

UNIVERSIDAD DEL BÍO-BÍO

Facultad de Ciencias Empresariales
Departamento Sistema de Informaciones



UNIVERSIDAD DEL BÍO-BÍO

MEMORIA PARA OPTAR A TÍTULO DE INGENIERO DE EJECUCIÓN EN
COMPUTACIÓN E INFORMÁTICA

SISTEMA DOMÓTICO MODULAR CON
AUTOMATIZACIÓN Y CONTROL DE DISPOSITIVOS
ELÉCTRICOS, BASADO EN NODEMCU-32S.

Alumno: Abdelkarin Sebastián Hadad Martínez

Profesor Guía: Daniel Enrique Bustos Romero

CONCEPCIÓN, 2019

RESUMEN

En la actualidad la sociedad está muy influenciada por los avances tecnológicos, pues estamos en una era en donde las nuevas tecnologías nacen a diario, por este motivo comienzan a surgir nuevas necesidades, de ahorro energético, confort, seguridad, comunicación y accesibilidad. En consecuencia, se empiezan a llevar a cabo prácticas en donde se mezclan estas tecnologías y dan origen a conceptos contemporáneos como la domótica y el Internet de las Cosas.

En lo que respecta a la domótica, es una tecnología que ayuda a gestionar el consumo y la actividad de diferentes componentes electrónicos de un hogar. De este modo, las personas pueden ver en tiempo real la actividad de sus electrodomésticos. Mientras que el Internet de las Cosas es la interconexión de objetos del mundo físico a Internet, los cuales están equipados con sensores, actuadores y tecnologías de comunicación.

Este proyecto propone una solución, para acercar la domótica y el Internet de las Cosas a las personas, sin importar el estrato socioeconómico. Para este propósito se hizo uso de herramientas y dispositivos bajo costo o gratuitas. El ESP32 es un dispositivo programable de valor reducido que originalmente se programa en el lenguaje Lua, pero fue adaptado para trabajar bajo el entorno de desarrollo Arduino. También se hizo uso del protocolo MQTT (Message Queue Telemetry Transport), pues proporciona un método ligero para llevar a cabo la mensajería, utilizando un modelo de publicación / suscripción. MQTTnet (biblioteca .NET de alto rendimiento para la comunicación basada en MQTT) es la biblioteca que se utilizó para el desarrollo de un servicio multiplataforma que cumple la función de servidor MQTT, el cual permite que la aplicación Android se comunique con el módulo domótico.

Por otro lado, para el manejo de datos y la validación de usuarios en el sistema, se implementó un web API. Finalmente, en este proyecto también se utilizaron sensores, actuadores y servicios externos, para así obtener un producto final que se pueda controlar y automatizar de forma remota, mediante el uso de una aplicación Android (forma parte del proyecto).

Palabras clave: domótica, Internet de las Cosas, ESP32, entorno de desarrollo, Arduino, MQTT, MQTTnet, servicio multiplataforma, sensores, actuadores, Android

INDICE

| | |
|---|----|
| CAPÍTULO 1: INTRODUCCIÓN | 17 |
| CAPÍTULO 2: MARCO TEORICO | 18 |
| 2.1. Internet de las cosas | 18 |
| 2.1.1. Definición del Internet de las cosas | 18 |
| 2.1.2. Origen | 18 |
| 2.1.3. Ecosistema | 19 |
| 2.1.4. Internet de las cosas y la domótica | 22 |
| CAPÍTULO 3: DEFINICION DE LA ACTIVIDAD..... | 23 |
| 3.1. Descripción del área de estudio | 23 |
| 3.2. Descripción de la problemática | 23 |
| CAPÍTULO 4: DEFINICIÓN DEL PROYECTO..... | 24 |
| 4.1. Objetivos del proyecto..... | 24 |
| 4.1.1. Objetivo General..... | 24 |
| 4.1.2. Objetivos Específicos | 24 |
| 4.2. Justificación del proyecto | 25 |
| 4.3. Ambiente de ingeniería en software | 26 |
| 4.3.1. Metodología empleada..... | 26 |
| 4.3.2. Las técnicas y notaciones: | 27 |
| 4.3.3. Estándares de documentación:..... | 28 |
| 4.3.4. Herramientas de apoyo a la planificación:..... | 28 |
| 4.3.5. Herramientas de apoyo al desarrollo del software:..... | 28 |

| | |
|--|-----------|
| 4.3.6. Herramientas al diagramado: | 29 |
| 4.3.7. Hardware de uso: | 29 |
| 4.3.8. Hardware de desarrollo:..... | 29 |
| 4.3.9. Lenguajes de programación: | 29 |
| 4.3.10. Protocolos de comunicación: | 29 |
| 4.3.11. Gestor de base de datos: | 29 |
| 4.3.12. Nombre de la base de datos: | 29 |
| 4.3.13. Dirección del servidor: | 30 |
| 4.3.14. Framework: | 30 |
| 4.4. Definiciones, siglas y abreviaciones..... | 30 |
| CAPÍTULO 5: ESPECIFICACION DE REQUERIMIENTOS DE SOFTWARE..... | 34 |
| 5.1. Alcances..... | 34 |
| 5.1.1. Alcances del servicio (Servidor MQTT) | 34 |
| 5.1.2. Alcances web API: | 35 |
| 5.1.3. Alcances de la aplicación Android | 35 |
| 5.1.4. Alcances del ESP32 (Arduino):..... | 36 |
| 5.1.5. Limitaciones | 38 |
| 5.2. Objetivo del software..... | 38 |
| 5.2.1. Objetivo general..... | 38 |
| 5.2.2. Objetivos específicos | 38 |
| 5.3. Descripción global del producto..... | 39 |
| 5.3.1. Perspectiva del producto..... | 39 |
| 5.3.2. Interfaces de usuario | 39 |
| 5.3.3. Interfaces de hardware..... | 39 |

| | |
|---|------------|
| 5.3.4. Interfaces de software | 40 |
| 5.3.5. Interfaces de comunicación | 40 |
| 5.3.6. Características de los usuarios | 41 |
| 5.3.7. Restricciones (complemento a los alcances) | 42 |
| 5.3.8. Suposiciones y dependencias..... | 42 |
| 5.4. Requerimientos Específicos..... | 42 |
| 5.4.1. Requerimientos funcionales..... | 42 |
| 5.4.1. Interfaces externas de entrada..... | 86 |
| 5.4.2. Interfaces externas de salida | 87 |
| 5.4.3. Atributos de calidad del sistema | 89 |
| CAPÍTULO 6: FACTIBILIDAD | 94 |
| 6.1. Factibilidad técnica..... | 94 |
| 6.1.1. Hardware..... | 94 |
| 6.1.2. Software..... | 95 |
| 6.1.3. Conclusión | 95 |
| 6.2. Factibilidad operativa | 96 |
| 6.3. Factibilidad económica..... | 97 |
| 6.4. Factibilidad Legal | 99 |
| 6.5. Conclusión de la factibilidad | 100 |
| CAPÍTULO 7: ANÁLISIS | 101 |
| 7.1. Diagrama de casos de uso..... | 101 |
| 7.1.1. Actores..... | 101 |
| 7.1.2. Diagramas de casos de Uso y descripción..... | 102 |
| 7.1.3. Especificación de los casos de uso | 108 |

| | |
|---|------------|
| 7.1.4. Modelamiento de datos..... | 172 |
| CAPÍTULO 8: DISEÑO..... | 174 |
| 8.1. Diseño físico de la base de datos | 174 |
| 8.1.1. Diccionario de datos | 175 |
| 8.2. Diseño de arquitectura funcional | 178 |
| 8.2.1. Web API | 179 |
| 8.2.2. Servidor MQTT | 181 |
| 8.2.3. Módulo domótico..... | 182 |
| 8.2.4. Aplicación móvil | 186 |
| 8.3. Diseño de interfaz y navegación..... | 188 |
| CAPÍTULO 9: PRUEBAS | 191 |
| 9.1. Elementos de prueba..... | 191 |
| 9.2. Especificación de las pruebas | 192 |
| 9.3. Responsables de las pruebas | 195 |
| 9.4. Conclusiones de Prueba..... | 195 |
| CONCLUSIONES..... | 196 |
| BIBLIOGRAFÍA | 197 |
| ANEXO 1: FORMATOS DE COMUNICACIÓN APP – MÓDULO DOMOTICO | 198 |
| 1.1. Establecer zona horaria (aplicación móvil a módulo) | 198 |
| 1.2. Encender o apagar relés (aplicación móvil a módulo)..... | 198 |
| 1.3. Automatizar encendido o apagado de los relés (aplicación móvil a módulo) | 198 |
| 1.4. Petición de sincronización (aplicación móvil a módulo)..... | 199 |
| 1.5. Eliminar horarios de los relés (aplicación móvil a módulo)..... | 199 |
| 1.6. Enviar zona horaria (módulo a aplicación móvil) | 200 |

| | |
|---|-----|
| 1.7. Enviar variables de ambiente (módulo a aplicación móvil) | 200 |
| 1.8. Enviar estado de los componentes | 200 |
| ANEXO 2: MANUAL DE USO PARA ADMINISTRADOR – WEB API..... | 202 |
| 2.1. Ingresar al sistema (iniciar sesión) | 202 |
| 2.2. Módulo - Usuarios | 203 |
| 2.2.1. Crear usuario..... | 203 |
| 2.2.2. Modificar usuario..... | 203 |
| 2.2.3. Buscar usuario | 203 |
| 2.2.4. Listar usuarios..... | 204 |
| 2.2.5. Eliminar usuario..... | 204 |
| 2.3. Módulo - Dispositivos domóticos..... | 204 |
| 2.3.1. Crear dispositivo domótico..... | 204 |
| 2.3.2. Modificar dispositivo domótico..... | 205 |
| 2.3.3. Buscar dispositivo domótico..... | 205 |
| 2.3.4. Listar dispositivos domóticos | 206 |
| 2.3.4. Eliminar dispositivo domótico..... | 206 |
| 2.4. Módulo - Tipos de Datos | 206 |
| 2.4.1. Crear tipo de dato..... | 206 |
| 2.4.2. Modificar tipo de dato | 207 |
| 2.4.3. Buscar tipo de dato | 207 |
| 2.4.4. Listar tipo de dato | 207 |
| 2.4.5. Eliminar tipo de dato | 208 |
| 2.5. Módulo - Tópicos | 208 |
| 2.5.1. Crear tópico | 208 |

| | |
|---|-----|
| 2.5.2. Modificar tópico | 209 |
| 2.5.3. Buscar tópico | 209 |
| 2.5.4. Listar tópicos | 209 |
| 2.5.5. Eliminar tópico | 210 |
| 2.6. Asignar módulos a usuarios (módulos o dispositivos domóticos)..... | 210 |
| 2.6.1. Asignar módulo a usuario | 210 |
| 2.6.2. Modificar módulo asignado a usuario | 210 |
| 2.6.3. Listar módulos asignados a un usuario | 211 |
| 2.6.4. Eliminar módulo asignado a usuario | 211 |
| 2.7. Asignar tópicos a módulos..... | 211 |
| 2.7.1. Asignar tópico a módulo..... | 211 |
| 2.7.2. Modificar tópico asignado a módulo | 212 |
| 2.7.3. Listar tópicos asignados a un módulo..... | 212 |
| 2.7.4. Eliminar tópico asignado a módulo | 212 |
| ANEXO 3: MATRIZ DE TRAZABILIDAD DE R.F. Y CASOS DE USO | 213 |

INDICE DE TABLAS

| | |
|---|----|
| Tabla 5-1 Características que debe poseer un administrador | 41 |
| Tabla 5-2 Características que debe poseer un usuario de la aplicación móvil | 41 |
| Tabla 5-3 Requerimiento funcional 01 | 43 |
| Tabla 5-4 Requerimiento funcional 02 | 44 |
| Tabla 5-5 Requerimiento funcional 03 | 44 |
| Tabla 5-6 Requerimiento funcional 04 | 45 |
| Tabla 5-7 Requerimiento funcional 05 | 46 |
| Tabla 5-8 Requerimiento funcional 06 | 47 |
| Tabla 5-9 Requerimiento funcional 07 | 48 |
| Tabla 5-10 Requerimiento funcional 08 | 49 |
| Tabla 5-11 Requerimiento funcional 09 | 49 |
| Tabla 5-12 Requerimiento funcional 10 | 50 |
| Tabla 5-13 Requerimiento funcional 11 | 51 |
| Tabla 5-14 Requerimiento funcional 12 | 52 |
| Tabla 5-15 Requerimiento funcional 13 | 52 |
| Tabla 5-16 Requerimiento funcional 14 | 53 |
| Tabla 5-17 Requerimiento funcional 15 | 53 |
| Tabla 5-18 Requerimiento funcional 16 | 54 |
| Tabla 5-19 Requerimiento funcional 17 | 55 |
| Tabla 5-20 Requerimiento funcional 18 | 55 |
| Tabla 5-21 Requerimiento funcional 19 | 56 |
| Tabla 5-22 Requerimiento funcional 20 | 57 |
| Tabla 5-23 Requerimiento funcional 21 | 57 |
| Tabla 5-24 Requerimiento funcional 22 | 58 |
| Tabla 5-25 Requerimiento funcional 23 | 59 |
| Tabla 5-26 Requerimiento funcional 24 | 59 |
| Tabla 5-27 Requerimiento funcional 25 | 60 |

| | |
|---|----|
| Tabla 5-28 Requerimiento funcional 26 | 61 |
| Tabla 5-29 Requerimiento funcional 27 | 62 |
| Tabla 5-30 Requerimiento funcional 28 | 62 |
| Tabla 5-31 Requerimiento funcional 29 | 63 |
| Tabla 5-32 Requerimiento funcional 30 | 64 |
| Tabla 5-33 Requerimiento funcional 31 | 65 |
| Tabla 5-34 Requerimiento funcional 32 | 65 |
| Tabla 5-35 Requerimiento funcional 33 | 66 |
| Tabla 5-36 Requerimiento funcional 34 | 67 |
| Tabla 5-37 Requerimiento funcional 35 | 67 |
| Tabla 5-38 Requerimiento funcional 36 | 68 |
| Tabla 5-39 Requerimiento funcional 37 | 69 |
| Tabla 5-40 Requerimiento funcional 38 | 70 |
| Tabla 5-41 Requerimiento funcional 39 | 70 |
| Tabla 5-42 Requerimiento funcional 40 | 71 |
| Tabla 5-43 Requerimiento funcional 41 | 72 |
| Tabla 5-44 Requerimiento funcional 42 | 73 |
| Tabla 5-45 Requerimiento funcional 43 | 74 |
| Tabla 5-46 Requerimiento funcional 44 | 74 |
| Tabla 5-47 Requerimiento funcional 45 | 75 |
| Tabla 5-48 Requerimiento funcional 46 | 76 |
| Tabla 5-49 Requerimiento funcional 47 | 76 |
| Tabla 5-50 Requerimiento funcional 48 | 77 |
| Tabla 5-51 Requerimiento funcional 49 | 78 |
| Tabla 5-52 Requerimiento funcional 50 | 79 |
| Tabla 5-53 Requerimiento funcional 51 | 80 |
| Tabla 5-54 Requerimiento funcional 52 | 81 |
| Tabla 5-55 Requerimiento funcional 53 | 82 |
| Tabla 5-56 Requerimiento funcional 54 | 82 |

| | |
|--|-----|
| Tabla 5-57 Requerimiento funcional 55 | 83 |
| Tabla 5-58 Requerimiento funcional 56 | 84 |
| Tabla 5-59 Requerimiento funcional 57 | 84 |
| Tabla 5-60 Requerimiento funcional 58 | 85 |
| Tabla 5-61 Requerimiento funcional 59 | 85 |
| Tabla 5-62 Requerimiento funcional 60 | 86 |
| Tabla 5-63 Interfaces externas de entrada | 87 |
| Tabla 5-64 Interfaces externas de salida..... | 88 |
| Tabla 5-65 Atributos de calidad de eficiencia | 89 |
| Tabla 5-66 Atributos de calidad de usabilidad | 90 |
| Tabla 5-67 Atributos de calidad de disponibilidad..... | 90 |
| Tabla 5-68 Atributos de calidad de seguridad | 91 |
| Tabla 5-69 Atributos de calidad de mantenibilidad..... | 92 |
| Tabla 5-70 Atributos de calidad de portabilidad | 92 |
| Tabla 5-71 Atributos de calidad organizacionales..... | 93 |
| Tabla 6-1 Tabla de costos - Factibilidad económica | 98 |
| Tabla 6-2 Costo por módulo - Factibilidad económica | 99 |
| Tabla 7-1 Caso de uso establecer conexión - Flujo normal..... | 108 |
| Tabla 7-2 Caso de uso establecer conexión - Flujo alternativo | 108 |
| Tabla 7-3 Caso de uso iniciar sesión - Flujo normal (App movil y administrador) | 109 |
| Tabla 7-4 Caso de uso iniciar sesión - Flujo alternativo (App movil y administrador) | 109 |
| Tabla 7-5 Caso de uso iniciar sesión - Flujo normal(módulo domótico) | 110 |
| Tabla 7-6 Caso de uso iniciar sesión - Flujo alternativo(módulo domótico) | 110 |
| Tabla 7-7 Caso de uso registrarse - Flujo normal..... | 111 |
| Tabla 7-8 Caso de uso registrarse - Flujo alternativo | 111 |
| Tabla 7-9 Caso de uso salir - Flujo normal..... | 112 |
| Tabla 7-10 Caso de uso publicar en tópicos - Flujo normal | 113 |
| Tabla 7-11 Caso de uso publicar en tópicos - Flujo alternativo | 113 |
| Tabla 7-12 Caso de uso mensaje para controlar relés - Flujo normal | 114 |

| | |
|--|-----|
| Tabla 7-13 Caso de uso mensaje para controlar relés - Flujo alternativo..... | 114 |
| Tabla 7-14 Caso de uso mensaje para automatizar relés - Flujo normal | 115 |
| Tabla 7-15 Caso de uso mensaje para automatizar relés - Flujo alternativo | 115 |
| Tabla 7-16 Caso de uso mensaje para establecer zona horaria - Flujo normal..... | 116 |
| Tabla 7-17 Caso de uso mensaje para establecer zona horaria - Flujo alternativo..... | 116 |
| Tabla 7-18 Caso de uso mensaje para la sincronización - Flujo normal | 117 |
| Tabla 7-19 Caso de uso mensaje para la sincronización - Flujo alternativo | 117 |
| Tabla 7-20 Caso de uso suscribirse a tópicos - Flujo normal | 118 |
| Tabla 7-21 Caso de uso suscribirse a tópicos - Flujo alternativo | 118 |
| Tabla 7-22 Caso de uso visualizar temperatura y humedad - Flujo normal | 119 |
| Tabla 7-23 Caso de uso visualizar temperatura y humedad - Flujo alternativo | 119 |
| Tabla 7-24 Caso de uso visualizar zona horaria - Flujo normal | 120 |
| Tabla 7-25 Caso de uso visualizar zona horaria - Flujo alternativo | 120 |
| Tabla 7-26 Caso de uso respuesta de sincronización - Flujo normal | 121 |
| Tabla 7-27 Caso de uso respuesta de sincronización - Flujo alternativo..... | 121 |
| Tabla 7-28 Caso de uso publicar tópicos - Flujo normal..... | 122 |
| Tabla 7-29 Caso de uso publicar tópicos - Flujo alternativo | 122 |
| Tabla 7-30 Caso de uso responder temperatura y humedad - Flujo normal..... | 123 |
| Tabla 7-31 Caso de uso responder temperatura y humedad - Flujo normal..... | 123 |
| Tabla 7-32 Caso de uso responder zona horaria - Flujo normal..... | 124 |
| Tabla 7-33 Caso de uso responder zona horaria - Flujo alternativo | 124 |
| Tabla 7-34 Caso de uso responder a la sincronización - Flujo normal..... | 125 |
| Tabla 7-35 Caso de uso responder a la sincronización - Flujo alternativo | 125 |
| Tabla 7-36 Caso de uso suscribirse a tópicos - Flujo normal | 126 |
| Tabla 7-37 Caso de uso suscribirse a tópicos - Flujo normal | 126 |
| Tabla 7-38 Caso de uso solicitud para automatizar relés - Flujo normal | 127 |
| Tabla 7-39 Caso de uso solicitud para automatizar relés - Flujo alternativo..... | 127 |
| Tabla 7-40 Caso de uso solicitud para cambiar zona horaria - Flujo normal | 128 |
| Tabla 7-41 Caso de uso solicitud para cambiar zona horaria - Flujo alternativo | 128 |

| | |
|---|-----|
| Tabla 7-42 Caso de uso automatizar relés - Flujo normal | 129 |
| Tabla 7-43 Caso de uso cambiar zona horaria - Flujo normal..... | 130 |
| Tabla 7-44 Caso de uso administrar usuarios - Flujo normal..... | 131 |
| Tabla 7-45 Caso de uso administrar usuarios - Flujo alternativo | 131 |
| Tabla 7-46 Caso de uso crear usuario - Flujo normal..... | 132 |
| Tabla 7-47 Caso de uso crear usuario - Flujo alternativo..... | 132 |
| Tabla 7-48 Caso de uso actualizar usuario - Flujo normal | 133 |
| Tabla 7-49 Caso de uso actualizar usuario - Flujo alternativo | 133 |
| Tabla 7-50 Caso de uso eliminar usuario - Flujo normal | 134 |
| Tabla 7-51 Caso de uso eliminar usuario - Flujo alternativo..... | 134 |
| Tabla 7-52 Caso de uso buscar usuario - Flujo normal | 135 |
| Tabla 7-53 Caso de uso buscar usuario - Flujo alternativo | 135 |
| Tabla 7-54 Caso de uso crear usuario normal - Flujo normal | 136 |
| Tabla 7-55 Caso de uso crear usuario normal - Flujo alternativo..... | 136 |
| Tabla 7-56 Caso de uso crear usuario administrador - Flujo normal | 137 |
| Tabla 7-57 Caso de uso crear usuario administrador - Flujo alternativo..... | 137 |
| Tabla 7-58 Caso de uso listar usuarios - Flujo normal | 138 |
| Tabla 7-59 Caso de uso listar usuario - Flujo alternativo..... | 138 |
| Tabla 7-60 Caso de uso administrar módulos - Flujo normal | 139 |
| Tabla 7-61 Caso de uso administrar módulos - Flujo alternativo..... | 139 |
| Tabla 7-62 Caso de uso crear módulo - Flujo normal | 140 |
| Tabla 7-63 Caso de uso crear módulo - Flujo alternativo | 140 |
| Tabla 7-64 Caso de uso actualizar módulo - Flujo normal..... | 141 |
| Tabla 7-65 Caso de uso actualizar módulo - Flujo alternativo..... | 141 |
| Tabla 7-66 Caso de uso eliminar módulo - Flujo normal..... | 142 |
| Tabla 7-67 Caso de uso actualizar módulo - Flujo alternativo..... | 142 |
| Tabla 7-68 Caso de uso buscar módulo - Flujo normal..... | 143 |
| Tabla 7-69 Caso de uso buscar módulo - Flujo alternativo..... | 143 |
| Tabla 7-70 Caso de uso listar módulos - Flujo normal..... | 144 |

| | |
|--|-----|
| Tabla 7-71 Caso de uso listar módulos - Flujo alternativo | 144 |
| Tabla 7-72 Caso de uso administrar tópicos - Flujo normal..... | 145 |
| Tabla 7-73 Caso de uso administrar tópico - Flujo alternativo | 145 |
| Tabla 7-74 Caso de uso crear tópico - Flujo normal | 146 |
| Tabla 7-75 Caso de uso crear tópico - Flujo alternativo..... | 146 |
| Tabla 7-76 Caso de uso asignar tipo de dato - Flujo normal | 147 |
| Tabla 7-77 Caso de uso asignar tipo de dato - Flujo alternativo | 147 |
| Tabla 7-78 Caso de uso actualizar tópico - Flujo normal..... | 148 |
| Tabla 7-79 Caso de uso actualizar tópico - Flujo alternativo | 148 |
| Tabla 7-80 Caso de uso eliminar tópico - Flujo normal | 149 |
| Tabla 7-81 Caso de uso eliminar tópico - Flujo alternativo | 149 |
| Tabla 7-82 Caso de uso buscar tópico - Flujo normal | 150 |
| Tabla 7-83 Caso de uso buscar tópico - Flujo alternativo | 150 |
| Tabla 7-84 Caso de uso listar tópicos - Flujo normal..... | 151 |
| Tabla 7-85 Caso de uso listar tópicos - Flujo alternativo | 151 |
| Tabla 7-86 Caso de uso administrar tipos de datos - Flujo normal | 152 |
| Tabla 7-87 Caso de uso administrar tipos de datos - Flujo alternativo | 152 |
| Tabla 7-88 Caso de uso crear tipo de dato - Flujo normal..... | 153 |
| Tabla 7-89 Caso de uso crear tipo de dato - Flujo alternativo..... | 153 |
| Tabla 7-90 Caso de uso actualizar tipo de dato - Flujo normal | 154 |
| Tabla 7-91 Caso de uso actualizar tipo de dato - Flujo alternativo | 154 |
| Tabla 7-92 Caso de uso eliminar tipo de dato - Flujo normal | 155 |
| Tabla 7-93 Caso de uso eliminar tipo de dato - Flujo alternativo | 155 |
| Tabla 7-94 Caso de uso buscar tipo de dato - Flujo normal | 156 |
| Tabla 7-95 Caso de uso buscar tipo de dato - Flujo alternativo | 156 |
| Tabla 7-96 Caso de uso listar tipos de datos - Flujo normal | 157 |
| Tabla 7-97 Caso de uso listar tipos de datos - Flujo alternativo..... | 157 |
| Tabla 7-98 Caso de uso administrar asignación de módulos a usuario – Flujo normal | 158 |
| Tabla 7-99 Caso de uso administrar asignación de módulos a usuario – Flujo alternativo.... | 158 |

| | |
|--|-----|
| Tabla 7-100 Caso de uso asignar módulo a usuario – Flujo normal | 159 |
| Tabla 7-101 Caso de uso asignar módulo a usuario – Flujo alternativo..... | 159 |
| Tabla 7-102 Caso de uso actualizar asignación de módulo a usuario – Flujo normal..... | 160 |
| Tabla 7-103 Caso de uso actualizar asignación de módulo a usuario – Flujo alternativo..... | 160 |
| Tabla 7-104 Caso de uso listar módulos asignados a usuario – Flujo normal..... | 162 |
| Tabla 7-105 Caso de uso listar módulos asignados a usuario – Flujo alternativo..... | 162 |
| Tabla 7-106 Caso de uso administrar asignación de tópicos a módulos - Flujo normal | 163 |
| Tabla 7-107 Caso de uso administrar asignación de tópicos a módulos - Flujo alternativo .. | 163 |
| Tabla 7-108 Caso de uso asignar tópico a módulo - Flujo normal..... | 164 |
| Tabla 7-109 Caso de uso asignar tópico a módulo - Flujo alternativo | 164 |
| Tabla 7-110 Caso de uso actualizar asignación de tópico a módulo - Flujo normal | 165 |
| Tabla 7-111 Caso de uso actualizar asignación de tópico a módulo - Flujo alternativo | 165 |
| Tabla 7-112 Caso de uso eliminar asignación de tópico a módulo - Flujo normal | 166 |
| Tabla 7-113 Caso de uso eliminar asignación de tópico a módulo - Flujo alternativo | 166 |
| Tabla 7-114 Caso de uso listar tópicos asignados a módulos - Flujo normal..... | 167 |
| Tabla 7-115 Caso de uso listar tópicos asignados a módulos - Flujo alternativo..... | 167 |
| Tabla 7-116 Caso de uso mensaje horarios automáticos relé - Flujo normal..... | 168 |
| Tabla 7-117 Caso de uso mensaje horarios automáticos relé - Flujo alternativo | 168 |
| Tabla 7-118 Caso de uso solicitud para desactivar horarios automáticos relés - Flujo normal | 169 |
| Tabla 7-119 Caso de uso solicitud para desactivar horarios automáticos relés - Flujo alternativo | 169 |
| Tabla 7-120 Caso de uso desactivar horarios - Flujo normal | 170 |
| Tabla 7-121Caso de uso control remoto - Flujo normal..... | 171 |
| Tabla 9-1 Resultado de las pruebas | 193 |
| Tabla 9-2 Detalle de las pruebas - Particiones equivalentes..... | 195 |
| Tabla A -1 Matriz de trazabilidad, requerimientos funcionales vs casos de uso..... | 217 |

INDICE DE ILUSTRACIONES

| | |
|--|-----|
| Ilustración 2-1 Concepto del IoT | 20 |
| Ilustración 2-2 Control y monitoreo de hogar usando el IoT | 22 |
| Ilustración 7-1 Casos de uso usuario - Control del módulo domótico | 103 |
| Ilustración 7-2 Casos de uso del módulo domótico - Respuesta a usuarios | 103 |
| Ilustración 7-3 Casos de uso administrador - Módulo usuarios | 104 |
| Ilustración 7-4 Casos de uso administrador - Módulo de dispositivos domóticos | 105 |
| Ilustración 7-5 Casos de uso administrador - Módulo tópicos | 105 |
| Ilustración 7-6 Casos de uso administrador - Módulo tipos de datos..... | 106 |
| Ilustración 7-7 Casos de uso administrador - Asignar módulos a usuarios..... | 106 |
| Ilustración 7-8 Casos de uso administrador - Asignar tópicos a módulos..... | 107 |
| Ilustración 7-9 Modelo entidad relación, notación de Martin y notación patas de gallo..... | 172 |
| Ilustración 8-1 Modelo relacional del ecosistema domótico | 174 |
| Ilustración 8-2 Arquitectura-perspectiva cliente HTTP | 178 |
| Ilustración 8-3 Arquitectura-perspectiva cliente de aplicación Android..... | 178 |
| Ilustración 8-4 Arquitectura-perspectiva módulo domótico..... | 179 |
| Ilustración 8-5 Estructura de directorios web API | 179 |
| Ilustración 8-6 Estructura de directorios servidor MQTT | 181 |
| Ilustración 8-7 Circuito del módulo domótico..... | 182 |
| Ilustración 8-8 Datasheet NodeMCU-32s | 184 |
| Ilustración 8-9 Circuito esquemático del módulo domótico..... | 185 |
| Ilustración 8-10 Estructura de directorios App Android | 186 |
| Ilustración 8-11 Diseño de interfaz y navegación | 188 |

CAPÍTULO 1: INTRODUCCIÓN

El presente proyecto se refiere a la creación de un sistema domótico modular basado en el dispositivo NODEMCU-ESP32, utilizado en proyectos del Internet de las Cosas, específicamente se creará un módulo para el control y automatización de electrodomésticos. Además, se desarrollarán las aplicaciones para el manejo de este. Las razones de la elección del tema recaen en la progresiva demanda de inclusión de la conectividad y de control a objetos de la vida cotidiana. También por la carencia de empresas nacionales dedicadas a fabricar dispositivos de esta índole.

