiene el fichero con la planificación del proyecto
<i>./documentacion/presupuesto</i>	Contiene un fichero Excel con el presupuesto del proyecto
<i>./documentacion/uml</i>	Contiene los diagramas de secuencia de los casos de uso del proyecto
<i>./herramientas</i>	Contiene todas las librerías de desarrollo necesarias para la implementación del código del microcontrolador
<i>./instalacion</i>	Contiene una copia de la base de datos MongoDB utilizados para el desarrollo. Estos archivos pueden ser importados en cualquier instancia de MongoDB

Directorio	Contenido
<i>./ Directorio raíz de "desarrollo"</i>	Contiene los proyectos de desarrollo del TFM
<i>./desarrollo/SmartMedicineDispenser-ARDUINO_MEGA_ESP</i>	Contiene los ficheros de implementación del microcontrolador del dispensador
<i>./desarrollo/SmartMedicineDispenser-ARDUINO_MEGA_ESP/SmartMedicineDispenser-ATMEGA</i>	Contiene el fichero de implementación del módulo ATmega2560 del microcontrolador del dispensador
<i>./desarrollo/SmartMedicineDispenser-ARDUINO_MEGA_ESP/SmartMedicineDispenser-ESP8266</i>	Contiene el fichero de implementación del módulo ESP8266 wifi del microcontrolador del dispensador
<i>./desarrollo/SmartMedicineDispenser-NODEJS_SERVER</i>	Contiene el proyecto Node.js del trabajo. Sistema distribuido y aplicación web
<i>./desarrollo/SmartMedicineDispenser-NODEJS_SERVER/api</i>	Contiene la implementación de la API REST del servidor y la que accederá el microcontrolador del dispensador
<i>./desarrollo/SmartMedicineDispenser-NODEJS_SERVER/db</i>	Contiene la implementación de acceso a la base de datos MongoDB
<i>./desarrollo/SmartMedicineDispenser-</i>	Contiene ficheros estáticos que se incluyen

<i>NODEJS_SERVER/public</i>	en la renderización de las páginas de la aplicación web. Estos son; estilos CSS, imágenes y código JavaScript de ejecución en el lado del cliente
<i>./desarrollo/SmartMedicineDispenser-NODEJS_SERVER/routes</i>	Contiene la implementación de las rutas GET y POST de la aplicación web
<i>./desarrollo/SmartMedicineDispenser-NODEJS_SERVER/util</i>	Contiene ficheros de utilidad que son usados a lo largo de toda la implementación del proyecto
<i>./desarrollo/SmartMedicineDispenser-NODEJS_SERVER/web</i>	Contiene los archivos HTML y plantillas de renderización de las páginas de la aplicación web
<i>./desarrollo/SmartMedicineDispenser-NODEJS_SERVER/app.js</i>	Es el fichero de inicio y entrada del proyecto Node.js. Es el encargado de coordinar el resto de los ficheros de implementación del proyecto y establecer configuraciones del sistema
<i>./desarrollo/SmartMedicineDispenser-NODEJS_SERVER/package.json</i>	Contiene configuraciones de despliegue del proyecto y referencias a los paquetes NPM externos utilizados en la implementación