La domótica corresponde a un conjunto de tecnologías que trabajan al unísono, estas son utilizadas para el control y automatización de las viviendas, con la finalidad de lograr una gestión energética eficiente, lo que genera un ambiente de seguridad y confort, pues se tiene conocimiento del estado de ciertos componentes de un hogar. Un sistema domótico recoge información de una serie de sensores y/o dispositivos diferentes, procesa las variables de entrada y actúa de acorde a ello, enviando órdenes a diferentes actuadores reconocidos por el sistema. Adicionalmente, el sistema puede acceder a redes externas tales como: web APIs o bases de datos.

El uso de la domótica en las viviendas permite dar respuesta a las nuevas necesidades que generan los cambios sociales y las nuevas tendencias de vida de las personas. En cuanto al Internet de las cosas, este hace referencia a la interconexión de los objetos del mundo físico a través de Internet, por tanto, el IoT no compite contra la domótica, la complementa (Salazar y Silvestre, 2016).

En relación a todo lo anterior, este trabajo tiene como objetivo construir un sistema de automatización de viviendas, con un módulo compacto y portable, que se controle de forma remota, haciendo uso de una aplicación Android.

Este proyecto nace del interés de acercar la domótica y el Internet de las Cosas a todos los sectores socioeconómicos, elaborando un ecosistema domótico (prototipo) de bajo costo y desarrollado mediante el uso de software libre, pues las opciones existentes en el mercado tienen escasas características y vienen acompañadas de costos elevados.

CAPÍTULO 2: MARCO TEORICO

2.1. Internet de las cosas

2.1.1. Definición del Internet de las cosas

En lo concerniente al Internet de las Cosas (en inglés, Internet of Things, abreviado IoT) se puede mencionar que es un concepto contemporáneo que hace alusión a la interconexión digital de objetos de uso cotidiano con la Internet. En palabras concretas el IoT potencia objetos que tradicionalmente se conectan mediante circuitos cerrados, tales como: cámaras, sensores o cualquier otro electrodoméstico, y les permite comunicarse globalmente mediante el uso de la red de redes (Isaías Bonilla, Arturo Tavizon y Melisa Morales, 2016).

2.1.2. Origen

En lo que respecta al nacimiento del Internet en las Cosas, se puede dilucidar que es muy popular en las áreas de Informática, Arquitectura, Electrónica y Telecomunicaciones; sin embargo, no es un concepto nuevo, pues en 1999, Kevin Ashton preparaba los senderos para lo que hoy en día conocemos como IoT, en el MIT AutoID Labs.

Ashton fue uno de los pioneros en concebir esta idea, mientras buscaba una solución para mejorar el negocio de Procter & Gamble mediante la vinculación de la información de identificación por radio frecuencia (RFID) a Internet. El concepto era simple y muy potente, pues si todos los objetos cotidianos estuvieran equipados con identificadores, sensores y conexión inalámbrica, estos podrían ser gestionados o controlados por computadores.

En 2009, en un artículo para RFID Journal Ashton escribió:

Los ordenadores actuales —y, por tanto, internet— son prácticamente dependientes de los seres humanos para recabar información. Una mayoría de los casi 50 petabytes (un petabyte son 1024 terabytes) de datos disponibles en internet fueron inicialmente creados por humanos, a base de teclear, presionar un botón, tomar una imagen digital o

escanear un código de barras. Los diagramas convencionales de internet, dejan fuera a los routers más importantes de todos: las personas. El problema es que las personas tienen un tiempo, una atención y una precisión limitados, y no se les da muy bien conseguir información sobre cosas en el mundo real. Y eso es un gran obstáculo. Somos cuerpos físicos, al igual que el medio que nos rodea. No podemos comer bits, ni quemarlos para resguardarnos del frío, ni meterlos en tanques de gas. Las ideas y la información son importantes, pero las cosas cotidianas tienen mucho más valor. Aunque, la tecnología de la información actual es tan dependiente de los datos escritos por personas que nuestros ordenadores saben más sobre ideas que sobre cosas. Si tuviéramos ordenadores que supieran todo lo que tuvieran que saber sobre las “cosas”, mediante el uso de datos que ellos mismos pudieran recoger sin nuestra ayuda, nosotros podríamos monitorizar, contar y localizar todo a nuestro alrededor, de esta manera se reducirían increíblemente gastos, pérdidas y costes. Sabríamos cuándo reemplazar, reparar o recuperar lo que fuera, así como conocer si su funcionamiento estuviera siendo correcto. El internet de las cosas tiene el potencial para cambiar el mundo tal y como hizo la revolución digital hace unas décadas. Tal vez incluso hasta más. (Ashton, 2009).

El auge de la Internet ha logrado la interconexión entre personas, gracias a la utilización de dispositivos inteligentes, los cuales permiten ejecutar las aplicaciones que nos mantienen comunicados, sin embargo, ahora es el turno de la interconexión de las cosas u objetos, para así dar un nuevo paso en la era de la información y la adaptación tecnológica.

2.1.3. Ecosistema

En lo que respecta al hábitat, se han utilizado los conceptos empleados por Jordi Salazar y Santiago Silvestre en su libro *Internet de las cosas (2016)*. En este libro se menciona que IoT puede ser visto como una combinación de sensores y actuadores, los cuales reciben y proporcionan información digital. Además de colocarla en redes bidireccionales capaces de transmitir todos los datos para ser utilizados por usuarios o servicios finales.

Las interacciones entre usuarios, procesos, cosas y datos están dando origen a nuevos servicios y aplicaciones, en donde los sensores, actuadores y el tipo de conectividad juegan un rol fundamental. La ilustración 2-1 muestra la composición del ecosistema IoT.

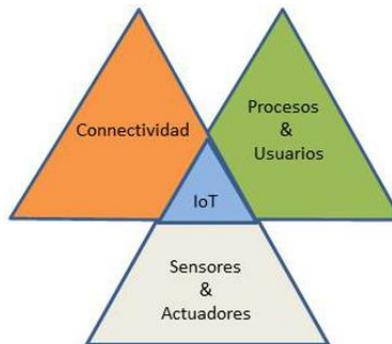


Ilustración 2-1 Concepto del IoT

2.1.3.1. Los sensores

En la actualidad gran porcentaje de las cosas del mundo físico no están conectadas a Internet, pero múltiples sensores se pueden unir a un objeto o dispositivo para medir una alta gama de variables físicas, o fenómenos y luego subir los datos a la nube para su posterior utilización. En los últimos años esta realidad se ha visto alterada, pues el IoT busca eliminar la brecha del mundo on y off (Salazar y Silvestre, 2016).

Los sensores forman parte de los pilares del internet de las cosas, estos pueden implementarse en cualquier parte, incluso bajo la piel humana. También varían su tamaño dependiendo del uso que se ha de dar a estos. Además, pueden poseer un tamaño milimétrico, sin embargo, su información es transmitida a cientos de kilómetros de distancia. Estos artefactos son utilizados como complemento a los sentidos humanos, y son indispensables en un gran número de industrias, desde la salud hasta la educación, pero sin duda una de las facultades que más destacan de

estos periféricos, es que tienen la ventaja de poder anticiparse a las necesidades humanas, gracias a los datos recopilados del entorno (Salazar y Silvestre, 2016).

2.1.3.2. La conectividad

La conectividad en los dispositivos del Internet de las Cosas puede ir desde el Bluetooth de bajo consumo hasta las más recientes tecnologías de comunicación móvil (4G LTE, LTE o incluso 5G, siempre y cuando haya cobertura de estas redes). Un ejemplo de conectividad sería cuando se ocupa la navegación GPS mientras se va de viaje en un automóvil. El dispositivo muestra la geolocalización gracias a que establece comunicación con varios satélites, por otro lado, el mapa de la ubicación se descarga gracias a la tecnología 4G.

En este sentido es muy importante el tipo de necesidad que se quiere cubrir y el tipo de conectividad que se va a utilizar, ya que los costos de implementación de cada tecnología son distintos y así también la cobertura que se busca alcanzar.

2.1.3.3. Las personas y procesos

La comunicación que llega a Internet a través de los dispositivos IoT puede ser combinada en sistema que integran los datos, personas, procesos y otros sistemas, que ayudarán a realizar una mejor toma de decisiones, para alcanzar objetivos específicos. Es decir, la interacción entre **Sensores + Conectividad + personas + Procesos** están creando nuevos tipos de aplicaciones y servicios; por ejemplo:

- Los termostatos inteligentes que adaptan la temperatura dependiendo de las necesidades de un hogar, lo que permite ahorrar recursos y dinero (claro ejemplo de domótica).
- Las pulseras inteligentes para deportes que pueden recoger continuamente y en tiempo real la presión arterial, el consumo de calorías y la temperatura corporal.

2.1.4. Internet de las cosas y la domótica

Como se ha mencionado con anterioridad, el Internet de las cosas consiste en combinar varios sensores, circuitos electrónicos, procesadores o cualquier dispositivo a Internet. Aquí es donde muchas personas se confunden con el concepto de domótica, ya que se popularizó junto con el IoT. La domótica es un conjunto de sistemas capaces de automatizar servicios de gestión energética, seguridad, bienestar y comunicación. Todo esto aplicado a las viviendas y tiene como objetivo programar respuestas o movimientos a cientos de elementos que faciliten o mejoren la vida de las personas.

En síntesis, el IoT no es reemplazado por la domótica o viceversa, dado a que ambos se complementan y actúan en forma conjunta. La domótica necesita del IoT. En la siguiente ilustración se contextualiza el flujo de la comunicación en un entorno domótico/IoT.

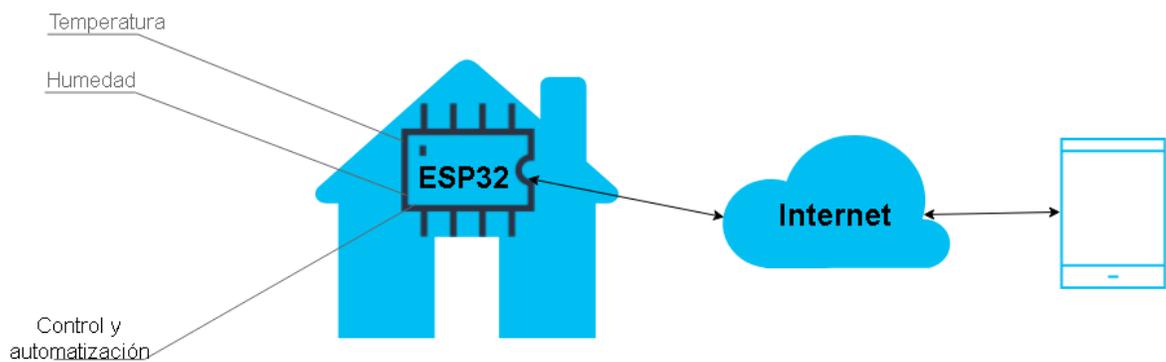


Ilustración 2-2 Control y monitoreo de hogar usando el IoT

CAPÍTULO 3: DEFINICION DE LA ACTIVIDAD

3.1. Descripción del área de estudio

En lo que atañe al proyecto, se ha de dilucidar que abarca conceptos contemporáneos, tales como: El internet de las cosas y la domótica. El enfoque de este va orientado a la automatización de viviendas, mediante la creación de módulos que permitan mecanizar y controlar remotamente el encendido y apagado de componentes eléctricos o electrónicos, además de medir variables de ambientes tales como temperatura y humedad. Para tal finalidad, se desarrollará un web API, un servicio MQTT, una aplicación móvil (Android) y el módulo que será controlado (ESP32).

3.2. Descripción de la problemática

La ocurrencia de crear un sistema en base a módulos domóticos surge por la necesidad de estar interconectados, ya no solo con otras personas, sino que también con las cosas. Debido a esto, en los últimos años han hecho conocidos los conceptos mencionados en el capítulo anterior. Hoy en día existen múltiples compañías que están acercando estos conceptos al común de las personas gracias a la incorporación de objetos inteligentes al mercado, los cuales pueden ser controlados o monitoreados desde un dispositivo móvil, sin embargo, adquirir un dispositivo que nos permita controlar y automatizar encendidos tiene un valor muy elevado. Además, existe la opción de adquirir casas domóticas, pero no todas las personas pueden conseguir ese lujo, debido a los exagerados costos que ello implica.

CAPÍTULO 4: DEFINICIÓN DEL PROYECTO

4.1. Objetivos del proyecto

4.1.1. Objetivo General

Construir un sistema de automatización de viviendas de bajo costo, con un módulo compacto y portable, que se controle de forma remota, mediante una aplicación Android.

4.1.2. Objetivos Específicos

- Desarrollar un producto con materiales de bajo costo, capaz de competir en contra de los módulos Sonoff de Itead.
- Elaborar un módulo domótico en base al dispositivo ESP32 con control remoto, opciones de automatización de sus componentes y lectura de variables de ambientes, tales como: Temperatura y humedad.
- Construir un web API multiplataforma, que permita que la aplicación móvil y el servidor MQTT establezcan comunicación a la base de datos. También el administrador podrá interactuar con este, haciendo uso de un cliente HTTP.
- Confeccionar una aplicación Android que permita el registro de usuarios en el sistema y el control de los módulos domóticos.
- Proveer seguridad necesaria para restringir control de los módulos a terceras personas.
- Utilizar la tecnología más reciente de Microsoft para desarrollar un sistema multiplataforma que pueda coordinar múltiples peticiones de usuarios.

4.2. Justificación del proyecto

En la actualidad las personas tienen una instalación eléctrica convencional en sus hogares, y se conforman sencillamente con iluminarse, calentarse, disponer de acceso al tendido eléctrico para el uso de sus electrodomésticos y contar con acceso a Internet.

Como seres humanos dejamos de lado las cosas que desconocemos, y es lo que ocurre con las casas inteligentes, pues se cree que los costos de transformar el hogar en uno inteligente son elevados, pero esta situación no es así.

Actualmente no es necesario reemplazar cableados eléctricos o hacer grandes intervenciones estéticas en el hogar, esto sucede gracias a la evolución los componentes electrónicos y al uso de tecnologías inalámbricas (wifi, bluetooth, RFID, etc.).

Vivimos en una era de interconexión digital, donde la Internet tiene el papel principal, pues es la red de comunicación más grande y diversa que jamás se ha visto en la historia, y si adicionalmente nos enfocamos en la capacidad inherente de dominación que tiene el ser humano y sumado al capricho de transformar la ficción en realidad. Esta obsesión del ser humano de conocer y dominar el mundo que le rodea, le ha llevado a intentar controlar todos los aspectos de su vida cotidiana. De esta forma nace la domótica (casa autónoma).

Interactuamos constantemente con la Internet, accediendo a ella más frecuentemente mediante el uso de computadores o dispositivos móviles.

Por tanto, la finalidad de este proyecto es fomentar la domótica de una manera sencilla y económica, sin hacer intervenciones en el hogar, otorgando control remoto y automatización del encendido y apagado de los aparatos eléctricos. Mediante la construcción de un sistema domótico modular con topología estrella (control sobre aparatos eléctricos conectados al módulo).

La forma de funcionamiento del módulo es en base a la creación y suscripción de tópicos en el servidor, y para el control de este se desarrollará una aplicación Android. Esta aplicación actuará como cliente en el servidor MQTT y se suscribirá a los tópicos del módulo (Dispositivo ESP32), dando control sobre este. La validación de usuarios en la aplicación Android y en el servidor MQTT se realizará mediante un web API.

4.3. Ambiente de ingeniería en software

4.3.1. Metodología empleada

La metodología a utilizar es Kanban, la implementación básica de esta, consta en dividir una pizarra en tres secciones (pendiente, haciendo y terminado), donde cada sección pertenece al estado de una tarea.

Se ha seleccionado esta metodología, ya que el proyecto está sujeto a múltiples cambios y se complementa bien con el patrón de diseño MVC (Modelo Vista Controlador) utilizado en el desarrollo del web API y la aplicación Android. También se acomoda a la evolución del código Arduino confeccionado.

Para poder agilizar las tareas, Kanban cuenta con cuatro principios básicos que lo fundamentan:

- **Visualización:** Para poder comprender en qué momento del desarrollo se encuentra el proyecto o revisar los temas involucrados, Kanban es totalmente transparente, de forma que se tiene acceso a todas las tareas en cualquier momento.
- **En proceso:** Tareas que se están realizando (no más de dos).
- **Priorización:** Gracias a este concepto, cuando se va al bloque de las tareas pendientes, ya se tiene claro cuál es el siguiente tema a tratar. La transparencia permite ver todos los ejercicios a realizar, esto hace que sea posible una mejor gestión del tiempo.
- **Medir el tiempo:** Gracias al sistema de situar las tareas en “haciendo”, durante el tiempo que se trabaja en ellas y etiquetar las tarjetas según el tema tratado, se puede hacer un seguimiento del tiempo invertido en cada función, departamento o campo.

Ventajas principales de Kanban:

- **Procesos innecesarios:** Debido a las facilidades de las diferentes herramientas Kanban, evitamos procesos que reduzcan la eficiencia.
- **Flexibilidad:** vivimos tiempos rápidos, tenemos que saber cómo enfrentarnos a los cambios, y Kanban permite esto, se puede hacer variaciones al producto tan

pronto como se detecte un error. Estos errores se localizan en breves periodos de tiempo debido a la corta duración de las tareas asignadas.

- **Más eficaces:** Kanban permite optimizar los tiempos, lo cual se traduce en eficiencia. Mayor calidad en menor tiempo.

Cada tarea de la carta Gantt será una tarjeta en Kanban y éstas velan por el cumplimiento de los objetivos específicos.

4.3.2. Las técnicas y notaciones:

Para la realización del presente proyecto se utilizaron los siguientes paradigmas de programación: programación modular y programación orientada a objetos.

En cuanto a la programación modular, se puede mencionar que consiste en dividir un programa en módulos o subprogramas con el fin de obtener un código más legible, manejable y así también poder reutilizar código. Este paradigma fue utilizado en lo que respecta a la programación del módulo domótico (dispositivo ESP32), otorgando la ventaja de dividir un problema, en problemas más pequeños, llamados subproblemas. De esta forma se logra separar la función principal del resto del código, y como resultado modularizamos las funciones del ESP32 (Beltrán y Aguirre, 2011).

En lo concerniente a la programación orientada a objetos, esta se aplicó en el desarrollo de la aplicación móvil y del web API. La POO ofrece un amplio dominio sobre el programa, pues este se controla a sí mismo y la mente del programador se libera enormemente permitiendo desarrollar aplicaciones más complejas al exigir menor esfuerzo de atención. Esto ocurre gracias a un conjunto de técnicas, tales como: herencia, cohesión, abstracción, polimorfismo, acoplamiento y encapsulamiento. Las cuales permiten incrementar enormemente el proceso de producción de software (Morero, 2000).

En cuanto al diagramado, se utilizó el lenguaje unificado de modelado (UML, por sus siglas en inglés, Unified Modeling Language) el cual está compuesto por diversos elementos gráficos que se combinan. Debido a que el UML es un lenguaje, cuenta con reglas para combinar tales elementos. La finalidad de los diagramas es presentar las

diversas perspectivas del sistema, a los cuales se les conoce como modelo. Un modelo es una representación simplificada de la realidad, que describe lo que hará el sistema, pero no dice cómo se implementará.

4.3.3. Estándares de documentación:

Para la documentación de la toma de requerimientos se utilizará una modificación del estándar IEEE 830-1998 y para la etapa de pruebas, se utilizará una adaptación basada en el estándar IEEE Software Test Documentation Std 829-1998, con el fin de obtener un documento, consistente y correcto.

4.3.4. Herramientas de apoyo a la planificación:

- **Gantt Project:** programa de código abierto para la administración de proyectos usando el diagrama de Gantt.
- **Trello:** Gestor de proyectos online que permite aclarar rutinas de trabajo. En este caso particular, se utilizó como pizarra para la metodología Kanban.

4.3.5. Herramientas de apoyo al desarrollo del software:

- **Android Studio:** Entorno de desarrollo integrado oficial para la plataforma Android.
- **Visual Estudio:** Entorno de desarrollo integrado para sistemas operativos Windows.
- **Arduino IDE for Visual Estudio:** Complemento de Visual Estudio para el desarrollo de código Arduino.
- **Java SE Development Kit 8:** Entorno de desarrollo java.
- **Módulo de desarrollo multiplataforma de .NET Core:** Compila aplicaciones multiplataforma con .NET Core, ASP.NET.
- **SQL:** Lenguaje que permite ejecutar operaciones sobre la base de datos.
- **Kestrel:** Es un Servidor web multiplataforma para ASP.NET Core (redes internas).
- **Apache:** Es un servidor web HTTP de código abierto y multiplataforma.

4.3.6. Herramientas al diagramado:

- **StarUML:** Editor para diagramas de estructura estáticos UML.
- **DataGrip:** Entorno de desarrollo SQL con opciones de diagramado.
- **Fritzing:** Es un programa libre de automatización de diseño electrónico.

4.3.7. Hardware de uso:

- Computador portátil con especificaciones básicas y conexión en red.

4.3.8. Hardware de desarrollo:

- VPS (Virtual Private Server) arrendado a Opencloud.cl.

4.3.9. Lenguajes de programación:

- C Sharp (web API)
- Java8 (aplicación Android)
- C++ (código Arduino)

4.3.10. Protocolos de comunicación:

- TCP
- NTP
- UDP
- MQTT

4.3.11. Gestor de base de datos:

- MySQL

4.3.12. Nombre de la base de datos:

- Proyecto

4.3.13. Dirección del servidor:

- 190.114.253.121

4.3.14. Framework:

- .NET Framework

4.4. Definiciones, siglas y abreviaciones

IDE (entorno de desarrollo integrado): Aplicación informática que proporciona servicios integrales que facilitan el desarrollo de software.

Protocolo: En informática se refiere a un conjunto de reglas predefinidas con el propósito de estandarizar el intercambio de información en actividades informáticas.

TCP (Protocolo de Control de Transmisión): Es uno de los principales protocolos de la capa de transporte del modelo TCP/IP.

MQTT: Protocolo de transporte de mensajes Cliente/Servidor ligero, basado en publicaciones y suscripciones a los denominados tópicos.

NTP (Protocolo de tiempo de red): Es un protocolo de Internet para sincronizar los relojes de los sistemas informáticos a través del enrutamiento de paquetes en redes con latencia variable. NTP utiliza UDP como su capa de transporte, usando el puerto 123.

UDP (Protocolo de datagramas de usuario): Permite el envío de datagramas a través de la red sin que se haya establecido previamente una conexión.

Puerto: Un puerto es una interfaz a través de la cual se pueden enviar y recibir los diferentes tipos de datos.

JSON (Notación de Objetos de JavaScript): Es un formato ligero de intercambio de datos.

Web API: Conjunto de reglas y especificaciones que las aplicaciones pueden seguir para comunicarse entre ellas.

Patrón de publicación/suscripción: Corresponde a la forma de interactuar con el servidor MQTT, en donde un cliente se suscribe a un tema(asunto) y puede realizar las tareas de publicación y escucha de mensajes

Eclipse Paho: Biblioteca para desarrollar un cliente MQTT en Android.

MQTTNet: Biblioteca .NET de alto rendimiento para la comunicación basada en MQTT. Proporciona un cliente MQTT y un servidor MQTT.

Servicio: Proceso que se ejecuta en segundo plano del sistema operativo.

Gson: Es una biblioteca de código abierto para el lenguaje de programación Java que permite la serialización y deserialización entre objetos Java y su representación en notación JSON.

ArduinoJson: Corresponde a una biblioteca de C ++ JSON para Arduino y IoT (Internet de las cosas).

PubSubClient: Es una biblioteca cliente para el Arduino que proporciona soporte para MQTT.

NTPClient: Biblioteca para conectarse a un servidor NTP y obtener la fecha y tiempo en formato Epoch.

RtcDS1302: Biblioteca para configurar el módulo DS1302 RTC.

Switch (Android): Es un widget conmutador de dos estados que puede seleccionar entre dos opciones.

Layout (Android): Define la estructura visual para una interfaz de usuario, como la IU para una actividad o widget de una app.

TimePicker (Android): Es un widget para seleccionar la hora del día, en modo de 24 horas o AM / PM.

EditText (Android): Es un elemento de la interfaz de usuario para ingresar y modificar texto.

Arduino: Es una tarjeta electrónica programable que permite la creación rápida de prototipos. Su diseño es de código abierto (open-source).

ASP.NET core: Framework compacto, multiplataforma y fácil de usar para crear aplicaciones web y en la nube.

DataGrip: Entorno de desarrollo SQL con opciones de diagramado.

Json Web Token: Los tokens web de JSON son un estándar abierto de la industria RFC7519 método para representar reclamos de forma segura entre dos partes.

Relé: Dispositivo electromagnético que es estimulado por una corriente eléctrica muy débil, este abre o cierra un circuito en el cual se disipa una potencia mayor que en el circuito estimulador.

Ecosistema domótico: Conformado por un Servidor MQTT, una aplicación Android y un módulo domótico.

Computación física: Se centra en diseñar dispositivos, objetos e incluso entornos que permitan establecer un canal de comunicación entre el mundo físico y el mundo virtual.

Serializar: Proceso de codificación de un objeto en un medio de almacenamiento, con el fin de transmitirlo a través de una conexión en red como una serie de bytes o en un formato humanamente más legible como XML o JSON.

Deserializar: Trasformar un formato humanamente legible a un objeto.

RTC (Reloj en tiempo real): Consiste en un circuito integrado alimentado por una batería el cual, en todo momento, registra la fecha, día de la semana y hora al igual que un reloj digital convencional.

DTH22: Sensor digital de temperatura y humedad.

ESP32: Microcontrolador con WIFI y Bluetooth integrado, creado y desarrollado por Espressif Systems.

SQL: Es un lenguaje específico del dominio utilizado en programación, diseñado para administrar, y recuperar información de sistemas de gestión de bases de datos relacionales.

Epoch: Es la hora actual medida en número de segundos desde el Epoch Unix. Epoch 0 es enero 1 1970 00:00:00 GMT (ISO 8601: 1970-01-01T00:00:00Z). No se usan segundos bisiestos en el tiempo Unix.

Multiplataforma: Se refiere a los programas, sistemas operativos, u otra clase de software que pueden funcionar en diversas plataformas.

Modulo domótico/Dispositivo domótico/Dispositivo ESP32: Es el artefacto controlable creado en este proyecto, y se le refiere de estas formas para evitar la redundancia.

Memoria flash SPI (Serial Peripheral Interface): Es un tipo de memoria electrónica no volátil, derivada de las EEPROM, que puede ser borrada y reprogramada electrónicamente (se deteriora con el tiempo).

ORM: Técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y la utilización de una base de datos relacional como motor de persistencia.

Entity Framework Core: EF Core puede servir como un mapeador relacional de objetos (ORM), permitiendo a los desarrolladores .NET trabajar con una base de datos utilizando objetos .NET, y eliminando la necesidad de la mayoría del código de acceso a datos que normalmente necesitan escribir.

Atributos de anotaciones de datos: Las anotaciones son un tipo de etiqueta que puede colocar en una clase o propiedad para especificar los metadatos sobre esa clase o propiedad.

CAPÍTULO 5: ESPECIFICACION DE REQUERIMIENTOS DE SOFTWARE

Adaptación basada en IEEE Software requirements Specifications Std 830-1998.

5.1. Alcances

El Sistema domótico modular con automatización y control de dispositivos eléctricos, basado en NodeMCU-32S, inicialmente estará conformado por un servicio, un web API, una aplicación Android y un Módulo Domótico, los cuales necesitaran de acceso a internet para su funcionamiento e interacción (el puerto de comunicación será el 1883, puerto MQTT por defecto). El ecosistema domótico/IoT será presentado como alternativa de bajo costo para la automatización de las viviendas. Se contempla la creación de un único módulo, pues no se pretende dar soporte a esta versión del proyecto, tampoco se pretende la puesta de este proyecto con un fin comercial (será de uso personal). Este sistema tiene como propósito adquirir el conocimiento necesario para la elaboración de un posterior proyecto que abarca el área de la domótica y el Internet de las Cosas. Por último, se mencionarán los alcances, desde el punto de vista específico de cada aplicación propuesta para la solución, y las limitantes serán expuestas desde un punto de vista general.

5.1.1. Alcances del servicio (Servidor MQTT)

- Correrá en un VSP de la empresa Opencloud.
- Permitir el registro de tópicos.
- Permitir la publicación de mensajes en los tópicos.
- Permitir ver los mensajes de un tópico.
- El servicio será multiplataforma.
- Se ejecutará en segundo plano.
- Estará desarrollado con el Framework .NET (ASP.NET Core, C Sharp).
- Validará a los usuarios, haciendo uso del web API.
- Dará acceso a clientes registrados a través de la aplicación Android.

5.1.2. Alcances web API:

El web API estará desarrollado con el Framework .NET (ASP.NET Core, C Sharp). Este correrá sobre un VPS (Opencloud) y será multiplataforma. También se comunicará con la base de datos del proyecto (motor de bases de datos: MySQL) y contará con validación de usuario mediante JSON web token. Las capacidades de este se reflejan en los siguientes módulos (las peticiones y las respuestas estarán en formato JSON):

- **Módulo Usuarios:** Sistema que maneja el registro de usuarios y los privilegios de estos.
- **Módulo de Dispositivos:** Sistema que registra los nuevos dispositivos domóticos en el sistema.
- **Módulo de Usuarios y Dispositivos:** Sistema que registra los dispositivos que poseen los usuarios.
- **Módulo de Tópicos:** Sistema que registra los tópicos que serán asignados a los módulos. También registra sus tipos de datos.
- **Módulo de tipos de datos:** Sistema que registra los tipos de datos que serán asignados a los tópicos (int, float, char, string, chart, etc.).
- **Módulo Dispositivo y Tópicos:** Sistema que asigna los tópicos a los dispositivos domóticos registrados en el sistema.

5.1.3. Alcances de la aplicación Android

La aplicación Android permitirá crear y validar usuarios. También otorgará control sobre el dispositivo domótico. Esta se dividirá en los siguientes módulos:

- **Módulo Dispositivo:** Actividad encargada de interactuar con el usuario, para el control y automatización del módulo.
- **Módulo MQTT:** Clase encargada de la suscripción y publicación en el servidor MQTT, es la clase que se encarga de interactuar con el dispositivo domótico. La clase escucha constantemente los mensajes publicados en los tópicos suscritos.

- **Modulo Serializador y Deserializador:** Clase encargado de transformar un mensaje codificado en JSON proveniente del dispositivo domótico a su respectivo objeto (siempre y cuando el objeto pertenezca a uno de los modelos de la aplicación). También se encarga de codificar un objeto a formato JSON, que es entendido por el dispositivo domótico.
- **Modulo Programador:** Fragment (panel de Android) dependiente del módulo dispositivos. En Este módulo se podrá configurar el encendido y apagado de cada relé perteneciente al ESP32(Dispositivo domótico).

5.1.4. Alcances del ESP32 (Arduino):

Corresponde al dispositivo programable que será controlado por la aplicación Android. Este posee una memoria flash SPI, por consiguiente, se optó a que los datos de validación con el servidor MQTT y de acceso a internet formen parte del código.

El módulo domótico presentará las siguientes funcionalidades:

- **Conectar a WIFI:** Función encargada de conectar el dispositivo a internet, mediante la tarjeta de red WIFI integrada.
- **Conectar a servidor MQTT:** Función encargada conectar dispositivo al servidor MQTT sobre el protocolo de comunicación TCP. Adicionalmente suscribe el dispositivo a dos canales(tópicos), por los cuales se comunicará el ESP32.
- **Configuración NTP RTC:** Función que se ejecuta durante el encendido, encargada de obtener datos relativos a la zona horaria, en formato Epoch y sincronizar un módulo RTC (Reloj de tiempo real) Para tener control de los días y la hora.
- **Cambiar zona horaria:** Función encargada de conectarse al servidor NTP con una nueva zona horaria y sincronizar el formato de tiempo con el RTC.
- **Escuchar mensajes:** Función encargada de escuchar los mensajes en los tópicos suscritos en servidor MQTT y transformarlos dinámicamente a un objeto, para su posterior lectura. Los tópicos escuchados por el módulo, son los siguientes:

- **Nombre:** Este es un tópico de escucha y respuesta, en donde el ESP32 recibe una solicitud preguntando el nombre del módulo, y este responde a esta petición.
- **Solicitud de ámbito general:** Los mensajes que llegan a este tópico, son deserializados, validados, y respondidos en un tópico exclusivo de respuestas, al cual el ESP32 no está suscrito. Las solicitudes de acción pueden ser las siguientes (solicitud de acción se refiere al atributo llamado acción perteneciente al objeto JSON):
 - **Relé:** El objeto recibido bajo esta acción corresponde a un array de objetos, que indica los relés que serán prendidos o apagados.
 - **Organizador:** Bajo esta acción se define el encendido y apagado automático de los relés.
 - **GTM:** bajo esta acción el dispositivo ESP32 actualiza su zona horaria, conectándose al servidor NTP y estableciendo la hora en el módulo RTC.
 - **Desorganizador:** Bajo esta acción se desactiva el encendido y apagado automático de los relés.
 - **Notificación:** cuando un mensaje arriba al dispositivo con la acción de notificación. Es
- **Obtener hora y minutos:** Función encargada de obtener la hora del módulo RTC en formato 24 horas, y con la siguiente estructura HMM (variable de tipo entero), para comparar si la hora actual corresponde a al encendido o apagado de un relé.
- **Obtener temperatura y humedad:** Porción de código de la función principal que obtiene la temperatura y humedad del sensor DTH22.
- **Alarmas:** Porción de código de la función principal encargado de ver los relés con alarma de encendido activa y proceder con el encendido o apagado, según corresponda.

5.1.5. Limitaciones

El proyecto excluye el uso de todo tipo de frameworks diseñado para el dispositivo ESP32, o para el manejo de los datos sobre el protocolo MQTT. Además, se excluye el uso de páginas o software utilizados para generar aplicaciones móviles mediante **drag and drop**. En este proyecto también se excluye la responsabilidad sobre los siguientes puntos:

- Realizar cambios mayores al software después de su entrega.
- Mantener el proyecto después de su entrega.
- Realizar modificaciones al hardware del módulo domótico.
- Crear más de un módulo domótico.
- Anexar al sistema características adicionales a las presentadas.

5.2. Objetivo del software

5.2.1. Objetivo general

El objetivo del presente desarrollo de software, es exponer un ecosistema Domótico/IoT seguro, que permitirá contralar y automatizar los relés del módulo basado en un ESP32, además de obtener los datos de variables de ambientes, como la temperatura la humedad. Tarea que llevará a cabo la aplicación móvil.

5.2.2. Objetivos específicos

- Mantener un registro en el sistema, de las siguientes entidades: Usuarios, módulos, tópicos y tipos de datos.
- Permitir a los administradores del web API crear, editar y modifica sobre todas las tablas de la base de datos.
- Implementar el ORM Entity Framework Core en el web API, para generar los modelos con sus reglas de validación mediante anotaciones.
- Validar los datos de ingreso, con reglas aplicadas al modelo
- Registrar usuarios en el sistema mediante la aplicación móvil.

- Validar el acceso de los usuarios en el sistema, haciendo uso del estándar JSON web token, para otorgar seguridad y privacidad a estos (los datos de acceso son enviados por la aplicación móvil al web API, y si es correctos, este responderá un token único).
- Permitir al servidor MQTT validar los datos de acceso del módulo domótico, haciendo uso del web API.
- Generar una estructura única, haciendo uso del formato JSON, la cual será enviada por la aplicación móvil, y procesada por el módulo domótico, con el fin de controlar a este último (el módulo domótico ejecutará una acción diferente, dependiendo del mensaje JSON recibido).

5.3. Descripción global del producto

5.3.1. Perspectiva del producto

El SDM (Sistema Domótico Modular) será un ecosistema domótico/IoT y estará diseñado para trabajar en dispositivos móviles con sistema operativo Android, mediante una aplicación móvil, lo que permitirá un cómodo, rápido y eficaz control sobre el módulo domótico (ESP32).

5.3.2. Interfaces de usuario

La interfaz para el usuario consistirá en un conjunto de ventanas con botones, switches, campos de textos y un navigation drawer (sidebar de navegación). Esta deberá ser construida única y exclusivamente para el sistema propuesto, y será visualizada desde dispositivos móviles con sistema operativo Android (versión mayor o igual a Android 4.4, KitKat).

5.3.3. Interfaces de hardware

Será necesario disponer de ordenador que actúe como servidor, en este caso se arrendará un VPS a la empresa Opencloud, las especificaciones de este son las siguientes:

- Disco duro: 1TB

- Procesador: XEON E7
- Trafico: 1TB
- RAM: 1GB

Las especificaciones mínimas para correr el sistema son las siguientes:

- **Procesador:** 1GHZ
- **Disco duro:** 4.5GB
- **RAM:** 512

En cuanto al hardware requerido para el módulo domótico, se detallará a continuación:

- NodeMCU ESP32
- Modulo RTC DS1302
- Módulo de 8 relés
- Módulo DTH22
- Fuente de alimentación YwRobot

5.3.4. Interfaces de software

- **Sistema operativo VPS:** CentOS 7/ Windows Server 2008 SP2 o superior
- **Sistema operativo móvil:** Android 4.4 KitKat o superior (Los usuarios podrán interactuar con el sistema exclusivamente desde la aplicación móvil en cuestión).

5.3.5. Interfaces de comunicación

El sistema al estar compuesto por un web API, un servicio (servidor MQTT), un Módulo domótico y una aplicación Android, necesitará hacer uso de varios protocolos para establecer comunicación entre estos, los cuales se definirán a continuación:

- **HTTP:** Es el protocolo utilizado para la comunicación con el web API
- **TCP/IP:** Esta diseñada para enlazar ordenadores de diferentes tipos que ejecuten sistemas operativos distintos sobre redes de área local y redes de área extensa y, por tanto, permite la conexión de equipos distantes geográficamente.

- **MQTT:** Es un protocolo de mensajería basado en publicación-suscripción. Funciona sobre el protocolo TCP / IP.
- **NTP:** Es un protocolo de Internet para sincronizar los relojes de los sistemas informáticos (para sincronizar el RTC).

5.3.6. Características de los usuarios

| | |
|-----------------------------------|--|
| Tipo de usuario | Administrador. |
| Formación mínima requerida | Analista programador. |
| Característica | El Administrador debe tener conocimiento total de la nomenclatura utilizada en el desarrollo del proyecto. También debe poseer dominio sobre el marco de desarrollo .NET, específicamente sobre el desarrollo de web API y servicios, mediante el lenguaje de programación C Sharp (C#). |
| Actividad | Control y manejo del sistema en general. |

Tabla 5-1 Características que debe poseer un administrador

| | |
|-----------------------------------|--|
| Tipo de usuario | Usuario de la Aplicación móvil. |
| Formación mínima requerida | Ninguna. |
| Característica | El usuario debe poseer un dispositivo Android. También debe tener las facultades físicas y cognitivas para poder controlar la aplicación móvil |
| Actividad | Control sobre el módulo domótico, haciendo uso de la aplicación Android. |

Tabla 5-2 Características que debe poseer un usuario de la aplicación móvil

5.3.7. Restricciones (complemento a los alcances)

- Interfaz para ser usada mediante conexión a Internet.
- El servidor carece de dominio (IP: 190.114.253.121).
- Lenguajes y tecnologías en uso: Java8, C Sharp y C++.
- El servidor y el web API deben ser capaces de atender consultas concurrentes.
- El sistema se diseñará según un modelo cliente/servidor.

5.3.8. Suposiciones y dependencias

- Se asume que se cumplen las especificaciones mínimas de hardware y software
- Los dispositivos móviles Android en los que se vaya a ejecutar la aplicación, deben poseer una versión superior a Android 4.4 KitKat.

5.4. Requerimientos Específicos

Esta sección tiene como objetivo mostrar detalladamente los requerimientos funcionales y no funcionales (los requerimientos funcionales también son conocidos como atributos de calidad de un sistema) identificados.

5.4.1. Requerimientos funcionales

5.4.1.1. Requerimientos funcionales del servidor

A continuación, se definirán una serie de requerimientos funcionales que corresponden al servidor del ecosistema domótico. Estos corresponderán a las funcionalidades que debe otorgar el web API para interactuar con la base de datos y a las funcionalidades que debe proveer el servidor MQTT.

5.4.1.1.1. Requerimientos funcionales generales del servidor

| | |
|--|--|
| Identificación del requerimiento: | RF_01 |
| Nombre del requerimiento: | Iniciar sesión |
| Características: | El sistema aceptara la petición de usuarios para que estos ingresen al sistema |
| Descripción del requerimiento: | El sistema controlará el acceso, y lo permitirá únicamente a usuarios autorizados. Los usuarios deben acceder al sistema ingresando un nombre de usuario y contraseña. |
| Requerimiento no funcional: | <ul style="list-style-type: none"> • RNF_01 • RNF_02 • RNF_04 • RNF_08 • RNF_09 • RNF_10 • RNF_11 |
| Interfaz externa asociada: | No aplica |
| Prioridad del requerimiento: | Alta |

Tabla 5-3 Requerimiento funcional 01

| | |
|--|---|
| Identificación del requerimiento: | RF_02 |
| Nombre del requerimiento: | Salir del sistema. |
| Características: | El sistema permitirá la desconexión de usuarios. |
| Descripción del requerimiento: | El sistema permitirá que los usuarios se desconecten en cualquier momento (La |

| | |
|---|--|
| | desconexión corresponde a que los usuarios no realicen peticiones HTTP). |
| Requerimiento no funcional: | <ul style="list-style-type: none"> • RNF_01 • RNF_02 |
| Interfaz externa asociada: | No aplica |
| Prioridad del requerimiento: Alta | |

Tabla 5-4 Requerimiento funcional 02

| | |
|---|--|
| Identificación del requerimiento: | RF_03 |
| Nombre del requerimiento: | Navegación segura. |
| Características: | La interacción de los usuarios con el sistema debe estar cifrada. |
| Descripción del requerimiento: | Cualquier intercambio de datos vía Internet que realice el software se realizará por medio del protocolo encriptado https. |
| Requerimiento no funcional: | <ul style="list-style-type: none"> • RNF_01 • RNF_02 • RNF_04 • RNF_11 • RNF_14 |
| Interfaz externa asociada: | No aplica |
| Prioridad del requerimiento: Alta | |

Tabla 5-5 Requerimiento funcional 03

| | |
|--|--|
| Identificación del requerimiento: | RF_04 |
| Nombre del requerimiento: | Publicación de mensajes en tópicos |
| Características: | El servidor será capaz de publicar los mensajes enviados por los clientes, en un tópico determinado. |
| Descripción del requerimiento: | El servidor enviará mensajes a los clientes sobre el protocolo MQTT, con el propósito de que el módulo domótico ejecute una acción |
| Requerimiento no funcional: | <ul style="list-style-type: none"> • RNF_01 • RNF_02 • RNF_03 • RNF_04 • RNF_08 • RNF_11 |
| Interfaz externa asociada: | No aplica |
| Prioridad del requerimiento: | Alta |

Tabla 5-6 Requerimiento funcional 04

| | |
|--|--|
| Identificación del requerimiento: | RF_05 |
| Nombre del requerimiento: | Suscripción a tópicos |
| Características: | El servidor debe ser capaz de enviar mensajes a los usuarios suscritos en determinadas temáticas |
| Descripción del requerimiento: | El servidor debe ser capaz de recibir los mensajes enviados por los clientes a un |

| | |
|-------------------------------------|--|
| | tópico, y debe reenviarlo a todos los clientes que estén suscritos a la temática. |
| Requerimiento no funcional: | <ul style="list-style-type: none"> • RNF_01 • RNF_02 • RNF_03 • RNF_04 • RNF_08 • RNF_11 |
| Interfaz externa asociada: | No aplica |
| Prioridad del requerimiento: | |
| Alta | |

Tabla 5-7 Requerimiento funcional 05

5.4.1.1.2. Maestro de usuarios

| | |
|--|--|
| Identificación del requerimiento: | RF_06 |
| Nombre del requerimiento: | Crear usuario. |
| Características: | El sistema aceptará la creación de nuevos usuarios. |
| Descripción del requerimiento: | <p>Crear usuario mediante la aplicación móvil: La persona que haga uso de la aplicación móvil deberá registrar una nueva cuenta, mediante el uso de esta y automáticamente se le asignará el rol normal</p> <p>Administrador crea usuario: un administrador posee la capacidad de crear cuentas con rol normal o rol de administrador</p> |
| Requerimiento no funcional: | <ul style="list-style-type: none"> • RNF_01 • RNF_02 • RNF_03 • RNF_04 • RNF_11 • RNF_12 |
| Interfaz externa asociada: | INT_ENT_01 |
| Prioridad del requerimiento: | Alta |

Tabla 5-8 Requerimiento funcional 06

| | |
|--|---|
| Identificación del requerimiento: | RF_07 |
| Nombre del requerimiento: | Modificar usuario. |
| Características: | El sistema aceptará la modificación de los datos de los usuarios. |

| | |
|--|--|
| Descripción del requerimiento: | El administrador tendrá derechos para actualizar los datos de los usuarios con rol normal |
| Requerimiento no funcional: | <ul style="list-style-type: none"> • RNF_01 • RNF_02 • RNF_03 • RNF_04 • RNF_11 • RNF_12 |
| Interfaz externa asociada: | <ul style="list-style-type: none"> • INT_ENT_01 • INT_SAL_07 |
| Prioridad del requerimiento: Media | |

Tabla 5-9 Requerimiento funcional 07

| | |
|--|--|
| Identificación del requerimiento: | RF_08 |
| Nombre del requerimiento: | Eliminar Usuario. |
| Características: | El sistema aceptará la eliminación de los usuarios. |
| Descripción del requerimiento: | El administrador tendrá derechos para borrar a los usuarios con rol normal. |
| Requerimiento no funcional: | <ul style="list-style-type: none"> • RNF_01 • RNF_02 • RNF_03 • RNF_04 • RNF_11 |
| Interfaz externa asociada: | <ul style="list-style-type: none"> • INT_SAL_07 |

| |
|--|
| <p>Prioridad del requerimiento: Media</p> |
|--|

Tabla 5-10 Requerimiento funcional 08

| | |
|--|--|
| Identificación del requerimiento: | RF_09 |
| Nombre del requerimiento: | Listar usuario. |
| Características: | El sistema podrá listar a los usuarios registrados en este. |
| Descripción del requerimiento: | El administrador podrá visualizar una lista de los usuarios registrados en el sistema. |
| Requerimiento no funcional: | <ul style="list-style-type: none"> • RNF_01 • RNF_02 • RNF_03 • RNF_04 • RNF_11 |
| Interfaz externa asociada: | <ul style="list-style-type: none"> • INT_SAL_01 |
| <p>Prioridad del requerimiento: Media</p> | |

Tabla 5-11 Requerimiento funcional 09

| | |
|--|--|
| Identificación del requerimiento: | RF_10 |
| Nombre del requerimiento: | Buscar usuario. |
| Características: | El sistema permitirá buscar usuarios registrados. |
| Descripción del requerimiento: | El administrador podrá realizar la búsqueda de un usuario en específico. |
| Requerimiento no funcional: | <ul style="list-style-type: none"> • RNF_01 |

| | |
|---|--|
| | <ul style="list-style-type: none"> • RNF_02 • RNF_03 • RNF_04 • RNF_11 |
| Interfaz externa asociada: | <ul style="list-style-type: none"> • INT_SAL_07 |
| Prioridad del requerimiento: Alta | |

Tabla 5-12 Requerimiento funcional 10

5.4.1.1.3. Maestro de módulos

| | |
|---|--|
| Identificación del requerimiento: | RF_11 |
| Nombre del requerimiento: | Crear módulo. |
| Características: | El sistema aceptará la creación de nuevos módulos. |
| Descripción del requerimiento: | El administrador tiene la facultad de ingresar nuevos módulos al sistema. |
| Requerimiento no funcional: | <ul style="list-style-type: none"> • RNF_01 • RNF_02 • RNF_03 • RNF_04 • RNF_11 |
| Interfaz externa asociada: | <ul style="list-style-type: none"> • INT_ENT_02 |
| Prioridad del requerimiento: Alta | |

Tabla 5-13 Requerimiento funcional 11

| | |
|--|--|
| Identificación del requerimiento: | RF_12 |
| Nombre del requerimiento: | Modificar módulo. |
| Características: | El sistema aceptará la modificación de los datos de los módulos. |
| Descripción del requerimiento: | El administrador tendrá los permisos para actualizar los datos de los módulos. |
| Requerimiento no funcional: | <ul style="list-style-type: none"> • RNF_01 • RNF_02 • RNF_03 • RNF_04 |

| | |
|-------------------------------------|--|
| | <ul style="list-style-type: none"> • RNF_11 |
| Interfaz externa asociada: | <ul style="list-style-type: none"> • INT_ENT_02 • INT_SAL_08 |
| Prioridad del requerimiento: | |
| Media | |

Tabla 5-14 Requerimiento funcional 12

| | |
|--|--|
| Identificación del requerimiento: | RF_13 |
| Nombre del requerimiento: | Eliminar módulo. |
| Características: | El sistema aceptará la eliminación de los módulos. |
| Descripción del requerimiento: | El administrador tendrá derechos para borrar a los usuarios con rol normal. |
| Requerimiento no funcional: | <ul style="list-style-type: none"> • RNF_01 • RNF_02 • RNF_03 • RNF_04 • RNF_11 |
| Interfaz externa asociada: | <ul style="list-style-type: none"> • INT_SAL_08 |
| Prioridad del requerimiento: | |
| Media | |

Tabla 5-15 Requerimiento funcional 13

| | |
|--|--|
| Identificación del requerimiento: | RF_14 |
| Nombre del requerimiento: | Listar módulo. |
| Características: | El sistema podrá listar los módulos registrados en este. |

| | |
|--|--|
| Descripción del requerimiento: | El administrador podrá visualizar una lista de los módulos registrados en el sistema. |
| Requerimiento no funcional: | <ul style="list-style-type: none"> • RNF_01 • RNF_02 • RNF_03 • RNF_04 • RNF_11 |
| Interfaz externa asociada: | <ul style="list-style-type: none"> • INT_SAL_02 |
| Prioridad del requerimiento: Media | |

Tabla 5-16 Requerimiento funcional 14

| | |
|--|--|
| Identificación del requerimiento: | RF_15 |
| Nombre del requerimiento: | Buscar módulo. |
| Características: | El sistema permitirá buscar módulos registrados. |
| Descripción del requerimiento: | El administrador podrá realizar la búsqueda de un módulo en específico. |
| Requerimiento no funcional: | <ul style="list-style-type: none"> • RNF_01 • RNF_02 • RNF_03 • RNF_04 • RNF_11 |
| Interfaz externa asociada: | <ul style="list-style-type: none"> • INT_SAL_08 |
| Prioridad del requerimiento: Media | |

Tabla 5-17 Requerimiento funcional 15

5.4.1.1.4. Maestro tipos de datos

| | |
|--|--|
| Identificación del requerimiento: | RF_16 |
| Nombre del requerimiento: | Crear tipo de dato. |
| Características: | El sistema permitirá crear nuevos tipos de datos |
| Descripción del requerimiento: | El administrador tiene la facultad de ingresar nuevos tipos de datos al sistema. |
| Requerimiento no funcional: | <ul style="list-style-type: none"> • RNF_01 • RNF_02 • RNF_03 • RNF_04 • RNF_11 |
| Interfaz externa asociada: | <ul style="list-style-type: none"> • INT_ENT_04 |
| Prioridad del requerimiento: | Alta |

Tabla 5-18 Requerimiento funcional 16

| | |
|--|--|
| Identificación del requerimiento: | RF_17 |
| Nombre del requerimiento: | Modificar tipo de dato. |
| Características: | El sistema aceptará la modificación de los tipos de datos. |
| Descripción del requerimiento: | El administrador tendrá derechos para actualizar los tipos de datos ingresados en el sistema. |
| Requerimiento no funcional: | <ul style="list-style-type: none"> • RNF_01 • RNF_02 • RNF_03 |

| | |
|-------------------------------------|--|
| | <ul style="list-style-type: none"> • RNF_04 • RNF_11 |
| Interfaz externa asociada: | <ul style="list-style-type: none"> • INT_ENT_04 • INT_SAL_09 |
| Prioridad del requerimiento: | |
| Media | |

Tabla 5-19 Requerimiento funcional 17

| | |
|--|--|
| Identificación del requerimiento: | RF_18 |
| Nombre del requerimiento: | Eliminar tipo de dato. |
| Características: | El sistema aceptará la eliminación de los tipos de datos. |
| Descripción del requerimiento: | El administrador tendrá la facultad para borrar a los tipos de datos. |
| Requerimiento no funcional: | <ul style="list-style-type: none"> • RNF_01 • RNF_02 • RNF_03 • RNF_04 • RNF_11 |
| Interfaz externa asociada: | <ul style="list-style-type: none"> • INT_SAL_09 |
| Prioridad del requerimiento: | |
| Media | |

Tabla 5-20 Requerimiento funcional 18

| | |
|--|--|
| Identificación del requerimiento: | RF_19 |
| Nombre del requerimiento: | Listar tipo de dato. |
| Características: | El sistema podrá listar a los tipos de datos registrados en este. |
| Descripción del requerimiento: | El administrador podrá visualizar una lista de los tipos de datos registrados en el sistema. |
| Requerimiento no funcional: | <ul style="list-style-type: none"> • RNF_01 • RNF_02 • RNF_03 • RNF_04 • RNF_11 |
| Interfaz externa asociada: | <ul style="list-style-type: none"> • INT_SAL_04 |
| Prioridad del requerimiento: | Media |

Tabla 5-21 Requerimiento funcional 19

| | |
|--|--|
| Identificación del requerimiento: | RF_20 |
| Nombre del requerimiento: | Buscar tipo de dato. |
| Características: | El sistema permitirá buscar entre los tipos de datos registrados en este. |
| Descripción del requerimiento: | El administrador podrá realizar la búsqueda de un tipo de dato en específico. |
| Requerimiento no funcional: | <ul style="list-style-type: none"> • RNF_01 • RNF_02 • RNF_03 • RNF_04 • RNF_11 |

| | |
|-------------------------------------|--|
| Interfaz externa asociada: | <ul style="list-style-type: none"> • INT_SAL_09 |
| Prioridad del requerimiento: | |
| Media | |

Tabla 5-22 Requerimiento funcional 20

5.4.1.1.5. Maestros tópicos

| | |
|--|--|
| Identificación del requerimiento: | RF_21 |
| Nombre del requerimiento: | Crear tópicos. |
| Características: | El sistema permitirá registro de nuevos tópicos. |
| Descripción del requerimiento: | El administrador podrá ingresar tópicos al sistema. |
| Requerimiento no funcional: | <ul style="list-style-type: none"> • INT_ENT_03 |
| Interfaz externa asociada: | <ul style="list-style-type: none"> • RNF_01 • RNF_02 • RNF_03 • RNF_04 • RNF_11 |
| Prioridad del requerimiento: | |
| Alta | |

Tabla 5-23 Requerimiento funcional 21

| | |
|--|--|
| Identificación del requerimiento: | RF_22 |
| Nombre del requerimiento: | Modificar tópicos. |
| Características: | El sistema soportará la modificación de los tópicos registrados en este. |

| | |
|--|--|
| Descripción del requerimiento: | El administrador tendrá el permiso para actualizar los datos de los tópicos registrados en el sistema. |
| Requerimiento no funcional: | <ul style="list-style-type: none"> • INT_ENT_03 • INT_SAL_10 |
| Interfaz externa asociada: | <ul style="list-style-type: none"> • RNF_01 • RNF_02 • RNF_03 • RNF_04 • RNF_11 |
| Prioridad del requerimiento: Media | |

Tabla 5-24 Requerimiento funcional 22

| | |
|--|--|
| Identificación del requerimiento: | RF_23 |
| Nombre del requerimiento: | Eliminar tópicos. |
| Características: | El sistema podrá listar a los tópicos registrados en este. |
| Descripción del requerimiento: | El administrador podrá visualizar una lista de los tópicos registrados en el sistema. |
| Requerimiento no funcional: | <ul style="list-style-type: none"> • INT_SAL_10 |
| Interfaz externa asociada: | <ul style="list-style-type: none"> • RNF_01 • RNF_02 • RNF_03 • RNF_04 • RNF_11 |
| Prioridad del requerimiento: | |

| |
|-------|
| Media |
|-------|

Tabla 5-25 Requerimiento funcional 23

| | |
|--|--|
| Identificación del requerimiento: | RF_24 |
| Nombre del requerimiento: | Listar tópicos. |
| Características: | El sistema entregará una lista de los tópicos ingresados en este. |
| Descripción del requerimiento: | El administrador del sistema podrá visualizar la lista de los tópicos que han sido ingresados al sistema. |
| Requerimiento no funcional: | <ul style="list-style-type: none"> • RNF_01 • RNF_02 • RNF_03 • RNF_04 • RNF_11 |
| Interfaz externa asociada: | <ul style="list-style-type: none"> • INT_SAL_03 |
| Prioridad del requerimiento: | Media |

Tabla 5-26 Requerimiento funcional 24

| | |
|--|---|
| Identificación del requerimiento: | RF_25 |
| Nombre del requerimiento: | Buscar tópicos. |
| Características: | El sistema permitirá realizar una búsqueda de los tópicos ingresados en este. |
| Descripción del requerimiento: | El administrador del sistema podrá realizar una búsqueda sobre los tópicos registrados. |
| Requerimiento no funcional: | <ul style="list-style-type: none"> • RNF_01 |

| | |
|--|---|
| | <ul style="list-style-type: none">• RNF_02• RNF_03• RNF_04• RNF_11 |
| Interfaz externa asociada: | <ul style="list-style-type: none">• INT_SAL_10 |
| Prioridad del requerimiento: Media | |

Tabla 5-27 Requerimiento funcional 25

5.4.1.1.6. Relación usuario - módulos

| | |
|--|--|
| Identificación del requerimiento: | RF_26 |
| Nombre del requerimiento: | Asignar módulos a usuarios. |
| Características: | El sistema permitirá asignar módulos a los usuarios registrados (relación de muchos a muchos). |
| Descripción del requerimiento: | Cuando los usuarios inicien sesión en la aplicación móvil, automáticamente asignara el módulo controlado a su cuenta. |
| Requerimiento no funcional: | <ul style="list-style-type: none"> • RNF_01 • RNF_02 • RNF_03 • RNF_04 • RNF_11 |
| Interfaz externa asociada: | <ul style="list-style-type: none"> • INT_ENT_05 |
| Prioridad del requerimiento: | Alta |

Tabla 5-28 Requerimiento funcional 26

| | |
|--|--|
| Identificación del requerimiento: | RF_27 |
| Nombre del requerimiento: | Modificar módulos asignados a usuarios. |
| Características: | El sistema permitirá modificar los módulos asignados a los usuarios. |
| Descripción del requerimiento: | El administrador tendrá los permisos para modificar los módulos asignados a los usuarios |
| Requerimiento no funcional: | <ul style="list-style-type: none"> • RNF_01 • RNF_02 |

| | |
|---|--|
| | <ul style="list-style-type: none"> • RNF_03 • RNF_04 • RNF_11 |
| Interfaz externa asociada: | <ul style="list-style-type: none"> • INT_ENT_05 • INT_SAL_08 |
| Prioridad del requerimiento: Baja | |

Tabla 5-29 Requerimiento funcional 27

| | |
|---|--|
| Identificación del requerimiento: | RF_28 |
| Nombre del requerimiento: | Eliminar módulos asignados a usuarios. |
| Características: | El sistema permitirá eliminar los módulos asignados a los usuarios. |
| Descripción del requerimiento: | El administrador tendrá los permisos para eliminar los módulos asignados a los usuarios |
| Requerimiento no funcional: | <ul style="list-style-type: none"> • RNF_01 • RNF_02 • RNF_03 • RNF_04 • RNF_11 |
| Interfaz externa asociada: | <ul style="list-style-type: none"> • INT_SAL_08 |
| Prioridad del requerimiento: Baja | |

Tabla 5-30 Requerimiento funcional 28

| | |
|--|--|
| Identificación del requerimiento: | RF_29 |
| Nombre del requerimiento: | Listar módulos asignados a usuario. |
| Características: | El sistema permitirá listar los módulos asignados a los usuarios. |
| Descripción del requerimiento: | El administrador podrá visualizar la lista de módulos asignados a un usuario en específico. |
| Requerimiento no funcional: | <ul style="list-style-type: none"> • RNF_01 • RNF_02 • RNF_03 • RNF_04 • RNF_11 |
| Interfaz externa asociada: | <ul style="list-style-type: none"> • INT_SAL_05 |
| Prioridad del requerimiento: | Alta |

Tabla 5-31 Requerimiento funcional 29

5.4.1.1.7. Relación módulo - tópicos

| | |
|---|--|
| Identificación del requerimiento: | RF_30 |
| Nombre del requerimiento: | Asignar tópicos a módulo. |
| Características: | El sistema permitirá asignar tópicos a los módulos registrados en este (relación de muchos a muchos). |
| Descripción del requerimiento: | El administrador tendrá la facultad de asignar tópicos a los módulos del sistema. |
| Requerimiento no funcional: | <ul style="list-style-type: none"> • RNF_01 • RNF_02 • RNF_03 • RNF_04 • RNF_11 |
| Interfaz externa asociada: | <ul style="list-style-type: none"> • INT_ENT_06 |
| Prioridad del requerimiento: Alta | |

Tabla 5-32 Requerimiento funcional 30

| | |
|--|---|
| Identificación del requerimiento: | RF_31 |
| Nombre del requerimiento: | Modificar tópicos asignado a módulo. |
| Características: | El sistema permitirá modificar los tópicos asignados a los módulos. |
| Descripción del requerimiento: | El administrador del sistema tendrá el permiso para modificar los tópicos asignado a un módulo. |
| Requerimiento no funcional: | <ul style="list-style-type: none"> • RNF_01 • RNF_02 |

| | |
|--|--|
| | <ul style="list-style-type: none"> • RNF_03 • RNF_04 • RNF_11 |
| Interfaz externa asociada: | <ul style="list-style-type: none"> • INT_ENT_06 • INT_SAL_10 |
| Prioridad del requerimiento: Media | |

Tabla 5-33 Requerimiento funcional 31

| | |
|---|--|
| Identificación del requerimiento: | RF_32 |
| Nombre del requerimiento: | Eliminar tópicos asignados a módulo |
| Características: | El sistema aceptará la eliminación de los tópicos asignados a los módulos. |
| Descripción del requerimiento: | El administrador del sistema, haciendo uso de un cliente HTTP podrá eliminar los tópicos asignados a un módulo. |
| Requerimiento no funcional: | <ul style="list-style-type: none"> • RNF_01 • RNF_02 • RNF_03 • RNF_04 • RNF_11 |
| Interfaz externa asociada: | <ul style="list-style-type: none"> • INT_SAL_10 |
| Prioridad del requerimiento: Baja | |

Tabla 5-34 Requerimiento funcional 32

| | |
|--|--|
| Identificación del requerimiento: | RF_33 |
| Nombre del requerimiento: | Listar tópicos asignado a módulo. |
| Características: | El sistema permitirá listar los tópicos asignado a un módulo. |
| Descripción del requerimiento: | El administrador podrá visualizar la lista de tópicos asignados a un módulo en específico. |
| Requerimiento no funcional: | <ul style="list-style-type: none"> • RNF_01 • RNF_02 • RNF_03 • RNF_04 • RNF_11 |
| Interfaz externa asociada: | <ul style="list-style-type: none"> • INT_SAL_06 |
| Prioridad del requerimiento: | Baja |

Tabla 5-35 Requerimiento funcional 33

5.4.1.1.8. Relación tópico - tipo de dato

| | |
|--|---|
| Identificación del requerimiento: | RF_34 |
| Nombre del requerimiento: | Asignar tipo de dato a tópico. |
| Características: | El sistema permitirá asignar tipos de datos a los tópicos registrados en este (relación de uno a muchos). |
| Descripción del requerimiento: | El administrador podrá registrar solo un tipo de dato por tópico. Para la asignación ver RF_19 |
| Requerimiento no funcional: | <ul style="list-style-type: none"> • RNF_01 • RNF_02 |

| | |
|---|--|
| | <ul style="list-style-type: none"> • RNF_03 • RNF_04 • RNF_11 |
| Interfaz externa asociada: | <ul style="list-style-type: none"> • INT_ENT_03(Solo el campo DataTypeId) |
| Prioridad del requerimiento: Alta | |

Tabla 5-36 Requerimiento funcional 34

| | |
|---|--|
| Identificación del requerimiento: | RF_35 |
| Nombre del requerimiento: | Modificar tipo de dato asignado a tópico. |
| Características: | El sistema permitirá modificar el tipo de dato asignado a un tópico. |
| Descripción del requerimiento: | El administrador tendrá los derechos para modificar un tipo de dato asignado a un tópico. |
| Requerimiento no funcional: | <ul style="list-style-type: none"> • RNF_01 • RNF_02 • RNF_03 • RNF_04 • RNF_11 |
| Interfaz externa asociada: | <ul style="list-style-type: none"> • INT_ENT_03(Solo el campo DataTypeId) |
| Prioridad del requerimiento: Baja | |

Tabla 5-37 Requerimiento funcional 35

5.4.1.2. Requerimientos funcionales de la aplicación Android

| | |
|--|--|
| Identificación del requerimiento: | RF_36 |
| Nombre del requerimiento: | Registrar usuario. |
| Características: | El sistema enviará datos de registro al servidor. |
| Descripción del requerimiento: | El sistema permitirá crear cuentas de usuarios y dará acceso a estos para interactuar con las demás interfaces. |
| Requerimiento no funcional: | <ul style="list-style-type: none"> • RNF_01 • RNF_02 • RNF_03 • RNF_04 • RNF_05 • RNF_06 • RNF_07 • RNF_08 |
| Interfaz externa asociada: | <ul style="list-style-type: none"> • INT_ENT_01 |
| Prioridad del requerimiento: | Alta |

Tabla 5-38 Requerimiento funcional 36

| | |
|--|--|
| Identificación del requerimiento: | RF_37 |
| Nombre del requerimiento: | Respuesta de registro. |
| Características: | El sistema debe recibir la respuesta a la solicitud de registro de usuario. |
| Descripción del requerimiento: | El sistema dará una respuesta visual a los usuarios, mediante un cambio en el panel de navegación. |

| | |
|-------------------------------------|--|
| Requerimiento no funcional: | <ul style="list-style-type: none"> • RNF_01 • RNF_02 • RNF_03 • RNF_04 • RNF_05 • RNF_06 • RNF_07 • RNF_08 • RNF_11 |
| Interfaz externa asociada: | No aplica |
| Prioridad del requerimiento: | |
| Alta | |

Tabla 5-39 Requerimiento funcional 37

| | |
|--|--|
| Identificación del requerimiento: | RF_38 |
| Nombre del requerimiento: | Iniciar sesión. |
| Características: | El sistema debe enviar el nombre de usuario y la contraseña al servidor. |
| Descripción del requerimiento: | El sistema permitirá enviar los datos del usuario al servidor para iniciar sesión. |
| Requerimiento no funcional: | <ul style="list-style-type: none"> • RNF_01 • RNF_02 • RNF_03 • RNF_04 • RNF_05 • RNF_06 • RNF_07 |

| | |
|---|--|
| | <ul style="list-style-type: none"> • RNF_08 • RNF_11 |
| Interfaz externa asociada: | No aplica |
| Prioridad del requerimiento: Alta | |

Tabla 5-40 Requerimiento funcional 38

| | |
|---|--|
| Identificación del requerimiento: | RF_39 |
| Nombre del requerimiento: | Respuesta de inicio de sesión. |
| Características: | El sistema debe recibir la respuesta a la solicitud de inicio de sesión. |
| Descripción del requerimiento: | El sistema dará una respuesta visual a los usuarios, mediante un cambio en el panel de navegación. |
| Requerimiento no funcional: | <ul style="list-style-type: none"> • RNF_01 • RNF_02 • RNF_03 • RNF_04 • RNF_05 • RNF_06 • RNF_07 • RNF_08 • RNF_11 |
| Interfaz externa asociada: | No aplica |
| Prioridad del requerimiento: Alta | |

Tabla 5-41 Requerimiento funcional 39

| | |
|---|--|
| Identificación del requerimiento: | RF_40 |
| Nombre del requerimiento: | Vincular módulo. |
| Características: | El sistema enviar los datos de vinculación del usuario con el módulo al servidor. |
| Descripción del requerimiento: | Cuando los usuarios cuando controlen el módulo por primera vez, este automáticamente se relacionará al usuario. |
| Requerimiento no funcional: | <ul style="list-style-type: none"> • RNF_01 • RNF_02 • RNF_08 • RNF_11 |
| Interfaz externa asociada: | <ul style="list-style-type: none"> • INT_ENT_05 |
| Prioridad del requerimiento: Alta | |

Tabla 5-42 Requerimiento funcional 40

| | |
|--|--|
| Identificación del requerimiento: | RF_41 |
| Nombre del requerimiento: | Suscripción a tópicos. |
| Características: | El sistema debe permitir suscribirse a tópicos |
| Descripción del requerimiento: | <p>La suscripción a tópicos consta en vincular un usuario a un tema en específico. Los usuarios suscritos a un tópico, estarán en constante escucha de los mensajes que se envíen en estos (escucha las respuestas que vienen en formato JSON desde el módulo doméstico).</p> <p>Requerimientos funcionales asociados:</p> <ul style="list-style-type: none"> • RF_45 |

| | |
|-------------------------------------|---|
| | <ul style="list-style-type: none"> • RF_49 • RF_52 • RF_56 • RF_59 • RF_60 |
| Requerimiento no funcional: | <ul style="list-style-type: none"> • RNF_01 • RNF_02 • RNF_08 • RNF_11 |
| Interfaz externa asociada: | No aplica |
| Prioridad del requerimiento: | |
| Alta | |

Tabla 5-43 Requerimiento funcional 41

| | |
|--|--|
| Identificación del requerimiento: | RF_42 |
| Nombre del requerimiento: | Publicación en tópicos. |
| Características: | El sistema permitirá a los usuarios a publicar en los tópicos. |
| Descripción del requerimiento: | <p>La publicación de mensajes en los tópicos no requiere de estar suscrito en estos (peticiones en formato JSON, para el control del módulo doméstico).</p> <p>Requerimientos funcionales asociados:</p> <ul style="list-style-type: none"> • RF_43 • RF_44 • RF_46 • RF_47 • RF_48 |

| | |
|---|--|
| | <ul style="list-style-type: none"> • RF_51 |
| Requerimiento no funcional: | <ul style="list-style-type: none"> • RNF_01 • RNF_02 • RNF_08 • RNF_11 |
| Interfaz externa asociada: | No aplica |
| Prioridad del requerimiento: Alta | |

Tabla 5-44 Requerimiento funcional 42

| | |
|--|--|
| Identificación del requerimiento: | RF_43 |
| Nombre del requerimiento: | Sincronizar con módulo. |
| Características: | El sistema debe permitir sincronizar el módulo domótico con la aplicación móvil. |
| Descripción del requerimiento: | La sincronización consta en solicitar el estado actual al módulo domótico. |
| Requerimiento no funcional: | <ul style="list-style-type: none"> • RNF_01 • RNF_02 • RNF_03 • RNF_04 • RNF_05 • RNF_06 • RNF_07 • RNF_08 • RNF_11 |
| Interfaz externa asociada: | No aplica |
| Prioridad del requerimiento: | |

| |
|------|
| Alta |
|------|

Tabla 5-45 Requerimiento funcional 43

| | |
|--|--|
| Identificación del requerimiento: | RF_44 |
| Nombre del requerimiento: | Controlar módulo. |
| Características: | El sistema debe permitir el control de los elementos del módulo domótico. |
| Descripción del requerimiento: | El sistema debe permitir el control de los ocho relés del módulo domótico, de forma independiente. |
| Requerimiento no funcional: | <ul style="list-style-type: none"> • RNF_01 • RNF_02 • RNF_03 • RNF_04 • RNF_05 • RNF_06 • RNF_07 • RNF_08 • RNF_11 |
| Interfaz externa asociada: | No aplica |
| Prioridad del requerimiento: | Alta |

Tabla 5-46 Requerimiento funcional 44

| | |
|--|---|
| Identificación del requerimiento: | RF_45 |
| Nombre del requerimiento: | Obtener temperatura y humedad. |
| Características: | El sistema debe mostrar la temperatura y humedad. |

| | |
|---|--|
| Descripción del requerimiento: | El sistema permitirá mostrar la temperatura y humedad obtenida del módulo domótico. |
| Requerimiento no funcional: | <ul style="list-style-type: none"> • RNF_01 • RNF_02 • RNF_03 • RNF_08 • RNF_11 |
| Interfaz externa asociada: | No aplica |
| Prioridad del requerimiento: Alta | |

Tabla 5-47 Requerimiento funcional 45

| | |
|--|--|
| Identificación del requerimiento: | RF_46 |
| Nombre del requerimiento: | Automatizar relés. |
| Características: | El sistema debe permitir automatizar los elementos del módulo domótico. |
| Descripción del requerimiento: | El sistema debe permitir automatizar el encendido y apagado de los relés, de forma independiente. |
| Requerimiento no funcional: | <ul style="list-style-type: none"> • RNF_01 • RNF_02 • RNF_03 • RNF_04 • RNF_05 • RNF_06 • RNF_07 • RNF_08 |

| | |
|---|--|
| | <ul style="list-style-type: none"> • RNF_11 |
| Interfaz externa asociada: | No aplica |
| Prioridad del requerimiento: Alta | |

Tabla 5-48 Requerimiento funcional 46

| | |
|---|--|
| Identificación del requerimiento: | RF_47 |
| Nombre del requerimiento: | Desactivar horarios automáticos relés. |
| Características: | El sistema debe permitir borrar los horarios automáticos de los relés. |
| Descripción del requerimiento: | El sistema permitirá eliminar los encendidos automáticos de cada relé del módulo domótico. |
| Requerimiento no funcional: | <ul style="list-style-type: none"> • RNF_01 • RNF_02 • RNF_03 • RNF_04 • RNF_05 • RNF_06 • RNF_07 • RNF_08 • RNF_11 |
| Interfaz externa asociada: | No aplica |
| Prioridad del requerimiento: Alta | |

Tabla 5-49 Requerimiento funcional 47

| | |
|--|--|
| Identificación del requerimiento: | RF_48 |
| Nombre del requerimiento: | Establecer zona horaria. |
| Características: | El sistema permitirá establecer una zona horaria. |
| Descripción del requerimiento: | El sistema permitirá mandar un valor GTM al módulo domótico par que se sincronice con un servidor NTP |
| Requerimiento no funcional: | <ul style="list-style-type: none"> • RNF_01 • RNF_02 • RNF_03 • RNF_04 • RNF_05 • RNF_06 • RNF_07 • RNF_08 • RNF_11 |
| Interfaz externa asociada: | No aplica |
| Prioridad del requerimiento: | Alta |

Tabla 5-50 Requerimiento funcional 48

| | |
|--|--|
| Identificación del requerimiento: | RF_49 |
| Nombre del requerimiento: | Ver zona horaria. |
| Características: | El sistema permitirá visualizar la zona horaria. |
| Descripción del requerimiento: | El módulo domótico enviará la zona horaria en el primer encendido y está quedará registrada en el sistema. |

| | |
|---|--|
| <p>Requerimiento no funcional:</p> | <ul style="list-style-type: none"> • RNF_01 • RNF_02 • RNF_03 • RNF_04 • RNF_05 • RNF_06 • RNF_07 • RNF_08 • RNF_11 |
| <p>Interfaz externa asociada:</p> | <p>No aplica</p> |
| <p>Prioridad del requerimiento: Alta</p> | |

Tabla 5-51 Requerimiento funcional 49

5.4.1.3. Requerimientos funcionales del módulo domótico

| | |
|--|--|
| Identificación del requerimiento: | RF_50 |
| Nombre del requerimiento: | Inicio de sesión. |
| Características: | El módulo iniciará sesión en el sistema con la misma cuenta de la aplicación Android. |
| Descripción del requerimiento: | El sistema debe permitir enviar los datos de usuario y contraseña para iniciar sesión en el servidor. |
| Requerimiento no funcional: | <ul style="list-style-type: none"> • RNF_01 • RNF_02 • RNF_03 • RNF_04 • RNF_08 |
| Interfaz externa asociada: | No aplica |
| Prioridad del requerimiento: | Alta |

Tabla 5-52 Requerimiento funcional 50

| | |
|--|--|
| Identificación del requerimiento: | RF_51 |
| Nombre del requerimiento: | Suscripción a tópicos. |
| Características: | El módulo escuchará publicados en el servidor por usuarios de la aplicación móvil. |
| Descripción del requerimiento: | El módulo domótico debe suscribirse a tópicos, y escuchará constantemente los mensajes mandados a estos. Los mensajes corresponden a las peticiones provenientes desde la aplicación móvil con el fin de |

| | |
|-------------------------------------|---|
| | <p>esperar una acción o respuesta del módulo domótico (mensajes en formato JSON).</p> <p>Requerimientos funcionales asociados:</p> <ul style="list-style-type: none"> • RF_42 • RF_43 • RF_44 • RF_46 • RF_47 • RF_48 |
| Requerimiento no funcional: | <ul style="list-style-type: none"> • RNF_01 • RNF_02 • RNF_03 • RNF_04 • RNF_08 |
| Interfaz externa asociada: | No aplica |
| Prioridad del requerimiento: | |
| Alta | |

Tabla 5-53 Requerimiento funcional 51

| | |
|--|--|
| Identificación del requerimiento: | RF_52 |
| Nombre del requerimiento: | Publicación a tópicos. |
| Características: | El módulo domótico podrá publicar mensajes en el servidor. |
| Descripción del requerimiento: | <p>El módulo domótico debe publicar mensajes de respuesta a solicitudes (mensajes en formato JSON).</p> <p>Requerimientos funcionales asociados:</p> <ul style="list-style-type: none"> • RF_41 |

| | |
|-------------------------------------|--|
| | <ul style="list-style-type: none"> • RF_45 • RF_49 • RF_56 • RF_59 • RF_60 |
| Requerimiento no funcional: | <ul style="list-style-type: none"> • RNF_01 • RNF_02 • RNF_03 • RNF_04 • RNF_08 |
| Interfaz externa asociada: | No aplica |
| Prioridad del requerimiento: | |
| Alta | |

Tabla 5-54 Requerimiento funcional 52

| | |
|--|--|
| Identificación del requerimiento: | RF_53 |
| Nombre del requerimiento: | Control remoto. |
| Características: | El módulo domótico aceptará el control remoto de usuarios vinculados a este. |
| Descripción del requerimiento: | El módulo debe ser capaz de controlar los ocho relés de forma independiente y guardar el estado de estos (encendido y apagado), para luego poder publicarlos en un determinado tópico (para sincronización). |
| Requerimiento no funcional: | <ul style="list-style-type: none"> • RNF_01 • RNF_02 • RNF_03 |

| | |
|---|--|
| | <ul style="list-style-type: none"> • RNF_04 • RNF_08 |
| Interfaz externa asociada: | No aplica |
| Prioridad del requerimiento: Alta | |

Tabla 5-55 Requerimiento funcional 53

| | |
|---|--|
| Identificación del requerimiento: | RF_54 |
| Nombre del requerimiento: | Automatizar relés. |
| Características: | El módulo domótico aceptará la automatización de encendido y apagado de los relés. |
| Descripción del requerimiento: | El módulo debe ser capaz de controlar los ocho relés de forma automática, según los horarios establecidos por un usuario. |
| Requerimiento no funcional: | <ul style="list-style-type: none"> • RNF_01 • RNF_02 • RNF_03 • RNF_04 • RNF_08 |
| Interfaz externa asociada: | No aplica |
| Prioridad del requerimiento: Alta | |

Tabla 5-56 Requerimiento funcional 54

| | |
|--|--|
| Identificación del requerimiento: | RF_55 |
| Nombre del requerimiento: | Desactivar horarios automáticos relés. |

| | |
|---|--|
| Características: | El módulo domótico aceptará la eliminación de los horarios de los relés. |
| Descripción del requerimiento: | El módulo debe ser capaz de eliminar los horarios de encendido y apagado de los ocho relés de forma independiente. |
| Requerimiento no funcional: | <ul style="list-style-type: none"> • RNF_01 • RNF_02 • RNF_03 • RNF_04 • RNF_08 |
| Interfaz externa asociada: | No aplica |
| Prioridad del requerimiento: Alta | |

Tabla 5-57 Requerimiento funcional 55

| | |
|--|---|
| Identificación del requerimiento: | RF_56 |
| Nombre del requerimiento: | Responder variables de ambiente. |
| Características: | El módulo debe ser capaz de medir la temperatura y humedad del ambiente |
| Descripción del requerimiento: | El módulo domótico debe poder obtener las variables de ambiente (temperatura y humedad), con el fin de publicarlas en un determinado tópico. También se soportará la sincronización de estos datos. |
| Requerimiento no funcional: | <ul style="list-style-type: none"> • RNF_01 • RNF_02 • RNF_03 • RNF_04 |

| | |
|---|--|
| | <ul style="list-style-type: none"> • RNF_08 |
| Interfaz externa asociada: | No aplica |
| Prioridad del requerimiento: Alta | |

Tabla 5-58 Requerimiento funcional 56

| | |
|---|---|
| Identificación del requerimiento: | RF_57 |
| Nombre del requerimiento: | Sincronizar hora. |
| Características: | El módulo debe ser capaz de sincronizar la hora, con una determinada zona horaria. |
| Descripción del requerimiento: | Durante el primer inicio el servidor se conectará a un servidor NTP, y establecerá una zona horaria por defecto. El valor GTM será almacenado (para la sincronización). |
| Requerimiento no funcional: | <ul style="list-style-type: none"> • RNF_01 • RNF_02 • RNF_03 • RNF_04 • RNF_08 |
| Interfaz externa asociada: | No aplica |
| Prioridad del requerimiento: Alta | |

Tabla 5-59 Requerimiento funcional 57

| | |
|--|--|
| Identificación del requerimiento: | RF_58 |
| Nombre del requerimiento: | Establecer zona horaria. |
| Características: | El módulo podrá cambiar de zona horaria. |

| | |
|---|--|
| Descripción del requerimiento: | El módulo soporta el cambio de zona horaria, con el fin de adaptarse a la zona geográfica en donde este se encuentre. |
| Requerimiento no funcional: | <ul style="list-style-type: none"> • RNF_01 • RNF_02 • RNF_03 • RNF_04 • RNF_08 |
| Interfaz externa asociada: | No aplica |
| Prioridad del requerimiento: Alta | |

Tabla 5-60 Requerimiento funcional 58

| | |
|---|---|
| Identificación del requerimiento: | RF_59 |
| Nombre del requerimiento: | Responder zona horaria. |
| Características: | El módulo podrá responder el valor GTM establecido. |
| Descripción del requerimiento: | El módulo domótico cada cierto periodo de tiempo, dará a conocer su zona horaria. |
| Requerimiento no funcional: | |
| Interfaz externa asociada: | No aplica |
| Prioridad del requerimiento: Alta | |

Tabla 5-61 Requerimiento funcional 59

| | |
|--|-----------------------------|
| Identificación del requerimiento: | RF_60 |
| Nombre del requerimiento: | Responder a sincronización. |

| | |
|---|--|
| Características: | El módulo debe ser capaz de responder: El estado de los ocho relés, las horas establecidas para el encendido o apagado automático de los relés, el valor GTM, la temperatura y humedad. |
| Descripción del requerimiento: | El módulo tendrá la capacidad de responder a una petición de sincronización, con el fin de dar a conocer el estado actual de sus componentes. |
| Requerimiento no funcional: | <ul style="list-style-type: none"> • RNF_01 • RNF_02 • RNF_03 • RNF_04 • RNF_08 |
| Interfaz externa asociada: | No aplica |
| Prioridad del requerimiento: Alta | |

Tabla 5-62 Requerimiento funcional 60

5.4.1. Interfaces externas de entrada

| Identificador | Nombre del ítem | Detalles de datos contenidos en el ítem |
|----------------------|------------------------|--|
| INT_ENT_01 | Usuarios. | Id, UserName, NormalizedUserName, Email, NormalizedEmail, EmailConfirmed, PasswordHash, SecurityStamp, CurrencyStamp, PhoneNumber, PhoneNumberConfirmed, TwoFactorEnabled, LockoutEnd, LockoutEnabled, AccessFailedCount, RoleId |

| | | |
|------------|---------------------------------|---|
| INT_ENT_02 | Módulos. | ModuleId, ModuleName, ModuleVersion, ModuleFirmware |
| INT_ENT_03 | Tópicos. | TopicId, TopicName, TopicTopic, DataTypeId |
| INT_ENT_04 | Tipos de datos | DataTypeId, DataTypeName, DataTypeType |
| INT_ENT_05 | Asignación de módulo a usuario. | UserId, ModuleId |
| INT_ENT_06 | Asignación de tópico a módulo. | ModuleId, TopicId |

Tabla 5-63 Interfaces externas de entrada

5.4.2. Interfaces externas de salida

| Identificador | Nombre del ítem | Detalles de datos contenidos en el ítem |
|---------------|--|--|
| INT_SAL_01 | Listar usuarios. | Id, UserName, NormalizedUserName, Email, NormalizedEmail, EmailConfirmed, PasswordHash, SecurityStamp, CurrencyStamp, PhoneNumber, PhoneNumberConfirmed, TwoFactorEnabled, LockoutEnd, LockoutEnabled, AccessFailedCount, RoleId |
| INT_SAL_02 | Listar módulos. | ModuleId, ModuleName, Module Versión, ModuleFirmware |
| INT_SAL_03 | Listar tópicos. | TopicId, TopicName, TopicTopic, DataTypeId |
| INT_SAL_04 | Listar tipos de datos | DataTypeId, DataTypeName, DataTypeType |
| INT_SAL_05 | Listar asignación de módulo a usuario. | INT_SAL_02 |
| INT_SAL_06 | Listar asignación de tópico a módulo. | INT_SAL_03 |

| | | |
|------------|----------------------|--|
| INT_SAL_07 | Buscar usuario. | Id, UserName, NormalizedUserName, Email, NormalizedEmail, EmailConfirmed, PasswordHash, SecurityStamp, CurrencyStamp, PhoneNumber, PhoneNumberConfirmed, TwoFactorEnabled, LockoutEnd, LockoutEnabled, AccessFailedCount |
| INT_SAL_08 | Buscar módulo. | ModuleId, ModuleName, Module Versión, ModuleFirmware |
| INT_SAL_09 | Buscar tipo de dato. | DataTypeId, DataTypeName, DataTypeType |
| INT_SAL_10 | Buscar tópico. | TopicId, TopicName, TopicTopic, DataTypeId |

Tabla 5-64 Interfaces externas de salida

5.4.3. Atributos de calidad del sistema

5.4.3.1. Requerimientos no funcionales del producto

5.4.3.1.1. Requerimientos de eficiencia

| Identificador | Nombre del requerimiento | Descripción del requerimiento |
|---------------|---|--|
| RNF_01 | Desempeño del sistema. | Garantizar que el diseño de las consultas y otros procesos del sistema, no afecten el desempeño de la base de datos, ni saturen el tráfico de red. |
| RNF_02 | Velocidad de respuesta del sistema. | Todas las funcionalidades y transacciones del sistema deben responder en un tiempo inferior a 6 segundos. |
| RNF_03 | Confianza en la respuesta de los módulos. | El sistema debe garantizar a los usuarios un buen desempeño en cuanto a la respuesta de los módulos controlados mediante la aplicación móvil y a los datos consultados por el administrador. |

Tabla 5-65 Atributos de calidad de eficiencia

5.4.3.1.2. Requerimientos de usabilidad

| Identificador | Nombre del requerimiento | Descripción del requerimiento |
|---------------|---------------------------------------|---|
| RNF_04 | Acceso indirecto al sistema. | La plataforma no puede ser accedida directamente, sino a través de una interfaz diseñada para estos propósitos (Aplicación móvil o cliente http). |
| RNF_05 | Aprendizaje sobre el uso del sistema. | El tiempo de aprendizaje del sistema por un usuario deberá ser menor a 5 minutos |
| RNF_06 | Ayuda para los usuarios. | La interfaz del usuario en la aplicación móvil deberá presentar un sistema de |

| | | |
|--------|------------------------------|--|
| | | ayuda, para proveer información del uso de la aplicación a usuarios que presenten dudas (el sistema debe contar con un módulo de ayuda). |
| RNF_07 | Consistencia de la interfaz. | El sistema debe tener una interfaz de uso bien formada, intuitiva y sencilla y que encapsule las todas las funcionalidades para el control del módulo domótico en una sola ventana (aplicación Android). |

Tabla 5-66 Atributos de calidad de usabilidad

5.4.3.1.3. Requerimientos de disponibilidad

| Identificador | Nombre del requerimiento | Descripción del requerimiento |
|---------------|---|---|
| RNF_08 | Nivel de servicio del sistema. | La disponibilidad del sistema debe ser continua, otorgando un nivel de servicio para los usuarios de 24 horas, los siete días de la semana. |
| RNF_09 | Promedio de duración a fallas del sistema. | El promedio del tiempo de duración de las fallas en el sistema, no debe ser mayor a 10 minutos. |
| RNF_10 | Tiempo de respuesta del reinicio del sistema. | El tiempo para iniciar o reiniciar el sistema no podrá ser mayor a 3 minutos. |

Tabla 5-67 Atributos de calidad de disponibilidad

5.4.3.1.4. Seguridad

| Identificador | Nombre del requerimiento | Descripción del requerimiento |
|---------------|-------------------------------|---|
| RNF_11 | Control de acceso al sistema. | El sistema debe asegurar que los datos estén protegidos del acceso no autorizado. |

| | | |
|--------|---|--|
| RNF_12 | Control de permisos. | Los permisos de acceso al sistema podrán ser cambiados solamente por los administradores del sistema. |
| RNF_13 | Patrones y técnicas para el desarrollo. | El sistema debe desarrollarse utilizando patrones de diseño y técnicas de programación que otorguen seguridad e integridad a los datos. |
| RNF_14 | Reacción contra ataques. | Si se identifican ataques de seguridad o brecha del sistema, el mismo no continuará operando hasta ser desbloqueado por un administrador de seguridad. |

Tabla 5-68 Atributos de calidad de seguridad

5.4.3.1.5. Requerimientos de mantenibilidad

| Identificador | Nombre del requerimiento | Descripción del requerimiento |
|---------------|--|---|
| RNF_15 | Documentación del sistema. | El sistema debe disponer de una documentación fácilmente actualizable, y que permita realizar operaciones de mantenimiento con el menor esfuerzo posible. |
| RNF_16 | Sistema de ayuda. | La interfaz debe estar complementada con un buen sistema de ayuda para usuarios con poca experiencia en el uso de aplicaciones móviles. |
| RNF_17 | Mantenimiento y actualización de hardware. | El dispositivo domótico debe tener una estructura que admita la integración de nuevo Hardware (sensores o actuadores). |

| | | |
|--------|-----------------------------------|---|
| RNF_18 | Documentación del código escrito. | Los programas escritos deben estar correctamente comentados, para el entendimiento futuro de estos, por parte de terceras personas. |
|--------|-----------------------------------|---|

Tabla 5-69 Atributos de calidad de mantenibilidad

5.4.3.1.6. Requerimientos de portabilidad

| Identificador | Nombre del requerimiento | Descripción del requerimiento |
|---------------|-----------------------------|---|
| RNF_19 | Compatibilidad del sistema. | El ecosistema domótico no dependerá de un sistema operativo en específico (web API y servidor MQTT), este podrá correr en Linux, Windows y MacOS. Los sistemas operativos deben poseer como único requisito el .NET Core SDK. |

Tabla 5-70 Atributos de calidad de portabilidad

5.4.3.2. Requerimientos no funcionales Organizacionales

| Identificador | Nombre del requerimiento | Descripción del requerimiento |
|---------------|----------------------------|---|
| RNF_20 | Toma de requerimientos. | Para la obtención de requerimientos se utilizará una adaptación basada en IEEE Software requirements Specifications Std 830-1998. |
| RNF_21 | Metodología de desarrollo. | La metodología de desarrollo de software será Kanban |
| RNF_22 | Gestión del proyecto. | El proceso de desarrollo se gestionará por medio de la herramienta web Trello, para gestionar el proceso de desarrollo de software. |

| | | |
|--------|------------------|---|
| RNF_22 | Plan de pruebas. | Para las pruebas se utilizará una adaptación basada en IEEE Software Test Documentation Std 829-1998. |
|--------|------------------|---|

Tabla 5-71 Atributos de calidad organizacionales

CAPÍTULO 6: FACTIBILIDAD

6.1. Factibilidad técnica

En el presente estudio se centra en la realización del análisis del equipamiento computacional en posesión, tanto el software como el hardware empleado en el desarrollo del proyecto, determinando si estos cumplen los requerimientos para su utilización o en caso contrario complementarlo con nuevos recursos de hardware o software.

6.1.1. Hardware

Para el desarrollo del ecosistema domótico/IoT se cuenta con un ordenador portátil con acceso a Internet, el cual posee las siguientes especificaciones:

- **Marca:** Lenovo
- **Procesador:** Intel(R) Core (TM) i5-7300HQ CPU @ 2.50GHz
- **RAM:** 16 GB
- **Disco duro:** 1TB
- **Conectividad:** WIFI 2X2 A
- **Periféricos E/S:** Ratón y teclado

De forma paralela, el Hardware necesario para el desarrollo del módulo domótico es el siguiente (hardware en posesión):

- NodeMCU ESP32
- Módulo RTC DS1302
- Módulo de 8 relés
- Módulo DTH22
- Fuente de alimentación YwRobot

Además, se hace necesario el arriendo de un **VPS** para instalar el software requerido y posteriormente pasar el sistema a producción (Web API y Servidor MQTT). Las especificaciones se definen a continuación:

- **Disco duro:** 1TB
- **Procesador:** XEON E7

- **Trafico:** 1TB
- **RAM:** 1GB

6.1.2. Software

El software que se dispone para abordar el proyecto, es el siguiente (todos de uso libre):

6.1.2.1. Computador personal (Desarrollo)

- Visual Estudio 2017 Community, **ver.** 15.9.5
- Visual Estudio Code, **ver.** 1.30.2
- Android Estudio, **ver.** 3.3
- Gantt Proyect, **ver.** 2.8.9
- StarUML, **ver.** 3.0.2
- DataGrip libre para estudiantes, **ver.** 2018.3.1
- Trello Desktop, **ver.** Microsoft Store (2019)
- Arduino IDE for Visual Estudio, **ver.** 1.0
- .NET Core, **ver.** 2.2

6.1.2.2. VPS(Producción)

- Windows (10) o Linux (CentOS 7)
- .NET Core, **ver.** 2.2
- Kestrel **ver.** 2.2.0
- Apache **ver.** 2.4

6.1.3. Conclusión

Habiendo expresado lo anterior, se puede dilucidar que llevar a cabo este proyecto, desde el punto de vista técnico, es factible, pues se posee el conocimiento de uso sobre el hardware y software ya mencionado. Además, cabe recalcar, que todo el software utilizado es de uso libre, y en cuanto al hardware y el arriendo del VPS, ya se tiene a disposición.

6.2. Factibilidad operativa

El sistema domótico modular con automatización y control de dispositivos eléctricos, basado en NodeMCU-32s (ecosistema domótico) ofrece una solución, para la automatización y control del encendido o apagado de los electrodomésticos de un hogar, específicamente de los usuarios poseedores del módulo, que se hayan registrado en la aplicación móvil.

En lo concerniente al impacto del sistema sobre los usuarios finales, se puede mencionar que, para la realización de este proyecto, se ha tomado en cuenta las necesidades latentes de la sociedad contemporánea, debido a que las tecnologías actuales permiten ofrecer servicios o productos que satisfagan necesidades, tales como:

- **Ahorro energético:** El ahorro energético es un concepto legible que consta en reducir el consumo de energía de los aparatos eléctricos de una vivienda, con el fin de escatimar en gastos.
- **Confort:** El confort corresponde a todas las actuaciones que se puedan llevar a cabo, para mejorar la comodidad en una vivienda.
- **Seguridad:** consiste en despreocupar a los usuarios sobre el estado (encendido o apagado) de sus equipos conectados al módulo, pues este mostrará si el relé este encendido o apagado.
- **Comunicaciones:** Las comunicaciones se refiere a los sistemas o infraestructuras de comunicaciones que posee un hogar. En el ámbito de este proyecto, se refiere al a la interacción remota con el módulo domótico, haciendo uso de una red WIFI.
- **Accesibilidad:** La accesibilidad corresponde a diseñar un producto o servicio, que tenga en cuenta las necesidades de todos los posibles usuarios, incluyendo personas con capacidades diferente o discapacidades. En este caso particular nos referimos a personas con capacidades cognitivas sobre el uso de una aplicación móvil, y el efecto que esta produce en el entorno y que posean facultades para interactuar con la aplicación mediante el uso de las manos o cualquier método que dispongan, para dicha acción

También se ha tomado en consideración que las personas se familiarizan más rápidamente con los gestos táctiles que con los realizados a través de periféricos como teclados o ratones. Por este mismo motivo, no es de extrañar que todos los consumidores móviles (incluido niños y personas de tercera edad) aprendan con facilidad a manejar los diferentes dispositivos. Por este motivo, la aplicación móvil en cuestión ha tomado en cuenta el término de usabilidad, pues una aplicación móvil debe ser efectiva y eficiente. Esto significa que debe ofrecer al usuario lo que busca de manera rápida y concisa. Una aplicación móvil debe satisfacer las necesidades de los usuarios en el menor tiempo posible, si necesidad de que estos tengan grandes conocimientos de su uso o requiera un largo proceso de aprendizaje.

Las aplicaciones deben llegar a la efectividad y la eficiencia, mediante una buena experiencia de usuario. Sólo de esta manera se generarán en los usuarios una serie de emociones que le aporten confianza a la aplicación. Por tanto, la usabilidad es un mecanismo fundamental a la hora de fidelizar a los usuarios en el ámbito de las aplicaciones móviles. (Yeeply, 2013).

De este modo se puede garantizar que la implementación del sistema será exitosa, debido a que han tomado en cuenta las necesidades latentes de la actual sociedad. También se ha tenido en consideración que a nivel nacional no hay empresas dedicadas a la fabricación de módulos domóticos/ IoT y que las alternativas similares en el mercado (origen extranjero) tienen un alto costo.

6.3. Factibilidad económica

En este estudio se pretende demostrar la rentabilidad del proyecto como una única unidad, no se pretende demostrar la rentabilidad comercial. También se procederá a comparar el módulo creado con uno existente en el mercado con similares características.

Las variables a tener en consideración son: Costos de los recursos utilizados y costo de desarrollo (el proyecto es propio y el costo de desarrollo solo pretende cuantificar el esfuerzo). Los costos de puesta en marcha no serán considerados, pues el servidor es el mismo utilizado en las pruebas y todo el software utilizado es de uso libre.

| Nombre del recurso | | Valor (pesos chilenos) |
|--|-----------------------------------|------------------------|
| Servidor en la nube (VPS 1 año): | | \$27.000 |
| Hardware: | NodeMCU ESP32 | \$2.100,23 |
| | Módulo RTC DS1302 | \$1.073,6 |
| | Módulo de 8 relés | \$1.596,98 |
| | Módulo DTH22 | \$764,65 |
| | Fuente de alimentación YwRobot | \$671 |
| Carcasa para carril caja 145x90x40mm: | | \$3.502,62 |
| Placa para componentes: | | \$664,29 |
| Cables para conexiones: | | \$700 |
| Costo desarrollo (8000 por hora de desarrollo): | 240 horas | \$1.920.000 |
| Consumo energético durante el desarrollo (8 horas diarias durante 1 mes, el kWh día cuesta \$ 144): | 4500 kWh /mes | \$3.413 |
| Total: | | \$1.957.983,75 |

Tabla 6-1 Tabla de costos - Factibilidad económica

El costo por módulo se presentará en la siguiente tabla (tabla 6-2):

| Nombre identificador | Valor |
|---------------------------|------------|
| NodeMCU ESP32: | \$2.100,23 |
| Módulo RTC DS1302: | \$1.073,6 |
| Módulo de 8 relés: | \$1.596,98 |
| Módulo DTH22: | \$764,65 |

| | |
|--|-------------|
| Fuente de alimentación YwRobot: | \$671 |
| Carcasa para carril caja 145x90x40mm: | \$3.502,62 |
| 0,1 % del costo de desarrollo: | \$1.920 |
| Placa para componentes: | \$664,29 |
| Cables para conexiones: | \$700 |
| Total: | \$12.993,37 |

Tabla 6-2 Costo por módulo - Factibilidad económica

Como se puede observar en la **tabla 6-2**, el costo por realizar cada módulo es de aproximadamente \$11.620 y este puede variar dependiendo del valor del dólar. Se debe tener en consideración que el módulo permite controlar y automatizar 8 relés, ya que actualmente una de las alternativas existentes en el mercado (Sonoff 4CH de itead) ofrece el control de 4 relés por un valor de \$15.365 en páginas y en el comercio nacional se puede obtener por un valor de \$25.000.

6.4. Factibilidad Legal

Para la realización de este proyecto se han tomado en cuenta las siguientes normas y leyes vigentes en la legislación chilena:

- Ley N°18.410, el artículo 60 de la Ley 18.681 y el Decreto Supremo N°399/85 del Ministerio de Economía, Fomento y Reconstrucción:
 - Para la certificación de un producto eléctrico de uso masivo y de libre comercialización.
- NCh Elec. 4/2003, el artículo 8.0.4.15: Los conductores de una canalización eléctrica se identificarán según el siguiente código de colores:
 - Conductor de la fase 1: Azul.
 - Conductor de la fase 2: Negro.
 - Conductor de la fase 3: Rojo.
 - Conductor de neutro o tierra de servicio: Blanco.
 - Conductor de protección: Verde o Verde/Amarillo.

6.5. Conclusión de la factibilidad

Considerando el análisis técnico, operativo, económico y legal visto en los puntos 6.1, 6.2, 6.3 y 6.4 respectivamente se puede observar, que la fabricación de un producto alternativo de bajos costo es posible gracias a que se cuenta con los recursos de software y hardware necesarios para su desarrollo y confección. También se ha tomado en cuenta lo que dice la ley chilena respecto a la certificación de productos eléctricos.

Como se ha mencionado con anterioridad, este sistema no tiene una mirada comercial, solo se pretende presentar una alternativa de bajo costo, a las ya existentes en el mercado.

CAPÍTULO 7: ANÁLISIS

7.1. Diagrama de casos de uso

7.1.1. Actores

Administrador del sistema: El rol de administrador lo puede ocupar cualquier persona que cumpla con las especificaciones señaladas en el **punto 5.3.6**. Las capacidades del administrador en el sistema, se listarán a continuación:

- Crear, editar, eliminar y buscar usuarios.
- Crear, editar, eliminar y buscar módulos.
- Crear, editar, eliminar y buscar tópicos.
- Crear, editar, eliminar y buscar tipos de datos.
- Asignar módulos a usuarios.
- Asignar tópicos a módulos.
- Asignar tipo de dato a tópicos.

Usuario (aplicación móvil): Las características que debe poseer un usuario del sistema están especificadas en el **punto 5.3.6**. Las interacciones que tendrá el usuario mediante el uso de la aplicación móvil con el módulo domótico, son las siguientes:

- Registrar cuenta.
- Iniciar sesión.
- Publicar mensajes en tópicos.
- Suscribirse a tópicos.
- Controlar relés.
- Automatizar relés.
- Visualizar temperatura.
- Establecer zona horaria.
- Visualizar zona horaria.
- Sincronizar con el módulo domótico.

Módulo domótico: El módulo domótico podrá ser usado por cualquier persona que utilice la aplicación móvil, este necesita una cuenta de usuario para ser validado en el sistema. Las funcionalidades de este dispositivo son las siguientes:

- Iniciar sesión.
- Publicar mensajes en tópicos.
- Suscribirse a tópicos.
- Interactuar con los relés.
- Automatizar relés.
- responder temperatura y humedad.
- recibir zona horaria.
- responder zona horaria.
- Responder estado de los componentes del módulo domótico.

7.1.2. Diagramas de casos de Uso y descripción

En este apartado se presentarán los casos de usos correspondientes al ecosistema domótico, de forma segmentada, y según los roles de los usuarios del sistema (administrador, usuario de la aplicación móvil y módulo domótico).

En primer lugar, se presenta la **ilustración 7-1**, en esta se muestran los casos de uso correspondientes a los usuarios de la aplicación móvil. El diagrama representa las interacciones que puede tener estos, con el módulo domótico (se dará por entendido que el administrador del sistema también es un usuario de la aplicación móvil).

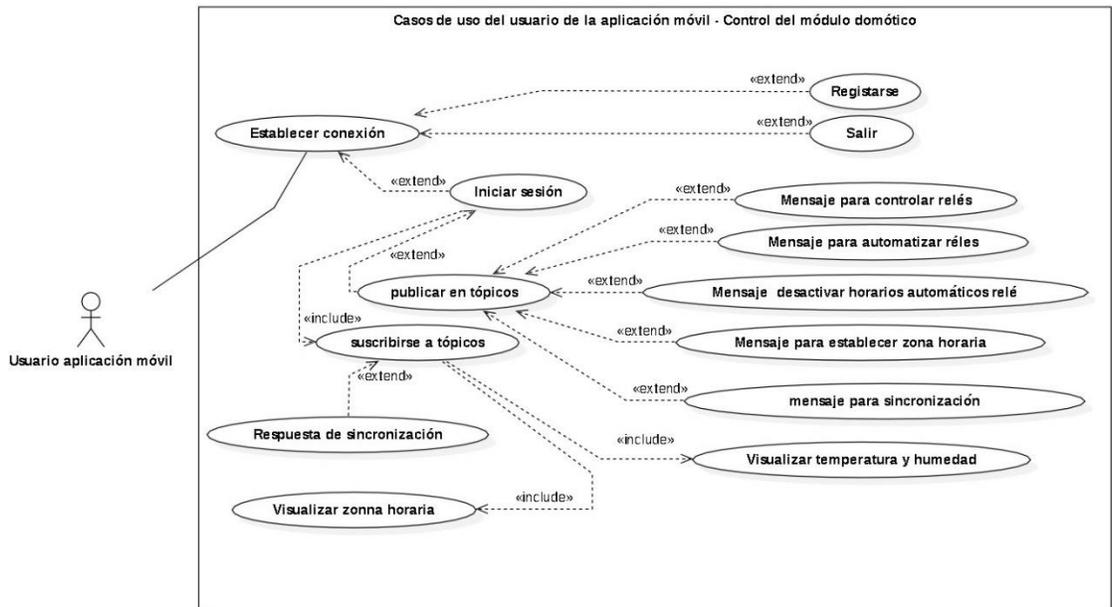


Ilustración 7-1 Casos de uso usuario - Control del módulo doméstico

A continuación, se presenta la **ilustración 7-2**, está da a conocer los casos de uso para usuarios de la aplicación móvil, vinculados al módulo doméstico. La imagen muestra el total de interacciones que podrá ejercer el usuario para controlar el módulo doméstico.

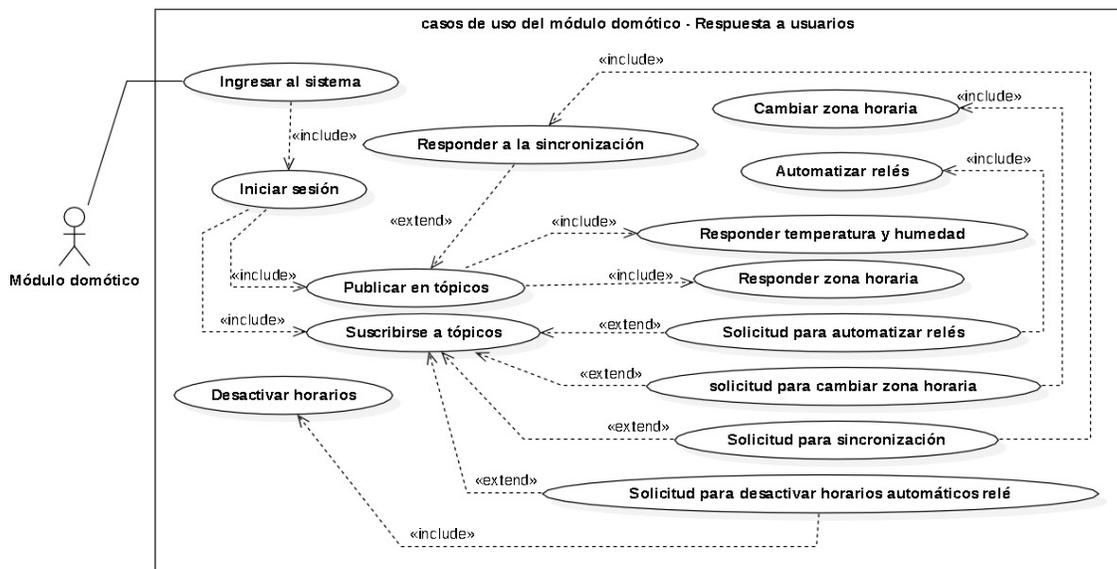


Ilustración 7-2 Casos de uso del módulo doméstico - Respuesta a usuarios

Por otra parte, en lo que concierne a los casos de uso del administrador, se ha optado por dividirlos en módulos (usuarios, dispositivos domóticos, tópicos, tipos de datos, asignaciones de módulos usuarios y asignaciones de tópicos a módulos).

En la **ilustración 7-3** se aprecian los casos de uso del módulo de usuarios, en los cuales se reflejan las interacciones que tendrá el administrador sobre este.

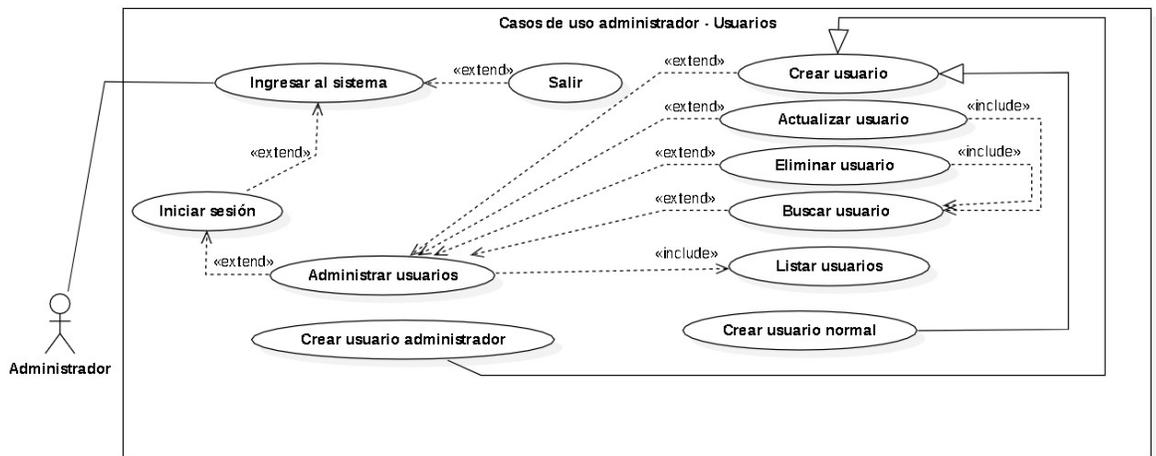


Ilustración 7-3 Casos de uso administrador - Módulo usuarios

La **ilustración 7-4** refleja las situaciones que se pueden dar para el administrador del sistema, sobre el módulo domótico.

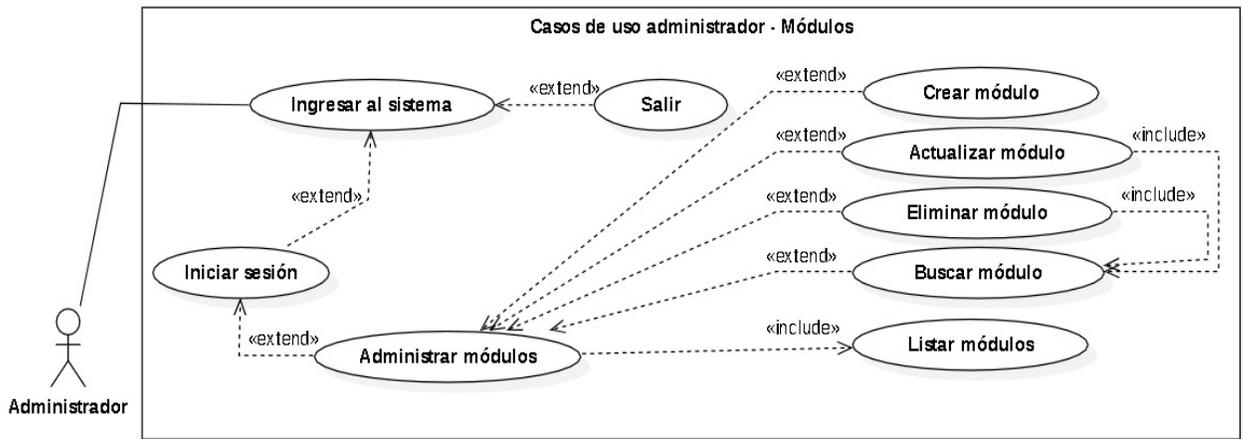


Ilustración 7-4 Casos de uso administrador - Módulo de dispositivos domésticos

En la **ilustración 7-4** se presentan los casos de uso para el administrador, referentes a las interacciones que puede tener este, con el módulo de tópicos.

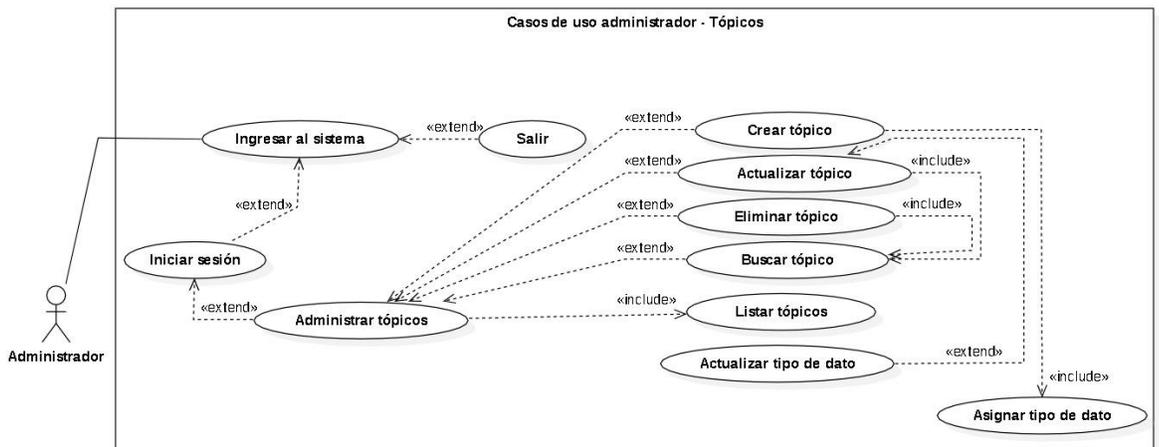


Ilustración 7-5 Casos de uso administrador - Módulo tópicos

Las situaciones que atañen al administrador con respecto al módulo de tipos de datos, se presentarán en la **ilustración 7-6**.

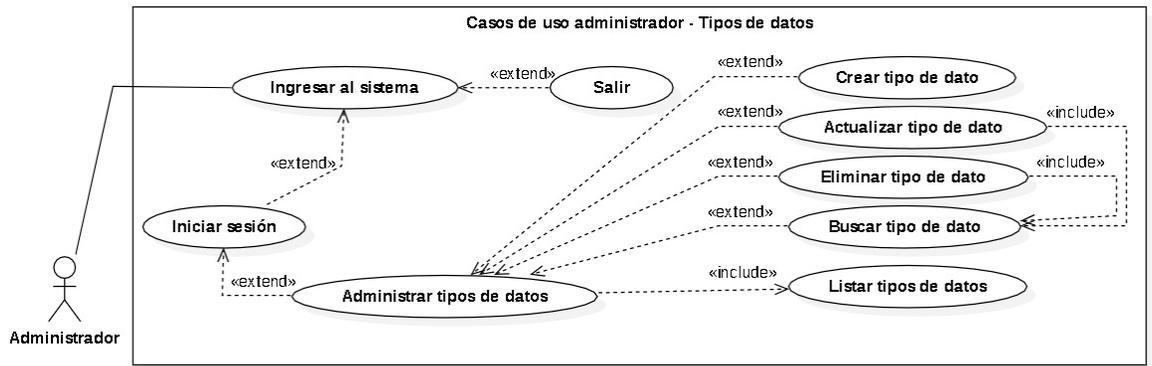


Ilustración 7-6 Casos de uso administrador - Módulo tipos de datos

En cuanto a las asignaciones de los módulos a los usuarios, la **ilustración 7-7** muestra el total de interacciones en las que está vinculado el administrador.

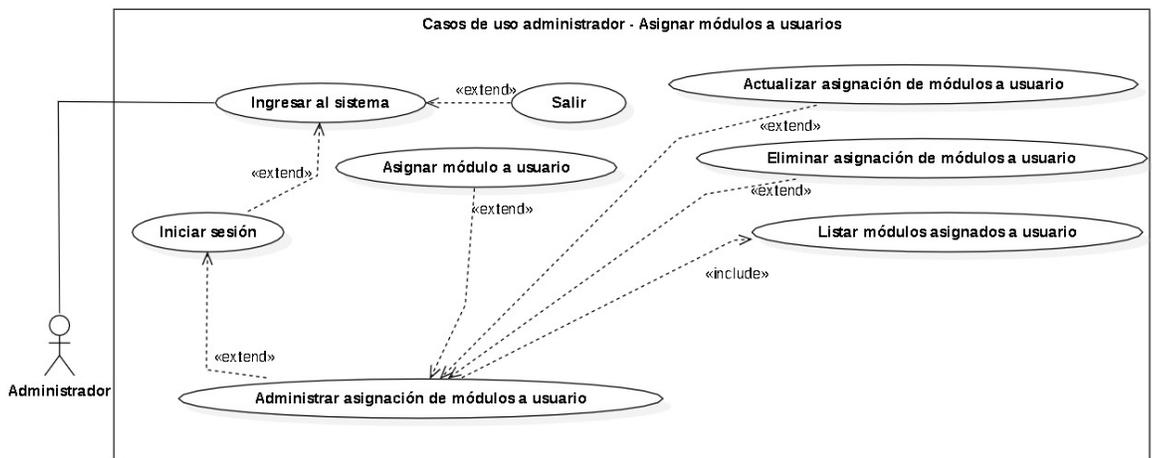


Ilustración 7-7 Casos de uso administrador - Asignar módulos a usuarios

Finalmente, para la asignación de tópicos a los módulos, el administrador podrá interactuar en los casos presentados en la **ilustración 7-8**.

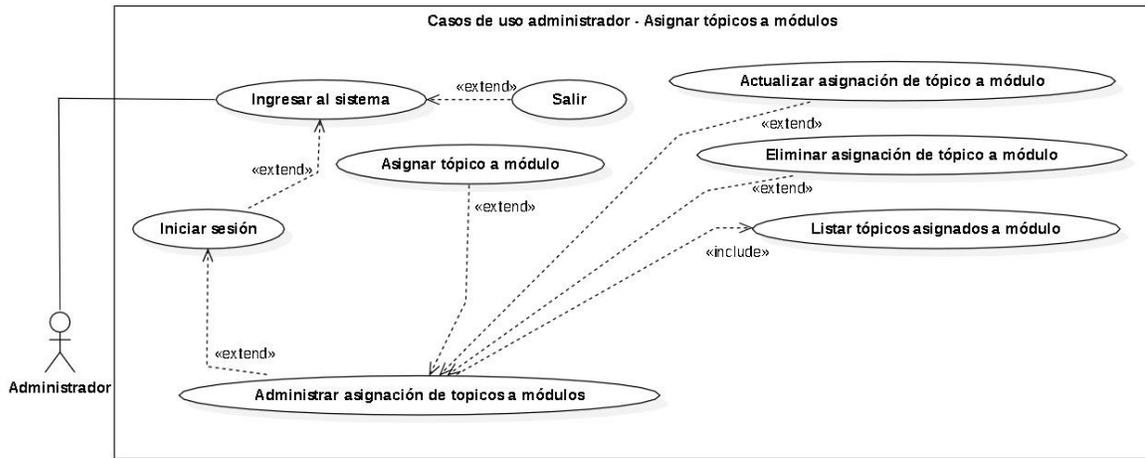


Ilustración 7-8 Casos de uso administrador - Asignar tópicos a módulos

7.1.3. Especificación de los casos de uso

7.1.3.1. Caso de uso <Establecer conexión>

- **Descripción:** Corresponde a una solicitud de conexión al servidor por parte de los usuarios, para interactuar con este.
- **Actores:** Administrador, usuario de la aplicación móvil y módulo domótico
- **Precondiciones:**
 1. Disponer de acceso a Internet
 2. Interactuar mediante un cliente HTTP o mediante la aplicación móvil (usuario normal o administrador). El módulo domótico debe estar encendido.
- **Flujo de eventos básicos:**

| Al actor | El sistema |
|---|---|
| 1. Los usuarios solicitan conectarse al servidor. | 2. Acepta la petición y responde que se está ejecutando satisfactoriamente. |
| 3. Los usuarios seleccionan una de las posibles opciones que les ofrece el sistema. | |

Tabla 7-1 Caso de uso establecer conexión - Flujo normal

- **Flujo de eventos alternativo:**

| Al actor | El sistema |
|---|---|
| 1. Los usuarios solicitan conectarse al servidor. | 2. Acepta la petición y responde que se está ejecutando satisfactoriamente. |
| 3. El tiempo de espera de la solicitud se agota. | |

Tabla 7-2 Caso de uso establecer conexión - Flujo alternativo

- **Postcondiciones:** Los usuarios tendrán conocimiento de que el servidor está en línea.

7.1.3.2. Caso de uso <Iniciar Sesión>

- **Descripción:** En este caso de uso, se validará la identidad del usuario en el sistema, comprobando su nombre de usuario y contraseña.
- **Actores:** Administrador y usuario de la aplicación móvil.
- **Precondiciones:**
 1. Los usuarios deben tener una cuenta registrada en el sistema.

- **Flujo de eventos básicos:**

| Al actor | El sistema |
|--|---|
| 1. Los actores ingresan sus datos correctamente (usuario y contraseña) y los mandan en una petición de inicio de sesión. | 2. Valida los datos ingresados por los usuarios, y responde un token de acceso único para el usuario. |

Tabla 7-3 Caso de uso iniciar sesión - Flujo normal (App móvil y administrador)

- **Flujo de eventos alternativo:**

| Al actor | El sistema |
|--|---|
| 1. Los actores ingresan sus datos correctamente (usuario y contraseña) y los mandan en una petición de inicio de sesión. | 2. La información ingresada por los usuarios no puede ser validada en el sistema, y este rechaza la petición. |

Tabla 7-4 Caso de uso iniciar sesión - Flujo alternativo (App móvil y administrador)

- **Postcondiciones:** Los actores no pueden interactuar con el sistema, pues no poseen un token de acceso.

7.1.3.3. Caso de uso <Iniciar Sesión>

- **Descripción:** El módulo domótico iniciará sesión de forma automática. Los datos para el login (usuario y contraseña) están registrados en el código.
- **Actores:** Módulo domótico.
- **Precondiciones:**
 1. El módulo debe tener una cuenta registrada en el sistema.
 2. El módulo tiene que estar encendido y conectado a Internet.
- **Flujo de eventos básicos:**

| Al actor | El sistema |
|---|--|
| 1. Envía el usuario y contraseña al servidor en una petición de inicio de sesión, esperando ser validado. | 2. Valida los datos y autoriza el ingreso del módulo domótico. |

Tabla 7-5 Caso de uso iniciar sesión - Flujo normal(módulo domótico)

- **Flujo de eventos alternativo:**

| Al actor | El sistema |
|---|--|
| 1. Envía el usuario y contraseña al servidor en una petición de inicio de sesión, esperando ser validado. | 2. No acepta la solicitud de inicio de sesión. |

Tabla 7-6 Caso de uso iniciar sesión - Flujo alternativo(módulo domótico)

- **Postcondiciones:** El dispositivo domótico no puede interactuar con el sistema, por tanto, no podrá ser controlado.

7.1.3.4. Caso de uso <Registrarse>

- **Descripción:** Permite a los usuarios crear una cuenta de acceso, para interactuar con el sistema.
- **Actores:** Usuario de la aplicación móvil.
- **Precondiciones:**
- **Flujo de eventos básicos:**

| Al actor | El sistema |
|--|---|
| 1. Abre la aplicación móvil y presiona el botón registrar. Luego, ingresa un nombre de usuario y contraseña presiona enviar. | 2. Recibe la solicitud, registra al usuario y responde un mensaje que contiene los datos de este. |

Tabla 7-7 Caso de uso registrarse - Flujo normal

- **Flujo de eventos alternativo:**

| Al actor | El sistema |
|---|--|
| 1. Abre la aplicación móvil y presiona el botón registrar. Luego, ingresa un nombre de usuario y contraseña presiona enviar | 2. Responde con un mensaje 400 Bad Request (el servidor no pudo interpretar la solicitud dada una sintaxis inválida). |

Tabla 7-8 Caso de uso registrarse - Flujo alternativo

- **Postcondiciones:** El usuario de la aplicación móvil registra su cuenta exitosamente.

7.1.3.5. Caso de uso <Salir>

- **Descripción:** Permite al usuario salir del sistema.
- **Actores:** Administrador y usuario de la aplicación móvil.
- **Precondiciones:**
 1. El usuario debe haber iniciado sesión en el sistema.
- **Flujo de eventos básicos:**

| Al actor | El sistema |
|--|---|
| 1. Cierra la aplicación (cliente HTTP o aplicación móvil). | 2. Se mantiene a la espera de conexiones. |

Tabla 7-9 Caso de uso salir - Flujo normal

- **Postcondiciones:** El usuario cierra la aplicación exitosamente y deja de interactuar con el sistema

7.1.3.6. Caso de uso <Publicar en tópicos>

- **Descripción:** El usuario de la aplicación móvil publica un mensaje en determinados tópicos del servidor.
- **Actores:** Usuario de la aplicación móvil.
- **Precondiciones:**
 1. El usuario debe haber iniciado sesión en el sistema

• **Flujo de eventos básicos:**

| Al actor | El sistema |
|---|---|
| 1. Publica mensajes satisfactoriamente en un determinado tópico del servidor. | 2. Recibe el mensaje y lo publica para todos los usuarios que se han suscrito a dicho tópico. |

Tabla 7-10 Caso de uso publicar en tópicos - Flujo normal

• **Flujo de eventos alternativo:**

| Al actor | El sistema |
|---|--|
| 1. Publica mensajes satisfactoriamente en un determinado tópico del servidor. | 2. Recibe el mensaje y lo publica, pero no hay otros usuarios suscritos al tópico. |

Tabla 7-11 Caso de uso publicar en tópicos - Flujo alternativo

- **Postcondiciones:** El usuario publicará un mensaje correctamente en el servidor.

7.1.3.7. Caso de uso <Mensaje para controlar relés>

- **Descripción:** El usuario manda un mensaje al servidor, que le permitirá controlar los ocho relés del módulo domótico.
- **Actores:** Usuario de la aplicación móvil.
- **Precondiciones:**
 1. El usuario debe poder publicar en tópicos.
 2. El módulo domótico debe estar en línea.
- **Flujo de eventos básicos:**

| Al actor | El sistema |
|---|---|
| 1. Presiona cualquiera de los diferentes switches de la aplicación móvil, destinados para el control de los relés. Luego, publica un mensaje formateado, en el que se registra una acción, para reconocer el mensaje; un id, para identificar a los relés y un estado de encendido o apagado, para cada relé. | 2. Recibe el mensaje y lo publica para todos los usuarios que se han suscrito a dicho tópico. |

Tabla 7-12 Caso de uso mensaje para controlar relés - Flujo normal

- **Flujo de eventos alternativo:**

| Al actor | El sistema |
|---|---|
| 1. Presiona cualquiera de los diferentes switches de la aplicación móvil, destinados para el control de los relés. Luego, publica un mensaje formateado, en el que se registra una acción, para reconocer el mensaje; un id, para identificar a los relés y un estado de encendido o apagado, para cada relé. | 2. Recibe el mensaje y lo publica para todos los usuarios que se han suscrito a dicho tópico. |

Tabla 7-13 Caso de uso mensaje para controlar relés - Flujo alternativo

- **Postcondiciones:** El usuario de la aplicación móvil publica un mensaje en el servidor, para controlar los relés del módulo domótico.

7.1.3.8. Caso de uso <Mensaje para automatizar relés>

- **Descripción:** El usuario manda un mensaje al servidor, que le permitirá automatizar los ocho relés del módulo domótico.
- **Actores:** Usuario de la aplicación móvil.
- **Precondiciones:**
 1. El usuario debe poder publicar en tópicos.
 2. El módulo domótico debe estar en línea.
- **Flujo de eventos básicos:**

| Al actor | El sistema |
|---|---|
| 1. El usuario hará usos de los timepickers de la aplicación móvil y enviará un mensaje formateado al servidor, en el que se registra una acción, para identificar el mensaje; un id, para identificar a los relés y una hora de encendido y una hora de apagado, para automatizar cada relé de forma independiente. | 2. Recibe el mensaje y lo publica para todos los usuarios que se han suscrito a dicho tópico. |

Tabla 7-14 Caso de uso mensaje para automatizar relés - Flujo normal

- **Flujo de eventos alternativo:**

| Al actor | El sistema |
|---|---|
| 1. El usuario hará usos de los timepickers de la aplicación móvil y enviará un mensaje formateado al servidor, en el que se registra una acción, para identificar el mensaje; un id, para identificar a los relés y una hora de encendido y una hora de apagado, para automatizar cada relé de forma independiente. | 2. Recibe el mensaje y lo publica para todos los usuarios que se han suscrito a dicho tópico. |

Tabla 7-15 Caso de uso mensaje para automatizar relés - Flujo alternativo

- **Postcondiciones:** El usuario de la aplicación móvil publica un mensaje en el servidor, que le permitirá automatizar los ocho relés del módulo domótico.

7.1.3.9. Caso de uso <Mensaje para establecer zona horaria>

- **Descripción:** El usuario Envía un mensaje al servidor, que le permitirá cambia la zona horaria del módulo domótico.
- **Actores:** Usuario de la aplicación móvil.
- **Precondiciones:**
 1. El usuario debe poder publicar en tópicos.
 2. El módulo domótico debe estar en línea.
- **Flujo de eventos básicos:**

| Al actor | El sistema |
|--|---|
| 1. Envía un mensaje formateado al servidor, en el que se registra una acción, para identificar el mensaje y una zona horaria, para que sea republicada por el servidor (lo hará mediante un EditText de la aplicación móvil, que está destinado para ese propósito). | 2. Recibe el mensaje y lo publica para todos los usuarios que se han suscrito a dicho tópico. |

Tabla 7-16 Caso de uso mensaje para establecer zona horaria - Flujo normal

- **Flujo de eventos alternativo:**

| Al actor | El sistema |
|--|---|
| 1. Envía un mensaje formateado al servidor, en el que se registra una acción, para identificar el mensaje y una zona horaria, para que sea republicada (lo hará mediante un EditText de la aplicación móvil, que está destinado para ese propósito). | 2. Recibe el mensaje, pero no hay otros usuarios suscritos al tópico. |

Tabla 7-17 Caso de uso mensaje para establecer zona horaria - Flujo alternativo

- **Postcondiciones:** El usuario de la aplicación móvil modifica la zona horaria del módulo domótico.

7.1.3.10. Caso de uso <Mensaje para la sincronización>

- **Descripción:** El usuario manda un mensaje al servidor, esperando que este le dé una respuesta que incluya el estado actual del módulo domótico.
- **Actores:** Usuario de la aplicación móvil.
- **Precondiciones:**
 1. El usuario debe poder publicar en tópicos.
 2. El módulo domótico debe estar en línea.
 3. El usuario debe estar suscrito en el tópico de la respuesta.
- **Flujo de eventos básicos:**

| Al actor | El sistema |
|--|---|
| 1. Envía un mensaje formateado al servidor, en el que se registra una acción, para identificar el mensaje (lo hará mediante el botón sincronizar que posee la aplicación móvil). | 2. Recibe el mensaje y lo reenvía al módulo domótico. Luego, espera la respuesta del módulo y la reenvía al usuario de la aplicación móvil. |

Tabla 7-18 Caso de uso mensaje para la sincronización - Flujo normal

- **Flujo de eventos alternativo:**

| Al actor | El sistema |
|--|---|
| 1. Envía un mensaje formateado al servidor, en el que se registra una acción, para identificar el mensaje (lo hará mediante el botón sincronizar que posee la aplicación móvil). | 2. Recibe el mensaje, pero el módulo domótico no está en línea. |

Tabla 7-19 Caso de uso mensaje para la sincronización - Flujo alternativo

- **Postcondiciones:** El usuario publica un mensaje satisfactoriamente en el servidor (Mensaje para establecer sincronización con el módulo domótico).

7.1.3.11. Caso de uso <Suscribirse a tópicos>

- **Descripción:** El usuario estará suscrito a los tópicos del servidor, de forma que pueda escuchar las publicaciones del módulo domótico.
- **Actores:** Usuario de la aplicación móvil.
- **Precondiciones:**
 1. El usuario debe haber iniciado sesión en el sistema.
- **Flujo de eventos básicos:**

| Al actor | El sistema |
|--|---|
| 1. Se suscribe a los tópicos del servidor. | 2. publica los mensajes enviados por el módulo domótico, para que lo oiga usuario de la aplicación móvil. |
| 3. Escucha los mensajes que el servidor publica en dichos tópicos. | |

Tabla 7-20 Caso de uso suscribirse a tópicos - Flujo normal

- **Flujo de eventos alternativo:**

| Al actor | El sistema |
|--|---|
| 1. Se suscribe a los tópicos del servidor. | 2. No recibe mensajes del módulo domótico, por tanto, no hace publicaciones |

Tabla 7-21 Caso de uso suscribirse a tópicos - Flujo alternativo

- **Postcondiciones:** El usuario podrá escuchar las publicaciones que haga el módulo domótico.

7.1.3.12. Caso de uso <Visualizar temperatura y humedad>

- **Descripción:** El usuario visualiza la temperatura y humedad publicada por el servidor (temperatura y humedad registrada por el módulo domótico).
- **Actores:** Usuario de la aplicación móvil.
- **Precondiciones:**
 1. El usuario debe estar suscrito a los tópicos establecidos en el servidor.
 2. El módulo domótico debe estar en línea.
- **Flujo de eventos básicos:**

| Al actor | El sistema |
|---|---|
| 2. Escucha el mensaje y lo visualiza desde un EditText. | 1. Periódicamente envía a los usuarios suscritos la temperatura y humedad publicada por el módulo domótico. |

Tabla 7-22 Caso de uso visualizar temperatura y humedad - Flujo normal

- **Flujo de eventos alternativo:**

| Al actor | El sistema |
|---|---|
| 2. Pierde la conexión, por tanto, visualiza la última temperatura y humedad arribada a la aplicación móvil. | 1. Periódicamente envía a los usuarios suscritos la temperatura y humedad publicada por el módulo domótico. |

Tabla 7-23 Caso de uso visualizar temperatura y humedad - Flujo alternativo

- **Postcondiciones:** El usuario será capaz de visualizar la última temperatura registrada por el módulo domótico.

7.1.3.13. Caso de uso <Visualizar zona horaria>

- **Descripción:** El usuario visualiza la zona horaria publicada por el servidor (zona horaria establecida en el módulo domótico).
- **Actores:** Usuario de la aplicación móvil.
- **Precondiciones:**
 1. El usuario debe estar suscrito a los tópicos establecidos en el servidor.
 2. El módulo domótico debe estar en línea.
- **Flujo de eventos básicos:**

| Al actor | El sistema |
|---|---|
| 2. Escucha el mensaje y lo visualiza desde un EditText. | 1. Publica la zona horaria enviada por el módulo domótico |

Tabla 7-24 Caso de uso visualizar zona horaria - Flujo normal

- **Flujo de eventos alternativo:**

| Al actor | El sistema |
|---|--|
| 2. Visualiza la última zona horaria arribada a la aplicación móvil. | 1. Publica la zona horaria enviada por el módulo domótico. |

Tabla 7-25 Caso de uso visualizar zona horaria - Flujo alternativo

- **Postcondiciones:** El usuario será capaz de visualizar zona horaria establecida en el módulo domótico.

7.1.3.14. Caso de uso <Respuesta de sincronización>

- **Descripción:** Todos los componentes que muestran información proveniente del módulo o los componentes destinados al control de los relés, se sincronizarán con el módulo domótico.
- **Actores:** Usuario de la aplicación móvil.
- **Precondiciones:**
 1. El usuario debe estar suscrito a los tópicos establecidos en el servidor.
 2. El módulo domótico debe estar en línea.
 3. El usuario debe haber enviado un mensaje para la sincronización.
- **Flujo de eventos básicos:**

| Al actor | El sistema |
|---|--|
| 2. Obtiene el mensaje, y la aplicación móvil lo traduce a sus componentes. Ejemplo: los switches se prenden o se apagan, dependiendo del estado actual del módulo. | 1. Publica la respuesta a la solicitud de sincronización enviada por el módulo domótico. |

Tabla 7-26 Caso de uso respuesta de sincronización - Flujo normal

- **Flujo de eventos alternativo:**

| Al actor | El sistema |
|---|--|
| 2. Mantiene la configuración actual en la aplicación móvil. | 1. La respuesta de sincronización proveniente del módulo domótico nunca llega al servidor. |

Tabla 7-27 Caso de uso respuesta de sincronización - Flujo alternativo

- **Postcondiciones:** El usuario sincronizará la aplicación móvil, con el estado actual de los componentes y configuraciones del dispositivo domótico.

7.1.3.15. Caso de uso <Publicar en tópicos>

- **Descripción:** El módulo domótico podrá publicar mensajes en determinados tópicos establecidos en el servidor.
- **Actores:** Módulo domótico.
- **Precondiciones:**
 1. El módulo domótico debe haber iniciado sesión en el sistema
- **Flujo de eventos básicos:**

| Al actor | El sistema |
|---------------------------------------|--|
| 1. Publica un mensaje en el servidor. | 2. Publica el mensaje enviado por el módulo domótico a los demás usuarios suscritos. |

Tabla 7-28 Caso de uso publicar tópicos - Flujo normal

- **Flujo de eventos alternativo:**

| Al actor | El sistema |
|---------------------------------------|--|
| 1. Publica un mensaje en el servidor. | 2. Recibe el mensaje y lo publica, pero no hay otros usuarios suscritos al tópico. |

Tabla 7-29 Caso de uso publicar tópicos - Flujo alternativo

- **Postcondiciones:** El módulo publicará un mensaje satisfactoriamente en el servidor.

7.1.3.16. Caso de uso <Responder temperatura y humedad>

- **Descripción:** El módulo domótico envía al servidor la temperatura y humedad registrada (se envían es cada cinco minutos).
- **Actores:** Módulo domótico.
- **Precondiciones:**
 1. El módulo domótico debe ser capaz de publicar en tópicos establecidos en el servidor.

- **Flujo de eventos básicos:**

| Al actor | El sistema |
|--|--|
| 1. Periódicamente mandará la temperatura y humedad a un tópico del servidor, el cual estará identificado por una acción. | 2. Recibe el mensaje y lo publica, para el usuario de la aplicación móvil lo oiga. |

Tabla 7-30 Caso de uso responder temperatura y humedad - Flujo normal

- **Flujo de eventos alternativo:**

| Al actor | El sistema |
|--|---|
| 1. Periódicamente mandará la temperatura y humedad a un tópico del servidor, el cual estará identificado por una acción. | 2. Recibe el mensaje y lo publica, pero no hay usuarios en línea. |

Tabla 7-31 Caso de uso responder temperatura y humedad - Flujo normal

- **Postcondiciones:** El módulo domótico enviará al servidor la temperatura y humedad registrada (el servidor dará a conocer la temperatura y humedad al usuario de la aplicación móvil).

7.1.3.17. Caso de uso <Responder zona horaria>

- **Descripción:** El módulo domótico envía al servidor la zona horaria que tiene establecida.
- **Actores:** Módulo domótico.
- **Precondiciones:**
 1. El módulo domótico debe ser capaz de publicar en tópicos establecidos en el servidor.

- **Flujo de eventos básicos:**

| Al actor | El sistema |
|--|---|
| 1. Periódicamente mandará la zona horaria a un tópico del servidor, la cual estará identificada por una acción | 2. Recibe el mensaje y lo publica para que el usuario de la aplicación móvil lo oiga. |

Tabla 7-32 Caso de uso responder zona horaria - Flujo normal

- **Flujo de eventos alternativo:**

| Al actor | El sistema |
|--|---|
| 1. Periódicamente mandará la zona horaria a un tópico del servidor, la cual estará identificada por una acción | 2. Recibe el mensaje y lo publica, pero no hay usuarios en línea. |

Tabla 7-33 Caso de uso responder zona horaria - Flujo alternativo

- **Postcondiciones:** El módulo domótico enviará al servidor la zona horaria que tiene establecida.

7.1.3.18. Caso de uso <Responder a la sincronización>

- **Descripción:** El módulo domótico, tras haber recibido una solicitud de sincronización, responderá el estado actual de todos sus componentes (La respuesta la publica en el servidor)
- **Actores:** Módulo domótico.
- **Precondiciones:**
 1. El módulo domótico debe ser capaz de publicar en tópicos establecidos en el servidor.
- **Flujo de eventos básicos:**

| Al actor | El sistema |
|--|---|
| 1. Periódicamente mandará la zona horaria a un tópico del servidor, la cual estará identificada por una acción | 2. Recibe el mensaje y lo publica para que el usuario de la aplicación móvil lo oiga. |

Tabla 7-34 Caso de uso responder a la sincronización - Flujo normal

- **Flujo de eventos alternativo:**

| Al actor | El sistema |
|--|---|
| 1. Periódicamente mandará la zona horaria a un tópico del servidor, la cual estará identificada por una acción | 2. Recibe el mensaje y lo publica, pero no hay usuarios en línea. |

Tabla 7-35 Caso de uso responder a la sincronización - Flujo alternativo

- **Postcondiciones:** El módulo domótico envía al servidor la zona horaria que tiene establecida.

7.1.3.19. Caso de uso <Suscribirse a tópicos>

- **Descripción:** El módulo domótico estará suscrito a los tópicos del servidor, de forma que pueda escuchar las publicaciones provenientes del usuario de la aplicación móvil.
- **Actores:** Módulo domótico.
- **Precondiciones:**
 1. El módulo domótico debe haber iniciado sesión en el sistema.
- **Flujo de eventos básicos:**

| Al actor | El sistema |
|--|--|
| 1. Se suscribe a los tópicos del servidor. | 2. Publica los mensajes enviados por el usuario de la aplicación móvil, para que el módulo domótico lo oiga. |
| 3. Escucha los mensajes que el servidor publica en dichos tópicos. | |

Tabla 7-36 Caso de uso suscribirse a tópicos - Flujo normal

- **Flujo de eventos alternativo:**

| Al actor | El sistema |
|--|--|
| 1. Se suscribirá a los tópicos del servidor. | 2. No recibe mensajes por parte de los usuarios. |

Tabla 7-37 Caso de uso suscribirse a tópicos - Flujo normal

- **Postcondiciones:** El módulo domótico escuchará las publicaciones que haga el usuario de la aplicación móvil.

7.1.3.20. Caso de uso <Solicitud para automatizar relés>

- **Descripción:** Al módulo domótico arriba un mensaje que contiene los horarios de encendido y apagado para los relés.
- **Actores:** Módulo domótico.
- **Precondiciones:**
 1. El módulo domótico debe estar suscritos a los tópicos establecidos en el servidor.
- **Flujo de eventos básicos:**

| Al actor | El sistema |
|---|---|
| 2. Escucha el mensaje de automatización de los relés y lo transforma a un objeto que pueda reconocer (los relés se automatizan de forma independiente). | 1. Recibe un mensaje bajo la acción para automatizar relés y lo publica para que el módulo domótico lo escuche. |

Tabla 7-38 Caso de uso solicitud para automatizar relés - Flujo normal

- **Flujo de eventos alternativo:**

| Al actor | El sistema |
|--|--|
| 2. Está fuera de línea y no recibe el mensaje. | 1. Recibe un mensaje bajo la acción para automatizar relés, lo reenvía a los usuarios suscritos (módulo domótico). |

Tabla 7-39 Caso de uso solicitud para automatizar relés - Flujo alternativo

- **Postcondiciones:** El módulo domótico recibe un mensaje en el cual se establecen los horarios de encendido y apagado, para cada relé.

7.1.3.21. Caso de uso <Solicitud para cambiar zona horaria>

- **Descripción:** El módulo domótico recibe un mensaje que contiene la nueva zona horaria.
- **Actores:** Módulo domótico.
- **Precondiciones:**
 1. El módulo domótico debe estar suscritos a los tópicos establecidos en el servidor.

- **Flujo de eventos básicos:**

| Al actor | El sistema |
|--|--|
| 2. Escucha el mensaje destinado para el cambio de la zona horaria. | 1. Recibe un mensaje bajo la acción para cambiar la zona horaria, lo reenvía al módulo domótico. |

Tabla 7-40 Caso de uso solicitud para cambiar zona horaria - Flujo normal

- **Flujo de eventos alternativo:**

| Al actor | El sistema |
|--|--|
| 2. Está fuera de línea y no recibe el mensaje. | 1. Recibe un mensaje bajo la acción para automatizar relés, lo reenvía al módulo domótico. |

Tabla 7-41 Caso de uso solicitud para cambiar zona horaria - Flujo alternativo

- **Postcondiciones:** El módulo domótico recibe un mensaje que contiene la nueva zona horaria (el módulo domótico tiene un reloj interno).

7.1.3.22. Caso de uso <Automatizar relés>

- **Descripción:** El módulo domótico establece los horarios para el encendido y apagado automático para los relés. Luego registra los que tienen alarma activa y finalmente guarda dichos horarios, para una posible sincronización (los horarios para cada relé se establecen de forma independiente, pero viajan en un mismo mensaje).
- **Actores:** Módulo domótico.
- **Precondiciones:**
 1. El módulo debe recibir la solicitud para la automatización de los relés.
- **Flujo de eventos básicos:**

| Al actor | El sistema |
|---|--|
| 1. Automatiza el encendido y apagado de los relés y guarda los datos para una posible sincronización. | 2. Espera una posible respuesta de sincronización. |

Tabla 7-42 Caso de uso automatizar relés - Flujo normal

- **Postcondiciones:** Los relés se encenderán o apagarán de forma automática, según los horarios establecidos.

7.1.3.23. Caso de uso <Cambiar zona horaria>

- **Descripción:** El módulo domótico establece la nueva zona hora y la salva para una posible respuesta sincronización.
- **Actores:** Módulo domótico.
- **Precondiciones:**
 1. El módulo debe recibir la solicitud para cambiar la zona horaria.
- **Flujo de eventos básicos:**

| Al actor | El sistema |
|---|--|
| 1. Cambia la zona horaria establecida y la salva, para una posterior respuesta de sincronización. | 2. Espera una posible respuesta de sincronización. |

Tabla 7-43 Caso de uso cambiar zona horaria - Flujo normal

- **Postcondiciones:** El relé cambia su zona horaria y en consecuencia cambia su hora interna.

7.1.3.24. Caso de uso <Administrar usuarios>

- **Descripción:** El administrador del sistema establece comunicación con el web API ingresando al módulo de usuarios. Este por defecto mostrará el listado de usuarios registrados. Luego, dependiendo del método de la solicitud el administrador podrá agregar, modificar, eliminar y buscar usuarios
- **Actores:** Administrador
- **Precondiciones:**
 1. El administrador debe haber iniciado sesión.
- **Flujo de eventos básicos:**

| Al actor | El sistema |
|---|--|
| 1. Desde el cliente HTTP envía una solicitud GET a la ruta por defecto del módulo usuarios. | 2. Recibe la solicitud y muestra la lista de usuarios registrados. |

Tabla 7-44 Caso de uso administrar usuarios - Flujo normal

- **Flujo de eventos alternativo:**

| Al actor | El sistema |
|---|--|
| 1. Desde el cliente HTTP envía una solicitud GET a la ruta por defecto del módulo usuarios. | 2. El sistema recibe la solicitud, y responde con un mensaje 400 Bad Request (el servidor no pudo interpretar la solicitud dada una sintaxis inválida). |

Tabla 7-45 Caso de uso administrar usuarios - Flujo alternativo

- **Postcondiciones:** El administrador establece comunicación exitosamente con el módulo usuarios.

7.1.3.25. Caso de uso <Crear usuario>

- **Descripción:** El administrador registra un nuevo usuario en el sistema.
- **Actores:** Administrador
- **Precondiciones:**
 1. El administrador debe iniciar sesión en el sistema.
- **Flujo de eventos básicos:**

| Al actor | El sistema |
|---|---|
| 1. Desde el cliente HTTP envía una solicitud POST con el objeto de la persona, en formato JSON. | 2. Otorgará la posibilidad de crear usuarios con rol normal o de administrador. |

Tabla 7-46 Caso de uso crear usuario - Flujo normal

- **Flujo de eventos alternativo:**

| Al actor | El sistema |
|--|--|
| 1. Desde el cliente HTTP envía una solicitud POST con el objeto en de la persona, en formato JSON. | 2. El sistema recibe la solicitud, y responde con un mensaje 400 Bad Request (el servidor no pudo interpretar la solicitud dada una sintaxis inválida). |

Tabla 7-47 Caso de uso crear usuario - Flujo alternativo

- **Postcondiciones:** El administrador registra exitosamente un nuevo usuario en el sistema.

7.1.3.26. Caso de uso <Actualizar usuario>

- **Descripción:** El administrador actualiza un usuario del sistema.
- **Actores:** Administrador
- **Precondiciones:**
 1. El administrador debe iniciar sesión en el sistema.
 2. El usuario a actualizar debe existir
- **Flujo de eventos básicos:**

| Al actor | El sistema |
|---|---|
| 1. Desde el cliente HTTP envía una solicitud PUT con los datos del usuario a actualizar (el id de la persona a actualizar se incluye en la URL y en objeto JSON). | 2. El sistema recibe la solicitud, actualiza los datos y responde con un mensaje que contiene los datos del usuario modificado. |

Tabla 7-48 Caso de uso actualizar usuario - Flujo normal

- **Flujo de eventos alternativo:**

| Al actor | El sistema |
|---|--|
| 1. Desde el cliente HTTP envía una solicitud PUT con los datos del usuario a actualizar (el id de la persona a actualizar se incluye en la URL y en objeto JSON). | 2. El sistema recibe la solicitud, y responde con un mensaje 400 Bad Request (el servidor no pudo interpretar la solicitud dada una sintaxis inválida). |

Tabla 7-49 Caso de uso actualizar usuario - Flujo alternativo

- **Postcondiciones:** El administrador actualiza un usuario exitosamente.

7.1.3.27. Caso de uso <Eliminar usuario>

- **Descripción:** El administrador elimina un usuario del sistema.
- **Actores:** Administrador
- **Precondiciones:**
 1. El administrador debe iniciar sesión en el sistema.
 2. El usuario a eliminar debe existir.
- **Flujo de eventos básicos:**

| Al actor | El sistema |
|---|---|
| 1. Desde el cliente HTTP envía una solicitud DELETE con el id del usuario a eliminar. | 2. El sistema recibe la solicitud, elimina el usuario, y responde con un mensaje que contiene los datos del usuario removido. |

Tabla 7-50 Caso de uso eliminar usuario - Flujo normal

- **Flujo de eventos alternativo:**

| Al actor | El sistema |
|---|--|
| 1. Desde el cliente HTTP envía una solicitud DELETE con el id del usuario a eliminar. | 2. El sistema recibe la solicitud, y responde con un mensaje 400 Bad Request (el servidor no pudo interpretar la solicitud dada una sintaxis inválida). |

Tabla 7-51 Caso de uso eliminar usuario - Flujo alternativo

- **Postcondiciones:** El administrador del sistema elimina un usuario exitosamente.

7.1.3.28. Caso de uso <Buscar usuario>

- **Descripción:** El administrador busca los datos de un usuario registrado en el sistema.
- **Actores:** Administrador
- **Precondiciones:**
 1. El administrador debe iniciar sesión en el sistema.
 2. El usuario a buscar debe existir.
- **Flujo de eventos básicos:**

| Al actor | El sistema |
|--|---|
| 1. Desde el cliente HTTP envía una solicitud GET con el id del usuario a buscar. | 2. Recibe la solicitud, busca al usuario, y responde con un mensaje que contiene los datos de este. |

Tabla 7-52 Caso de uso buscar usuario - Flujo normal

- **Flujo de eventos alternativo:**

| Al actor | El sistema |
|--|--|
| 1. Desde el cliente HTTP envía una solicitud GET con el id del usuario a buscar. | 2. El sistema recibe la solicitud, y responde con un mensaje 400 Bad Request (el servidor no pudo interpretar la solicitud dada una sintaxis inválida). |

Tabla 7-53 Caso de uso buscar usuario - Flujo alternativo

- **Postcondiciones:** El administrador visualizara lo datos del usuario buscado.

7.1.3.29. Caso de uso <Crear usuario normal>

- **Descripción:** El administrador registra un usuario con rol normal.
- **Actores:** Administrador
- **Precondiciones:**
 1. El administrador debe iniciar sesión en el sistema.
 2. El administrador debe poder crear usuarios en el sistema.
- **Flujo de eventos básicos:**

| Al actor | El sistema |
|--|---|
| 1. Desde el cliente HTTP envía una solicitud POST para la creación de un usuario, especificando que tendrá un rol de usuario normal. | 2. El sistema recibe la solicitud, crea el usuario, y responde un mensaje que contiene los datos de este. |

Tabla 7-54 Caso de uso crear usuario normal - Flujo normal

- **Flujo de eventos alternativo:**

| Al actor | El sistema |
|--|---|
| 1. Desde el cliente HTTP envía una solicitud POST para la creación de un usuario, especificando que tendrá un rol de usuario normal. | 2. El sistema responde con un mensaje 400 Bad Request (el servidor no pudo interpretar la solicitud dada una sintaxis inválida). |

Tabla 7-55 Caso de uso crear usuario normal - Flujo alternativo

- **Postcondiciones:** El administrador crea un usuario exitosamente (el usuario solo tendrá permisos para controlar la aplicación móvil).

7.1.3.30. Caso de uso <Crear usuario administrador>

- **Descripción:** El administrador registra un usuario con rol administrador.
- **Actores:** Administrador
- **Precondiciones:**
 1. El administrador debe iniciar sesión en el sistema.
 2. El administrador debe poder crear usuarios en el sistema.
- **Flujo de eventos básicos:**

| Al actor | El sistema |
|--|---|
| 1. Desde el cliente HTTP envía una solicitud POST para la creación de un usuario, especificando que tendrá un rol de un administrador. | 2. El sistema recibe la solicitud, crea el usuario, y responde un mensaje que contiene los datos de este. |

Tabla 7-56 Caso de uso crear usuario administrador - Flujo normal

- **Flujo de eventos alternativo:**

| Al actor | El sistema |
|--|---|
| 1. Desde el cliente HTTP envía una solicitud POST para la creación de un usuario, especificando que tendrá un rol de un administrador. | 2. El sistema responde con un mensaje 400 Bad Request (el servidor no pudo interpretar la solicitud dada una sintaxis inválida). |

Tabla 7-57 Caso de uso crear usuario administrador - Flujo alternativo

- **Postcondiciones:** El administrador crea un usuario administrador exitosamente.

7.1.3.31. Caso de uso <Listar usuarios>

- **Descripción:** El administrador visualiza la lista de usuarios registrados en el sistema.
- **Actores:** Administrador
- **Precondiciones:**
 1. El administrador debe iniciar sesión en el sistema.
 2. Debe haber usuarios registrados en el sistema.
- **Flujo de eventos básicos:**

| Al actor | El sistema |
|---|---|
| 1. Mediante el cliente HTTP envía una solicitud GET para obtener una lista de todos los usuarios registrados en el sistema. | 2. Retorna la lista solicitada por el usuario administrador (La lista se encuentra codificada en formato JSON). |

Tabla 7-58 Caso de uso listar usuarios - Flujo normal

- **Flujo de eventos alternativo:**

| Al actor | El sistema |
|---|---|
| 1. Mediante el cliente HTTP envía una solicitud GET para obtener una lista de todos los usuarios registrados en el sistema. | 2. El sistema responde con un mensaje 400 Bad Request (el servidor no pudo interpretar la solicitud dada una sintaxis inválida). |

Tabla 7-59 Caso de uso listar usuario - Flujo alternativo

- **Postcondiciones:** El administrador visualizará una lista de todos los usuarios que se han registrado en el sistema.

7.1.3.32. Caso de uso <Administrar módulos>

- **Descripción:** El administrador del sistema establece comunicación con el web API ingresando al módulo de dispositivos domóticos. Este por defecto mostrará el listado de los dispositivos registrados. Luego, dependiendo del método de la solicitud el administrador podrá agregar, modificar, eliminar y buscar dispositivos domóticos.
- **Actores:** Administrador
- **Precondiciones:**
 1. El administrador debe haber iniciado sesión.
- **Flujo de eventos básicos:**

| Al actor | El sistema |
|--|--|
| 1. Desde el cliente HTTP envía una solicitud GET a la ruta por defecto del módulo de dispositivos. | 2. Recibe la solicitud y muestra la lista de dispositivos registrados. |

Tabla 7-60 Caso de uso administrar módulos - Flujo normal

- **Flujo de eventos alternativo:**

| Al actor | El sistema |
|--|---|
| 1. Desde el cliente HTTP envía una solicitud GET a la ruta por defecto del módulo de dispositivos. | 2. El sistema recibe la solicitud, y responde con un mensaje 400 Bad Request (el servidor no pudo interpretar la solicitud dada una sintaxis inválida). |

Tabla 7-61 Caso de uso administrar módulos - Flujo alternativo

- **Postcondiciones:** El administrador establece comunicación exitosamente con el módulo de dispositivos.

7.1.3.33. Caso de uso <Crear módulo>

- **Descripción:** El administrador registra un nuevo módulo en el sistema.
- **Actores:** Administrador
- **Precondiciones:**
 1. El administrador debe iniciar sesión en el sistema.
- **Flujo de eventos básicos:**

| Al actor | El sistema |
|--|--|
| 1. Desde el cliente HTTP envía una solicitud POST con los datos del módulo a crear (La solicitud tiene el cuerpo en formato JSON). | 2. El sistema recibe la solicitud, crea el módulo, y responde un mensaje que contiene los datos de este. |

Tabla 7-62 Caso de uso crear módulo - Flujo normal

- **Flujo de eventos alternativo:**

| Al actor | El sistema |
|--|---|
| 1. Desde el cliente HTTP envía una solicitud POST con los datos del módulo a crear (La solicitud tiene el cuerpo en formato JSON). | 2. El sistema responde con un mensaje 400 Bad Request (el servidor no pudo interpretar la solicitud dada una sintaxis inválida). |

Tabla 7-63 Caso de uso crear módulo - Flujo alternativo

- **Postcondiciones:** El administrador registra exitosamente un nuevo módulo en el sistema.

7.1.3.34. Caso de uso <Actualizar módulo>

- **Descripción:** El administrador actualiza los datos de un módulo del sistema.
- **Actores:** Administrador
- **Precondiciones:**
 1. El administrador debe iniciar sesión en el sistema.
 2. El módulo a actualizar debe existir
- **Flujo de eventos básicos:**

| Al actor | El sistema |
|---|--|
| 1. Desde el cliente HTTP envía una solicitud PUT con los datos del módulo a actualizar (el id del módulo a actualizar se incluye en la URL y en objeto JSON). | 2. El sistema recibe la solicitud, actualiza los datos y responde con un mensaje que contiene los datos del módulo modificado. |

Tabla 7-64 Caso de uso actualizar módulo - Flujo normal

- **Flujo de eventos alternativo:**

| Al actor | El sistema |
|---|--|
| 1. Desde el cliente HTTP envía una solicitud PUT con los datos del módulo a actualizar (el id del módulo a actualizar se incluye en la URL y en objeto JSON). | 2. El sistema recibe la solicitud, y responde con un mensaje 400 Bad Request (el servidor no pudo interpretar la solicitud dada una sintaxis inválida). |

Tabla 7-65 Caso de uso actualizar módulo - Flujo alternativo

- **Postcondiciones:** El administrador actualiza un módulo exitosamente.

7.1.3.35. Caso de uso <Eliminar módulo>

- **Descripción:** El administrador elimina un módulo del sistema.
- **Actores:** Administrador
- **Precondiciones:**
 1. El administrador debe iniciar sesión en el sistema.
 2. El módulo a eliminar debe existir.
- **Flujo de eventos básicos:**

| Al actor | El sistema |
|--|---|
| 1. Desde el cliente HTTP envía una solicitud DELETE con el id del módulo a eliminar. | 2. El sistema recibe la solicitud, elimina el módulo, y responde con un mensaje que contiene los datos del módulo removido. |

Tabla 7-66 Caso de uso eliminar módulo - Flujo normal

- **Flujo de eventos alternativo:**

| Al actor | El sistema |
|--|--|
| 1. Desde el cliente HTTP envía una solicitud DELETE con el id del módulo a eliminar. | 2. El sistema recibe la solicitud, y responde con un mensaje 400 Bad Request (el servidor no pudo interpretar la solicitud dada una sintaxis inválida). |

Tabla 7-67 Caso de uso actualizar módulo - Flujo alternativo

- **Postcondiciones:** El administrador del sistema elimina un módulo exitosamente.

7.1.3.36. Caso de uso <Buscar módulo>

- **Descripción:** El administrador busca los datos de un módulo registrador en el sistema.
- **Actores:** Administrador
- **Precondiciones:**
 1. El administrador debe iniciar sesión en el sistema.
 2. El módulo a buscar debe existir.
- **Flujo de eventos básicos:**

| Al actor | El sistema |
|---|--|
| 1. Desde el cliente HTTP envía una solicitud GET con el id del módulo a buscar. | 2. Recibe la solicitud, busca el módulo, y responde con un mensaje que contiene los datos de este. |

Tabla 7-68 Caso de uso buscar módulo - Flujo normal

- **Flujo de eventos alternativo:**

| Al actor | El sistema |
|---|--|
| 1. Desde el cliente HTTP envía una solicitud GET con el id del módulo a buscar. | 2. El sistema recibe la solicitud, y responde con un mensaje 400 Bad Request (el servidor no pudo interpretar la solicitud dada una sintaxis inválida). |

Tabla 7-69 Caso de uso buscar módulo - Flujo alternativo

- **Postcondiciones:** El administrador visualizara lo datos del módulo buscado.

7.1.3.37. Caso de uso <Listar módulos>

- **Descripción:** El administrador visualiza la lista de módulos registrados en el sistema.
- **Actores:** Administrador
- **Precondiciones:**
 1. El administrador debe iniciar sesión en el sistema.
 2. Debe haber módulos registrados en el sistema.
- **Flujo de eventos básicos:**

| Al actor | El sistema |
|--|---|
| 1. Mediante el cliente HTTP envía una solicitud GET para obtener una lista de todos los módulos registrados en el sistema. | 2. Retorna la lista solicitada por el usuario administrador (La lista se encuentra codificada en formato JSON). |

Tabla 7-70 Caso de uso listar módulos - Flujo normal

- **Flujo de eventos alternativo:**

| Al actor | El sistema |
|--|---|
| 1. Mediante el cliente HTTP envía una solicitud GET para obtener una lista de todos los módulos registrados en el sistema. | 2. El sistema responde con un mensaje 400 Bad Request (el servidor no pudo interpretar la solicitud dada una sintaxis inválida). |

Tabla 7-71 Caso de uso listar módulos - Flujo alternativo

- **Postcondiciones:** El administrador visualizará una lista de todos los módulos que se han registrado en el sistema.

7.1.3.38. Caso de uso <Administrar tópicos>

- **Descripción:** El administrador del sistema establece comunicación con el web API ingresando al módulo tópicos. Este por defecto mostrará el listado de los tópicos registrados. Luego, dependiendo del método de la solicitud el administrador podrá agregar, modificar, eliminar y buscar tópicos.
- **Actores:** Administrador
- **Precondiciones:**
 1. El administrador debe haber iniciado sesión.
- **Flujo de eventos básicos:**

| Al actor | El sistema |
|---|--|
| 3. Desde el cliente HTTP envía una solicitud GET a la ruta por defecto del módulo de tópicos. | 4. Recibe la solicitud y responde la lista de tópicos registrados. |

Tabla 7-72 Caso de uso administrar tópicos - Flujo normal

- **Flujo de eventos alternativo:**

| Al actor | El sistema |
|--|---|
| 3. Desde el cliente HTTP envía una solicitud GET a la ruta por defecto del módulo tópicos. | 4. El sistema recibe la solicitud, y responde con un mensaje 400 Bad Request (el servidor no pudo interpretar la solicitud dada una sintaxis inválida). |

Tabla 7-73 Caso de uso administrar tópico - Flujo alternativo

- **Postcondiciones:** El administrador establece comunicación exitosamente con el módulo de tópicos.

7.1.3.39. Caso de uso <Crear tópico>

- **Descripción:** El administrador registra un nuevo tópico en el sistema.
- **Actores:** Administrador
- **Precondiciones:**
 1. El administrador debe iniciar sesión en el sistema.
- **Flujo de eventos básicos:**

| Al actor | El sistema |
|--|--|
| 1. Desde el cliente HTTP envía una solicitud POST con los datos del tópico a crear (La solicitud tiene el cuerpo en formato JSON). | 2. El sistema recibe la solicitud, crea el tópico, y responde un mensaje que contiene los datos de este. |

Tabla 7-74 Caso de uso crear tópico - Flujo normal

- **Flujo de eventos alternativo:**

| Al actor | El sistema |
|--|---|
| 1. Desde el cliente HTTP envía una solicitud POST con los datos del tópico a crear (La solicitud tiene el cuerpo en formato JSON). | 2. El sistema responde con un mensaje 400 Bad Request (el servidor no pudo interpretar la solicitud dada una sintaxis inválida). |

Tabla 7-75 Caso de uso crear tópico - Flujo alternativo

- **Postcondiciones:** El administrador registra exitosamente un nuevo tópico en el sistema.

7.1.3.40. Caso de uso <Asignar tipo de dato>

- **Descripción:** El administrador debe asignar un tipo de dato al tópico que se está creando.
- **Actores:** Administrador
- **Precondiciones:**
 1. El administrador debe estar creando un tópico.
 2. El tipo de dato debe existir en el sistema.
- **Flujo de eventos básicos:**

| Al actor | El sistema |
|--|--|
| 1. Durante la creación del tópico, el administrador debe asignar el id de un tipo de dato. | 2. El sistema reconoce que el tipo de dato existe y lo asigna correctamente. |

Tabla 7-76 Caso de uso asignar tipo de dato - Flujo normal

- **Flujo de eventos alternativo:**

| Al actor | El sistema |
|--|--|
| 1. Durante la creación del tópico, el administrador debe asignar el id de un tipo de dato. | 2. El sistema no reconoce la existencia del tópico, y retorna un mensaje 400 Bad Request (el servidor no pudo interpretar la solicitud dada una sintaxis inválida). |

Tabla 7-77 Caso de uso asignar tipo de dato - Flujo alternativo

- **Postcondiciones:** El administrador asignara un tipo de dato a un tópico.
 - **Ejemplo:** al tópico humedad se asigna el tipo de dato float.

7.1.3.41. Caso de uso <Actualizar tópico>

- **Descripción:** El administrador actualiza los datos de un tópico del sistema.
- **Actores:** Administrador
- **Precondiciones:**
 1. El administrador debe iniciar sesión en el sistema.
 2. El tópico a actualizar debe existir
- **Flujo de eventos básicos:**

| Al actor | El sistema |
|---|--|
| 1. Desde el cliente HTTP envía una solicitud PUT con los datos del tópico a actualizar (el id del módulo a actualizar se incluye en la URL y en objeto JSON). | 2. El sistema recibe la solicitud, actualiza los datos y responde con un mensaje que contiene los datos del tópico modificado. |

Tabla 7-78 Caso de uso actualizar tópico - Flujo normal

- **Flujo de eventos alternativo:**

| Al actor | El sistema |
|---|--|
| 1. Desde el cliente HTTP envía una solicitud PUT con los datos del tópico a actualizar (el id del módulo a actualizar se incluye en la URL y en objeto JSON). | 2. El sistema recibe la solicitud, y responde con un mensaje 400 Bad Request (el servidor no pudo interpretar la solicitud dada una sintaxis inválida). |

Tabla 7-79 Caso de uso actualizar tópico - Flujo alternativo

- **Postcondiciones:** El administrador actualiza un tópico exitosamente.

7.1.3.42. Caso de uso <Eliminar tópico>

- **Descripción:** El administrador elimina un tópico del sistema.
- **Actores:** Administrador
- **Precondiciones:**
 1. El administrador debe iniciar sesión en el sistema.
 2. El tópico a eliminar debe existir.
- **Flujo de eventos básicos:**

| Al actor | El sistema |
|--|---|
| 1. Desde el cliente HTTP envía una solicitud DELETE con el id del tópico a eliminar. | 2. El sistema recibe la solicitud, elimina el tópico, y responde con un mensaje que contiene los datos de este. |

Tabla 7-80 Caso de uso eliminar tópico - Flujo normal

- **Flujo de eventos alternativo:**

| Al actor | El sistema |
|--|--|
| 1. Desde el cliente HTTP envía una solicitud DELETE con el id del tópico a eliminar. | 2. El sistema recibe la solicitud, y responde con un mensaje 400 Bad Request (el servidor no pudo interpretar la solicitud dada una sintaxis inválida). |

Tabla 7-81 Caso de uso eliminar tópico - Flujo alternativo

- **Postcondiciones:** El administrador del sistema elimina un tópico exitosamente.

7.1.3.43. Caso de uso <Buscar tópico>

- **Descripción:** El administrador busca los datos de un tópico registrador en el sistema.
- **Actores:** Administrador
- **Precondiciones:**
 1. El administrador debe iniciar sesión en el sistema.
 2. El tópico a buscar debe existir.
- **Flujo de eventos básicos:**

| Al actor | El sistema |
|---|--|
| 1. Desde el cliente HTTP envía una solicitud GET con el id del tópico a buscar. | 2. Recibe la solicitud, busca el tópico, y responde con un mensaje que contiene los datos de este. |

Tabla 7-82 Caso de uso buscar tópico - Flujo normal

- **Flujo de eventos alternativo:**

| Al actor | El sistema |
|---|--|
| 1. Desde el cliente HTTP envía una solicitud GET con el id del tópico a buscar. | 2. El sistema recibe la solicitud, y responde con un mensaje 400 Bad Request (el servidor no pudo interpretar la solicitud dada una sintaxis inválida). |

Tabla 7-83 Caso de uso buscar tópico - Flujo alternativo

- **Postcondiciones:** El administrador visualizara lo datos del tópico buscado.

7.1.3.44. Caso de uso <Listar tópicos>

- **Descripción:** El administrador visualiza la lista de usuarios registrados en el sistema.
- **Actores:** Administrador
- **Precondiciones:**
 1. El administrador debe iniciar sesión en el sistema.
 2. Debe haber usuarios registrados en el sistema.
- **Flujo de eventos básicos:**

| Al actor | El sistema |
|---|---|
| 1. Mediante el cliente HTTP envía una solicitud GET para obtener una lista de todos los usuarios registrados en el sistema. | 2. Retorna la lista solicitada por el usuario administrador (La lista se encuentra codificada en formato JSON). |

Tabla 7-84 Caso de uso listar tópicos - Flujo normal

- **Flujo de eventos alternativo:**

| Al actor | El sistema |
|---|---|
| 1. Mediante el cliente HTTP envía una solicitud GET para obtener una lista de todos los usuarios registrados en el sistema. | 2. El sistema responde con un mensaje 400 Bad Request (el servidor no pudo interpretar la solicitud dada una sintaxis inválida). |

Tabla 7-85 Caso de uso listar tópicos - Flujo alternativo

- **Postcondiciones:** El administrador visualizará una lista de todos los usuarios que se han registrado en el sistema.

7.1.3.45. Caso de uso <Administrar tipos de datos>

- **Descripción:** El administrador del sistema establece comunicación con el web API ingresando al módulo tipos de datos. Este por defecto mostrará el listado de los tipos de datos registrados. Luego, dependiendo del método de la solicitud el administrador podrá agregar, modificar, eliminar y buscar tipos de datos.
- **Actores:** Administrador
- **Precondiciones:**
 1. El administrador debe haber iniciado sesión.
- **Flujo de eventos básicos:**

| Al actor | El sistema |
|--|--|
| 5. Desde el cliente HTTP envía una solicitud GET a la ruta por defecto del módulo de tipos de datos. | 6. Recibe la solicitud y muestra la lista de tipos de datos registrados. |

Tabla 7-86 Caso de uso administrar tipos de datos - Flujo normal

- **Flujo de eventos alternativo:**

| Al actor | El sistema |
|--|---|
| 5. Desde el cliente HTTP envía una solicitud GET a la ruta por defecto del módulo de tipos de datos. | 6. El sistema recibe la solicitud, y responde con un mensaje 400 Bad Request (el servidor no pudo interpretar la solicitud dada una sintaxis inválida). |

Tabla 7-87 Caso de uso administrar tipos de datos - Flujo alternativo

- **Postcondiciones:** El administrador establece comunicación exitosamente con el módulo de tipos de datos.

7.1.3.46. Caso de uso <Crear tipo de dato>

- **Descripción:** El administrador registra un nuevo tipo de dato en el sistema.
- **Actores:** Administrador
- **Precondiciones:**
 1. El administrador debe iniciar sesión en el sistema.
- **Flujo de eventos básicos:**

| Al actor | El sistema |
|---|--|
| 1. Desde el cliente HTTP envía una solicitud POST con los campos completos del tipo de dato a crear (La solicitud tiene el cuerpo en formato JSON). | 2. El sistema recibe la solicitud, crea el tipo de dato, y responde un mensaje que contiene los datos de este. |

Tabla 7-88 Caso de uso crear tipo de dato - Flujo normal

- **Flujo de eventos alternativo:**

| Al actor | El sistema |
|---|---|
| 1. Desde el cliente HTTP envía una solicitud POST con los campos completos del tipo de dato a crear (La solicitud tiene el cuerpo en formato JSON). | 2. El sistema responde con un mensaje 400 Bad Request (el servidor no pudo interpretar la solicitud dada una sintaxis inválida). |

Tabla 7-89 Caso de uso crear tipo de dato - Flujo alternativo

- **Postcondiciones:** El administrador registra exitosamente un nuevo tipo de dato en el sistema.

7.1.3.47. Caso de uso <Actualizar tipo de dato>

- **Descripción:** El administrador actualiza los valores de un tipo de dato del sistema.
- **Actores:** Administrador
- **Precondiciones:**
 1. El administrador debe iniciar sesión en el sistema.
 2. El tipo de dato a actualizar debe existir
- **Flujo de eventos básicos:**

| Al actor | El sistema |
|---|--|
| 1. Desde el cliente HTTP envía una solicitud PUT con los valores del tipo de dato a actualizar (el id del tipo de dato a actualizar se incluye en la URL y en objeto JSON). | 2. El sistema recibe la solicitud, actualiza los datos y responde con un mensaje que contiene los valores del tipo de dato modificado. |

Tabla 7-90 Caso de uso actualizar tipo de dato - Flujo normal

- **Flujo de eventos alternativo:**

| Al actor | El sistema |
|---|--|
| 1. Desde el cliente HTTP envía una solicitud PUT con los valores del tipo de dato a actualizar (el id del tipo de dato a actualizar se incluye en la URL y en objeto JSON). | 2. El sistema recibe la solicitud, y responde con un mensaje 400 Bad Request (el servidor no pudo interpretar la solicitud dada una sintaxis inválida). |

Tabla 7-91 Caso de uso actualizar tipo de dato - Flujo alternativo

- **Postcondiciones:** El administrador actualiza un tipo de dato exitosamente.

7.1.3.48. Caso de uso <Eliminar tipo de dato>

- **Descripción:** El administrador elimina un tipo de dato del sistema.
- **Actores:** Administrador
- **Precondiciones:**
 1. El administrador debe iniciar sesión en el sistema.
 2. El tipo de dato a eliminar debe existir.
- **Flujo de eventos básicos:**

| Al actor | El sistema |
|--|---|
| 1. Desde el cliente HTTP envía una solicitud DELETE con el id del tipo de dato a eliminar. | 2. El sistema recibe la solicitud, elimina el tipo de dato, y responde con un mensaje que contiene los valores del tipo de dato removido. |

Tabla 7-92 Caso de uso eliminar tipo de dato - Flujo normal

- **Flujo de eventos alternativo:**

| Al actor | El sistema |
|--|--|
| 1. Desde el cliente HTTP envía una solicitud DELETE con el id del tipo de dato a eliminar. | 2. El sistema recibe la solicitud, y responde con un mensaje 400 Bad Request (el servidor no pudo interpretar la solicitud dada una sintaxis inválida). |

Tabla 7-93 Caso de uso eliminar tipo de dato - Flujo alternativo

- **Postcondiciones:** El administrador del sistema elimina un tipo de dato exitosamente.

7.1.3.49. Caso de uso <Buscar tipo de dato>

- **Descripción:** El administrador busca los valores de un tipo de dato registrador en el sistema.
- **Actores:** Administrador
- **Precondiciones:**
 1. El administrador debe iniciar sesión en el sistema.
 2. El tipo de dato a buscar debe existir.
- **Flujo de eventos básicos:**

| Al actor | El sistema |
|---|---|
| 1. Desde el cliente HTTP envía una solicitud GET con el id del tipo de dato a buscar. | 2. El sistema recibe la solicitud, busca el tipo de dato, y responde con un mensaje que contiene los valores de este. |

Tabla 7-94 Caso de uso buscar tipo de dato - Flujo normal

- **Flujo de eventos alternativo:**

| Al actor | El sistema |
|---|--|
| 1. Desde el cliente HTTP envía una solicitud GET con el id del tipo de dato a buscar. | 2. El sistema recibe la solicitud, y responde con un mensaje 400 Bad Request (el servidor no pudo interpretar la solicitud dada una sintaxis inválida). |

Tabla 7-95 Caso de uso buscar tipo de dato - Flujo alternativo

- **Postcondiciones:** El administrador visualizara lo valores del tipo de dato buscado.

7.1.3.50. Caso de uso <Listar tipos de datos>

- **Descripción:** El administrador visualiza la lista de tipos de datos registrados en el sistema.
- **Actores:** Administrador
- **Precondiciones:**
 1. El administrador debe iniciar sesión en el sistema.
 2. Debe haber tipos de datos registrados en el sistema.
- **Flujo de eventos básicos:**

| Al actor | El sistema |
|---|---|
| 1. Mediante el cliente HTTP envía una solicitud GET para obtener una lista de todos los tipos de datos registrados en el sistema. | 2. Retorna la lista solicitada por el usuario administrador (La lista se encuentra codificada en formato JSON). |

Tabla 7-96 Caso de uso listar tipos de datos - Flujo normal

- **Flujo de eventos alternativo:**

| Al actor | El sistema |
|---|---|
| 1. Mediante el cliente HTTP envía una solicitud GET para obtener una lista de todos los tipos de datos registrados en el sistema. | 2. El sistema responde con un mensaje 400 Bad Request (el servidor no pudo interpretar la solicitud dada una sintaxis inválida). |

Tabla 7-97 Caso de uso listar tipos de datos - Flujo alternativo

- **Postcondiciones:** El administrador visualizará una lista de todos los tipos de datos que se han registrado en el sistema.

7.1.3.51. Caso de uso <Administrar asignación de módulos a usuarios>

- **Descripción:** El administrador del sistema podrá asignar módulos a los usuarios registrados.
- **Actores:** Administrador
- **Precondiciones:**
 1. El administrador debe haber iniciado sesión en el sistema.
 2. El usuario debe estar registrado en el sistema.
 3. Los módulos a asignar deben existir en el sistema.
- **Flujo de eventos básicos:**

| Al actor | El sistema |
|--|--|
| 1. Desde el cliente HTTP envía una solicitud POST con el Id del módulo y del usuario a vincular. | 2. El sistema recibe la solicitud, asigna el módulo al usuario y responde un mensaje 200 OK . |

Tabla 7-98 Caso de uso administrar asignación de módulos a usuario – Flujo normal

- **Flujo de eventos alternativo:**

| Al actor | El sistema |
|--|---|
| 1. Desde el cliente HTTP envía una solicitud POST con el Id del módulo y del usuario a vincular. | 2. El sistema responde con un mensaje 400 Bad Request (el servidor no pudo interpretar la solicitud dada una sintaxis inválida). |

Tabla 7-99 Caso de uso administrar asignación de módulos a usuario – Flujo alternativo

- **Postcondiciones:** El administrador asigna un módulo a un usuario exitosamente.

7.1.3.52. Caso de uso <Asignar módulo a usuario>

- **Descripción:**
- **Actores:** Administrador
- **Precondiciones:**
 1. El administrador debe haber iniciado sesión en el sistema.
 2. El usuario debe estar registrado en el sistema.
 3. El módulo a asignar debe existir en el sistema.
- **Flujo de eventos básicos:**

| Al actor | El sistema |
|---|---|
| 1. Mediante una solicitud POST (desde el cliente http) podrá asignar un módulo a un usuario en particular. El cuerpo de la solicitud debe estar en formato JSON y además debe contener los id a relacionar. | 2. El sistema recibe la solicitud, asigna el módulo al usuario, y responde un mensaje que contiene los datos de dicha asignación. |

Tabla 7-100 Caso de uso asignar módulo a usuario – Flujo normal

- **Flujo de eventos alternativo:**

| Al actor | El sistema |
|---|--|
| 1. Mediante una solicitud POST (desde el cliente http) podrá asignar un módulo a un usuario en particular. El cuerpo de la solicitud debe estar en formato JSON y además debe contener los id a relacionar. | 2. El sistema recibe la solicitud, y responde con un mensaje 400 Bad Request (el servidor no pudo interpretar la solicitud dada una sintaxis inválida). |

Tabla 7-101 Caso de uso asignar módulo a usuario – Flujo alternativo

- **Postcondiciones:** El administrador del sistema asigna exitosamente un módulo a un usuario.

7.1.3.53. Caso de uso <Actualizar asignación de módulo a usuario>

- **Descripción:** El administrador cambia un módulo asignado a un usuario.
- **Actores:** Administrador
- **Precondiciones:**
 1. El administrador debe iniciar sesión en el sistema.
 2. El usuario al cual se le cambiará el módulo debe existir.
 3. El módulo a cambiar debe existir

- **Flujo de eventos básicos:**

| Al actor | El sistema |
|--|---|
| 1. Desde el cliente HTTP envía una solicitud PUT con que contiene el id del usuario y del módulo asignado actualmente, y el id del nuevo módulo. | 2. Recibe la solicitud, actualiza los datos y responde con un mensaje que contiene los datos de la asignación |

Tabla 7-102 Caso de uso actualizar asignación de módulo a usuario – Flujo normal

- **Flujo de eventos alternativo:**

| Al actor | El sistema |
|---|--|
| 1. Desde el cliente HTTP envía una solicitud PUT con que contiene el id del usuario y del módulo asignado actualmente, y el id del nuevo módulo | 2. El sistema recibe la solicitud, y responde con un mensaje 400 Bad Request (el servidor no pudo interpretar la solicitud dada una sintaxis inválida). |

Tabla 7-103 Caso de uso actualizar asignación de módulo a usuario – Flujo alternativo

- **Postcondiciones:** El administrador cambia exitosamente un módulo asignado a un usuario.

7.1.3.54. Caso de uso <Eliminar asignación de módulo a usuario>

- **Descripción:** El administrador elimina un módulo asignado a un usuario.
- **Actores:** Administrador
- **Precondiciones:**
 1. El administrador debe iniciar sesión en el sistema.
 2. La asignación que se quiere eliminar debe existir.
- **Flujo de eventos básicos:**

| Al actor | El sistema |
|---|---|
| 1. Desde el cliente HTTP envía una solicitud DELETE con el id del módulo y del usuario. | 2. El sistema recibe la solicitud, elimina la asignación, y responde con un mensaje que contiene los datos de esta. |

- **Flujo de eventos alternativo:**

| Al actor | El sistema |
|---|--|
| 1. Desde el cliente HTTP envía una solicitud DELETE con el id del módulo y del usuario. | 2. El sistema recibe la solicitud, y responde con un mensaje 400 Bad Request (el servidor no pudo interpretar la solicitud dada una sintaxis inválida). |

- **Postcondiciones:** El administrador del sistema elimina la asignación de un módulo exitosamente.

7.1.3.55. Caso de uso <Listar módulos asignados a usuario>

- **Descripción:** El administrador visualizara una lista de los módulos que están asignados a un usuario.
- **Actores:** Administrador
- **Precondiciones:**
 1. El administrador debe iniciar sesión en el sistema.
 2. Las asignaciones deben existir en el sistema.
- **Flujo de eventos básicos:**

| Al actor | El sistema |
|---|---|
| 1. Mediante el cliente HTTP envía una solicitud GET para obtener una lista de todos los módulos asignados a un usuario. | 2. Retorna la lista solicitada por el usuario administrador (La lista se encuentra codificada en formato JSON). |

Tabla 7-104 Caso de uso listar módulos asignados a usuario – Flujo normal

- **Flujo de eventos alternativo:**

| Al actor | El sistema |
|---|---|
| 1. Mediante el cliente HTTP envía una solicitud GET para obtener una lista de todos los módulos asignados a un usuario. | 2. El sistema responde con un mensaje 400 Bad Request (el servidor no pudo interpretar la solicitud dada una sintaxis inválida). |

Tabla 7-105 Caso de uso listar módulos asignados a usuario – Flujo alternativo

- **Postcondiciones:** El administrador visualizará una lista de todos los módulos que tiene asignado un usuario.

7.1.3.56. Caso de uso <Administrar asignación de tópicos a módulos>

- **Descripción:** El administrador del sistema podrá asignar tópicos a los módulos registrados.
- **Actores:** Administrador
- **Precondiciones:**
 1. El administrador debe haber iniciado sesión en el sistema.
 2. El módulo (dispositivo domótico) debe estar registrado en el sistema.
 3. Los tópicos a asignar deben existir en el sistema.
- **Flujo de eventos básicos:**

| Al actor | El sistema |
|--|--|
| Desde el cliente HTTP envía una solicitud POST con el Id del módulo y del tópico a vincular. | El sistema recibe la solicitud, asigna el tópico al módulo y responde un mensaje 200 OK . |

Tabla 7-106 Caso de uso administrar asignación de tópicos a módulos - Flujo normal

- **Flujo de eventos alternativo:**

| Al actor | El sistema |
|--|--|
| Desde el cliente HTTP envía una solicitud POST con el Id del módulo y del tópico a vincular. | El sistema responde con un mensaje 400 Bad Request (el servidor no pudo interpretar la solicitud dada una sintaxis inválida). |

Tabla 7-107 Caso de uso administrar asignación de tópicos a módulos - Flujo alternativo

- **Postcondiciones:** El administrador asigna un tópico a un módulo exitosamente.

7.1.3.57. Caso de uso <Asignar tópico a módulo>

- **Descripción:**
- **Actores:** Administrador
- **Precondiciones:**
 1. El administrador debe haber iniciado sesión en el sistema.
 2. El usuario debe estar registrado en el sistema.
 3. El módulo a asignar debe existir en el sistema.
- **Flujo de eventos básicos:**

| Al actor | El sistema |
|---|---|
| 3. Mediante una solicitud POST (desde el cliente http) podrá asignar un módulo a un usuario en particular. El cuerpo de la solicitud debe estar en formato JSON y además debe contener los id a relacionar. | 4. El sistema recibe la solicitud, asigna el módulo al usuario, y responde un mensaje que contiene los datos de dicha asignación. |

Tabla 7-108 Caso de uso asignar tópico a módulo - Flujo normal

- **Flujo de eventos alternativo:**

| Al actor | El sistema |
|---|--|
| 3. Mediante una solicitud POST (desde el cliente http) podrá asignar un módulo a un usuario en particular. El cuerpo de la solicitud debe estar en formato JSON y además debe contener los id a relacionar. | 4. El sistema recibe la solicitud, y responde con un mensaje 400 Bad Request (el servidor no pudo interpretar la solicitud dada una sintaxis inválida). |

Tabla 7-109 Caso de uso asignar tópico a módulo - Flujo alternativo

- **Postcondiciones:** El administrador del sistema asigna exitosamente un módulo a un usuario.

7.1.3.58. Caso de uso <Actualizar asignación de tópico a módulo >

- **Descripción:** El administrador cambia un módulo asignado a un usuario.
- **Actores:** Administrador
- **Precondiciones:**
 1. El administrador debe iniciar sesión en el sistema.
 2. El usuario al cual se le cambiará el módulo debe existir.
 3. El módulo a cambiar debe existir

- **Flujo de eventos básicos:**

| Al actor | El sistema |
|--|---|
| 3. Desde el cliente HTTP envía una solicitud PUT con que contiene el id del usuario y del módulo asignado actualmente, y el id del nuevo módulo. | 4. Recibe la solicitud, actualiza los datos y responde con un mensaje que contiene los datos de la asignación |

Tabla 7-110 Caso de uso actualizar asignación de tópico a módulo - Flujo normal

- **Flujo de eventos alternativo:**

| Al actor | El sistema |
|---|--|
| 3. Desde el cliente HTTP envía una solicitud PUT con que contiene el id del usuario y del módulo asignado actualmente, y el id del nuevo módulo | 4. El sistema recibe la solicitud, y responde con un mensaje 400 Bad Request (el servidor no pudo interpretar la solicitud dada una sintaxis inválida). |

Tabla 7-111 Caso de uso actualizar asignación de tópico a módulo - Flujo alternativo

- **Postcondiciones:** El administrador cambia exitosamente un módulo asignado a un usuario.

7.1.3.59. Caso de uso <Eliminar asignación de tópico a módulo >

- **Descripción:** El administrador elimina un módulo asignado a un usuario.
- **Actores:** Administrador
- **Precondiciones:**
 1. El administrador debe iniciar sesión en el sistema.
 2. La asignación que se quiere eliminar debe existir.
- **Flujo de eventos básicos:**

| Al actor | El sistema |
|---|---|
| 3. Desde el cliente HTTP envía una solicitud DELETE con el id del módulo y del usuario. | 4. El sistema recibe la solicitud, elimina la asignación, y responde con un mensaje que contiene los datos de esta. |

Tabla 7-112 Caso de uso eliminar asignación de tópico a módulo - Flujo normal

- **Flujo de eventos alternativo:**

| Al actor | El sistema |
|---|--|
| 3. Desde el cliente HTTP envía una solicitud DELETE con el id del módulo y del usuario. | 4. El sistema recibe la solicitud, y responde con un mensaje 400 Bad Request (el servidor no pudo interpretar la solicitud dada una sintaxis inválida). |

Tabla 7-113 Caso de uso eliminar asignación de tópico a módulo - Flujo alternativo

- **Postcondiciones:** El administrador del sistema elimina la asignación de un módulo exitosamente.

7.1.3.60. Caso de uso <Listar tópicos asignados a módulo >

- **Descripción:** El administrador visualizara una lista de los módulos que están asignados a un usuario.
- **Actores:** Administrador
- **Precondiciones:**
 1. El administrador debe iniciar sesión en el sistema.
 2. Las asignaciones deben existir en el sistema.
- **Flujo de eventos básicos:**

| Al actor | El sistema |
|---|---|
| 1. Mediante el cliente HTTP envía una solicitud GET para obtener una lista de todos los módulos asignados a un usuario. | 2. Retorna la lista solicitada por el usuario administrador (La lista se encuentra codificada en formato JSON). |

Tabla 7-114 Caso de uso listar tópicos asignados a módulos - Flujo normal

- **Flujo de eventos alternativo:**

| Al actor | El sistema |
|---|--|
| 1. Mediante el cliente HTTP envía una solicitud GET para obtener una lista de todos los módulos asignados a un usuario. | 2. El sistema responde con un mensaje 400 Bad Request (el servidor no pudo interpretar la solicitud dada una sintaxis inválida). |

Tabla 7-115 Caso de uso listar tópicos asignados a módulos - Flujo alternativo

- **Postcondiciones:** El administrador visualizará una lista de todos los módulos que tiene asignado un usuario.

7.1.3.61. Caso de uso <Mensaje borrar horarios automáticos relés>

- **Descripción:** El usuario manda un mensaje al servidor, que le permitirá eliminar los horarios de cualquiera de los ocho relés del módulo domótico.
- **Actores:** Usuario de la aplicación móvil.
- **Precondiciones:**
 1. El usuario debe poder publicar en tópicos.
 2. El módulo domótico debe estar en línea.
- **Flujo de eventos básicos:**

| Al actor | El sistema |
|--|---|
| 1. Selecciona mediante la aplicación móvil el botón con forma de basurero, del relé al cual quiere borrar el horario y luego presiona enviar | 2. El sistema recibe el mensaje y lo republica a los usuarios que estén suscritos a dichos tópicos (módulo domótico). |

Tabla 7-116 Caso de uso mensaje horarios automáticos relé - Flujo normal

- **Flujo de eventos alternativo:**

| Al actor | El sistema |
|--|--|
| 1. Selecciona mediante la aplicación móvil el botón con forma de basurero, del relé al cual quiere borrar el horario y luego presiona enviar | 2. Recibe el mensaje y lo reenvía, pero no hay otros usuarios suscritos al tópico. |

Tabla 7-117 Caso de uso mensaje horarios automáticos relé - Flujo alternativo

- **Postcondiciones:** El usuario quitara el encendido y apagado automático de un relé exitosamente.

7.1.3.62. Caso de uso <Solicitud para desactivar horarios automáticos relés>

- **Descripción:** Al módulo domótico arriba un mensaje que contiene el identificador del relé al que se le borrarán los horarios de encendido y apagado.
- **Actores:** Módulo domótico.
- **Precondiciones:**
 1. El módulo domótico debe estar suscritos a los tópicos establecidos en el servidor.
- **Flujo de eventos básicos:**

| Al actor | El sistema |
|---|--|
| 2. Escucha el mensaje de desactivación de horarios y lo transforma a un objeto que pueda reconocer. | 1. Recibe un mensaje bajo la acción para desactivar horarios, lo reenvía al módulo domótico. |

Tabla 7-118 Caso de uso solicitud para desactivar horarios automáticos relés - Flujo normal

- **Flujo de eventos alternativo:**

| Al actor | El sistema |
|--|--|
| 2. Está fuera de línea y no recibe el mensaje. | 1. Recibe un mensaje bajo la acción para desactivar horarios, lo reenvía al módulo domótico. |

Tabla 7-119 Caso de uso solicitud para desactivar horarios automáticos relés - Flujo alternativo

- **Postcondiciones:** El módulo domótico recibe un mensaje exitosamente, este posee el id del relé al que se le desactivara el horario.

7.1.3.63. Caso de uso <Desactivar horarios>

- **Descripción:** El módulo domótico elimina los horarios para el encendido y apagado automático para los relés.
- **Actores:** Módulo domótico.
- **Precondiciones:**
 1. El módulo debe recibir la solicitud para la desactivar horarios automáticos de los relés.
- **Flujo de eventos básicos:**

| Al actor | El sistema |
|--|--|
| 1. Elimina los horarios de encendido y apagado de los relés. También los remueve del registro de sincronización. | 2. Espera una posible respuesta de sincronización. |

Tabla 7-120 Caso de uso desactivar horarios - Flujo normal

- **Postcondiciones:** El módulo domótico no encenderá automáticamente el relé.

7.1.3.64. Caso de uso <Control remoto>

- **Descripción:** El módulo domótico prendera o apagará uno o varios relés.
- **Actores:** Módulo domótico.
- **Precondiciones:**
 1. El módulo debe recibir la solicitud para la controlar los relés.
- **Flujo de eventos básicos:**

| Al actor | El sistema |
|------------------------------|---|
| 1. Prende o apaga los relés. | 2. Espera una notificación de variables de ambiente, valor GTM o una respuesta de sincronización. |

Tabla 7-121 Caso de uso control remoto - Flujo normal

- **Postcondiciones:** El módulo domótico prenderá o apagará sus relés de forma satisfactoria.

7.1.4. Modelamiento de datos

En el presente apartado de detallarán las entidades que estarán presente en el ecosistema domótico (Usuarios, Módulos, Tópicos, Tipos de Datos y Roles). Luego, se procederá a explicar en lenguaje natural, las relaciones que vinculan a dichas entidades.

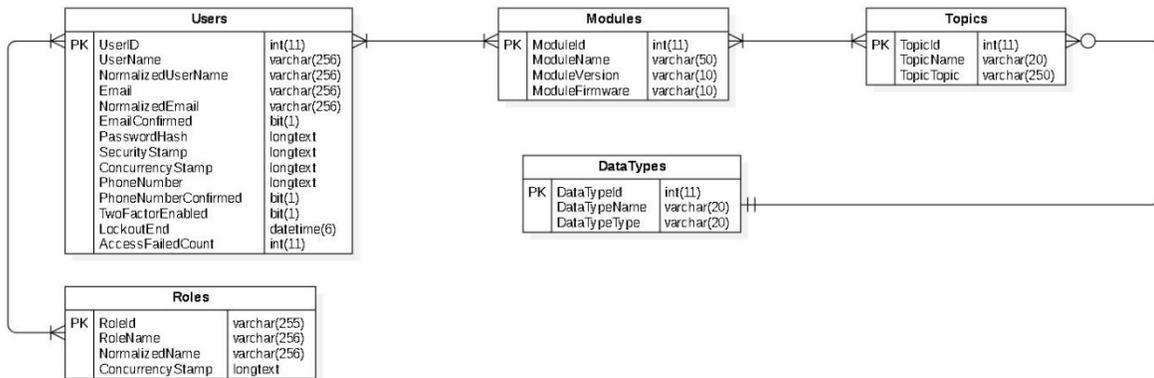


Ilustración 7-9 Modelo entidad relación, notación de Martin y notación patas de gallo

Como se aprecia en la **ilustración 7-8**, se ha optado por utilizar una nomenclatura en la cual está presente el idioma inglés, y en donde el nombre de cada atributo de una entidad está precedido por el nombre de esta (cada palabra empieza con una letra mayúscula, tanto en entidades como en atributos). El Listado de entidades es el siguiente:

- **Users:** Esta entidad representa a los usuarios que interactuarán con el sistema, estos pueden ser el administrador y el usuario de la aplicación móvil. Sin embargo, en esta entidad no se marca la diferencia(rol) entre un usuario app móvil y un administrador.
- **Modules:** Los módulos representan a los dispositivos domóticos que se crearán, y que serán utilizados por los usuarios de la aplicación móvil (en el presente proyecto solo se crea un módulo, sin embargo, el sistema admite n cantidad de módulos).
- **Topics:** Los tópicos representan las temáticas que maneja cada módulo domótico, y actúan como canal de comunicación mediante el cual los usuarios de la aplicación móvil podrán visualizar mensajes (mensajes mandados por el módulo domótico) o mandar mensajes (Mensaje para el control y automatización del módulo demótico).

- **DataTypes:** Esta entidad representa los tipos de datos que serán asignados a los tópicos. El propósito de esta entidad inicialmente es tener un registro de del tipo de valor que se manda sobre cada tópico; por ejemplo: El tópico temperatura y humedad, tiene un tipo de dato flotante (float).
- **Roles:** Esta es la entidad que marca la diferencia entre los usuarios del sistema, y tiene como propósito asignar un rol a cada usuario, dicho rol establecerá el nivel de acceso que tendrá cada estos(usuarios). Los tipos de roles que existirán son: Administrador, para un usuario que administra el sistema completamente (incluido el módulo domótico) y normal, para un usuario que controla el módulo domótico.

En lenguaje natural, las relaciones de las entidades se definen de la siguiente forma: Un usuario podrá tener asignado uno o muchos módulos, un módulo puede ser asignado a uno o muchos usuarios; en un módulo pueden existir uno o muchos tópicos, un tópico puede existir en uno o muchos módulos; un tópico puede tener solo un tipo de dato, un tipo de dato puede pertenecer a muchos tópicos y un usuario puede tener uno o muchos roles, un rol puede ser asignado a uno o muchos usuarios.

CAPÍTULO 8: DISEÑO

8.1. Diseño físico de la base de datos

En la ilustración 8-1 se presenta el modelo físico de la base de datos del ecosistema domótico, el cual se ha obtenido a partir del modelo entidad relación ilustrado en el capítulo anterior, y sirve para entender la realidad que está siendo abstraída, desde un punto de vista técnico.

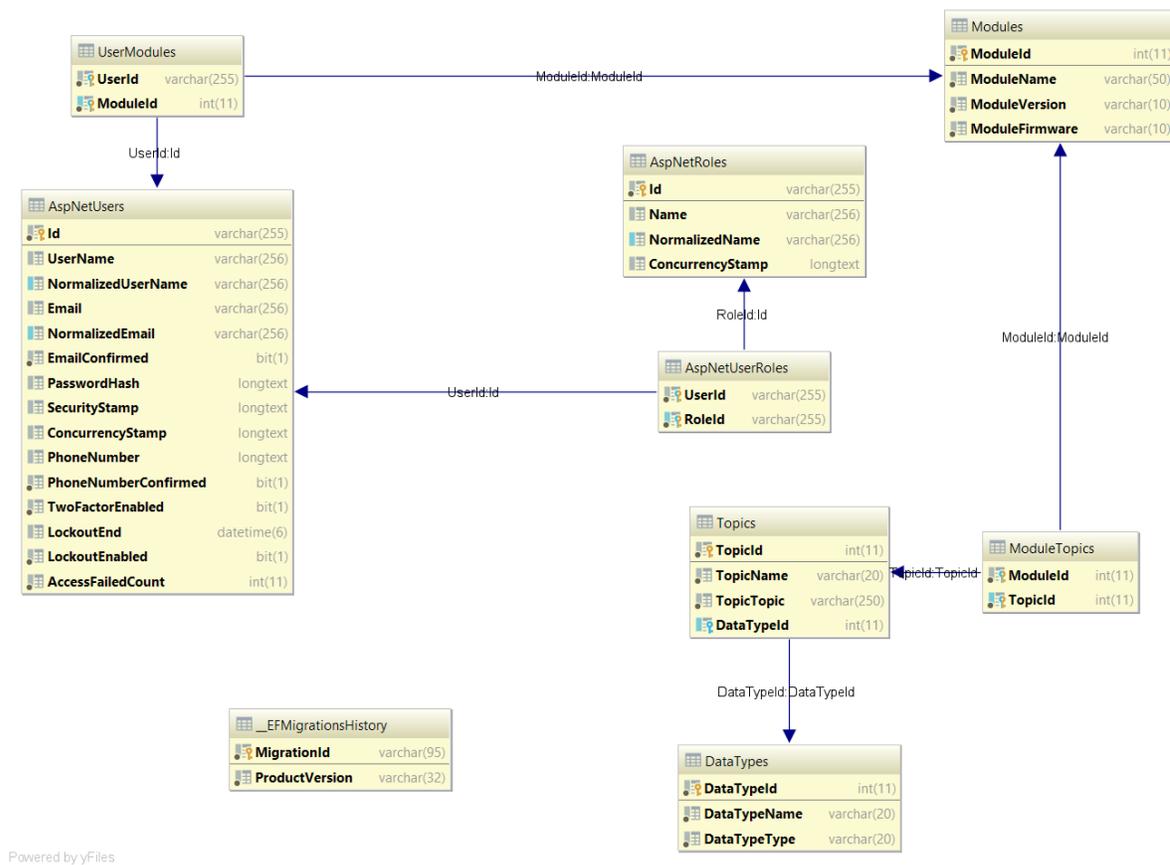


Ilustración 8-1 Modelo relacional del ecosistema domótico

En la ilustración anterior se muestran cinco tablas principales (AspNetUsers, Modules, Tópicos, DataTypes & AspNetRoles) y otras tres obtenidas a partir de las relaciones y sus respectivas cardinalidades (UserModules, MóduloTopics, AspNetUserRoles). Las cuales dejan en evidencia el contexto que se busca representar. Por último, se puede apreciar la tabla _EFMigrationsHistory, la cual lleva un historial de cambios en la base de datos, que es independiente de contexto de la solución.

8.1.1. Diccionario de datos

AspNetRoles:

| Columna | Tipo | Nulo | Predeterminado | Comentarios |
|------------------------|--------------|------|----------------|---|
| Id (<i>Primaria</i>) | varchar(255) | No | | Id del rol |
| Name | varchar(256) | No | NULL | Nombre del rol |
| NormalizedName | varchar(256) | Sí | NULL | Nombre del rol normalizado |
| ConcurrencyStamp | longtext | Sí | NULL | Sello de concurrencia para el control de las transacciones. |

AspNetUserRoles:

| Columna | Tipo | Nulo | Predeterminado | Comentarios |
|----------------------------|--------------|------|----------------|---------------------------|
| UserId (<i>Primaria</i>) | varchar(255) | No | | Id del usuario a vincular |
| RoleId (<i>Primaria</i>) | varchar(255) | No | | Id del rol a vincular |

AspNetUsers:

| Columna | Tipo | Nulo | Predeterminado | Comentarios |
|------------------------|--------------|------|----------------|-----------------------------------|
| Id (<i>Primaria</i>) | varchar(255) | No | | Identificador de usuario |
| UserName | varchar(256) | No | NULL | Nombre de la cuenta (Email) |
| NormalizedUserName | varchar(256) | No | NULL | Nombre de la cuenta en mayúsculas |
| Email | varchar(256) | Sí | NULL | Email del usuario |
| NormalizedEmail | varchar(256) | Sí | NULL | Email en mayúsculas |
| EmailConfirmed | bit(1) | Si | | |
| PasswordHash | longtext | Sí | NULL | Contraseña cifrada |
| SecurityStamp | longtext | Sí | NULL | |
| ConcurrencyStamp | longtext | Sí | NULL | |
| PhoneNumber | longtext | Sí | NULL | Numero de celular del usuario |
| PhoneNumberConfirmed | bit(1) | Sí | | |

| | | | | |
|-------------------|-------------|----|------|--|
| TwoFactorEnabled | bit(1) | No | | |
| LockoutEnd | datetime(6) | Sí | NULL | |
| LockoutEnabled | bit(1) | No | | |
| AccessFailedCount | int(11) | No | | |

DataTypes:

| Columna | Tipo | Nulo | Predeterminado | Comentarios |
|--------------------------------|-------------|------|----------------|-------------------------|
| DataTypeId (<i>Primaria</i>) | int(11) | No | | Id del tipo de dato |
| DataTypeName | varchar(20) | No | | Nombre del tipo de dato |
| DataTypeType | varchar(20) | No | | Tipo de dato |

ModuleTopics:

| Columna | Tipo | Nulo | Predeterminado | Comentarios |
|------------------------------|---------|------|----------------|--------------------------|
| ModuleId (<i>Primaria</i>) | int(11) | No | | Id del módulo a vincular |
| TopicId (<i>Primaria</i>) | int(11) | No | | Id del tópico a vincular |

Modules:

| Columna | Tipo | Nulo | Predeterminado | Comentarios |
|------------------------------|-------------|------|----------------|--|
| ModuleId (<i>Primaria</i>) | int(11) | No | | Id del módulo |
| ModuleName | varchar(50) | No | | Nombre del módulo |
| ModuleVersion | varchar(10) | No | | Versión del modulo en base al hardware |
| ModuleFirmware | varchar(10) | No | | Versión del módulo en base al firmware |

Topics:

| Columna | Tipo | Nulo | Predeterminado | Comentarios |
|-----------------------------|---------|------|----------------|---------------|
| TopicId (<i>Primaria</i>) | int(11) | No | | Id del tópico |

| | | | | |
|------------|--------------|----|------|------------------------------|
| TopicName | varchar(20) | No | | Nombre del tópico |
| TopicTopic | varchar(250) | No | | Tópico |
| DataTypeId | int(11) | Sí | NULL | Id del tipo de dato asignado |

UserModules:

| Columna | Tipo | Nulo | Predeterminado | Comentarios |
|------------------------------|--------------|------|----------------|---------------------------|
| UserId (<i>Primaria</i>) | varchar(255) | No | | Id del usuario a vincular |
| ModuleId (<i>Primaria</i>) | int(11) | No | | Id del módulo a vincular |

__EFMigrationsHistory:

| Columna | Tipo | Nulo | Predeterminado | Comentarios |
|---------------------------------|-------------|------|----------------|-------------------------|
| MigrationId (<i>Primaria</i>) | varchar(95) | No | | Id de la migración |
| ProductVersion | varchar(32) | No | | Versión de la migración |

8.2. Diseño de arquitectura funcional

En este proyecto se ha implementado el patrón de diseño modelo vista controlador, para el desarrollo del web API y de la aplicación móvil; orientación a objetos de forma simple, para el desarrollo del servidor MQTT y programación funcional, para el desarrollo del módulo domótico.

A continuación, se presentará la arquitectura del sistema, desde la perspectiva de cada cliente.

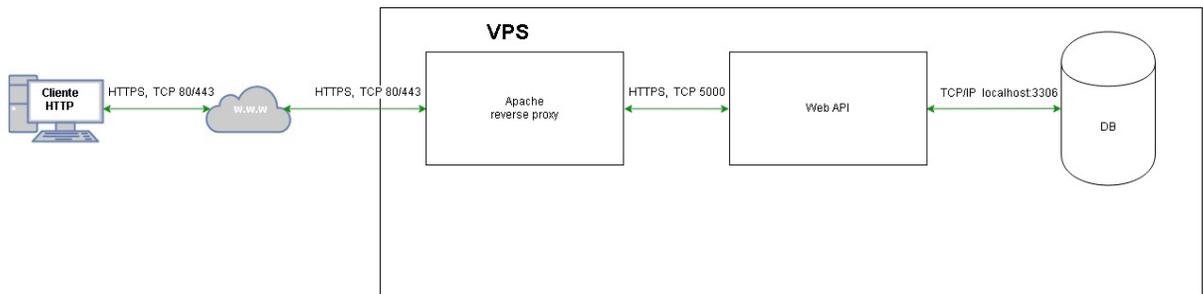


Ilustración 8-2 Arquitectura-perspectiva cliente HTTP

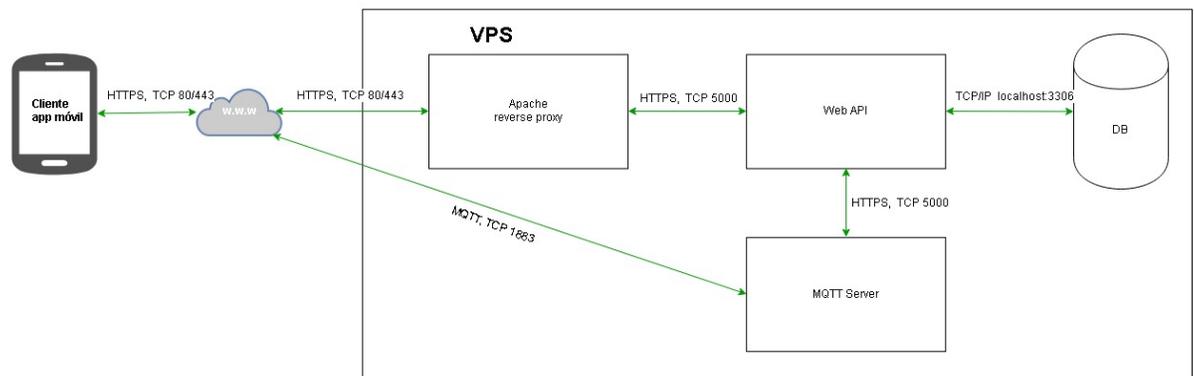


Ilustración 8-3 Arquitectura-perspectiva cliente de aplicación Android

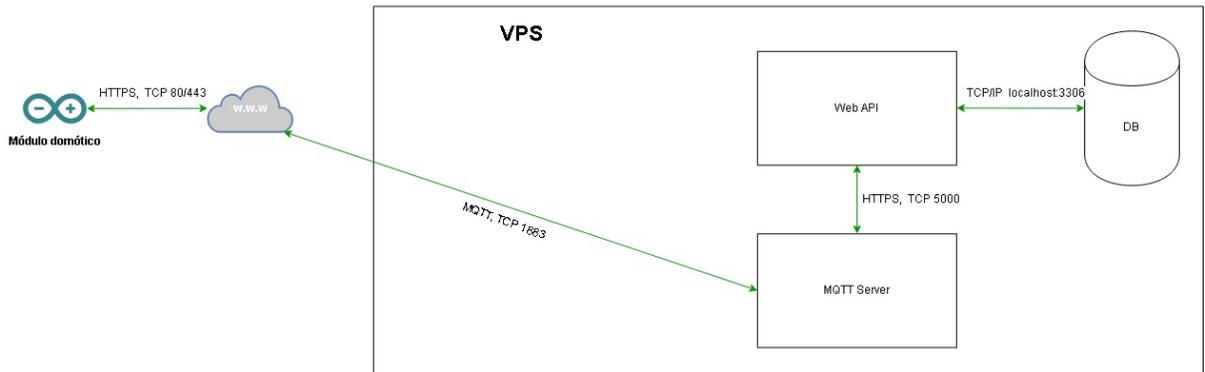


Ilustración 8-4 Arquitectura-perspectiva módulo doméstico

8.2.1. Web API

En la **ilustración 8-5** se aprecia la estructura del directorio del web API, la imagen fue obtenida desde el explorador de soluciones de visual estudio 2017. Luego de la ilustración se procederá a dar un detalle de cada directorio visible en la imagen.

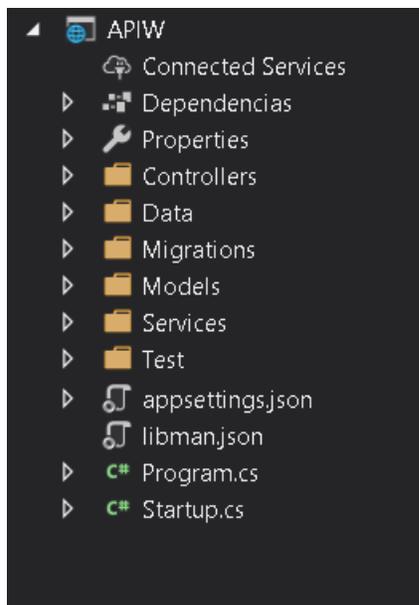


Ilustración 8-5 Estructura de directorios web API

- **APIW/Connected Services:** No corresponde a un directorio. Connected services permite publicar la aplicación ASP.NET Core en el servidor preconfigurando un el dato de acceso.
- **APIW/Dependencias:** Directorio en donde se almacenan las dependencias externas, importadas mediante el administrador de paquetes NuGet.
- **APIW/Properties:** Directorio que contiene archivos con configuraciones de lanzamiento del web API (puerto, URL, SSL, etc.).
- **APIW/Controllers/:** Directorio que contiene los controladores del web API. Los controles son llamados desde una solicitud HTTP.
- **APIW/Data/:** Directorio que contiene el contexto de la base de datos de la aplicación. El archivo contenido en este directorio es el que nos permite ejecutar las migraciones o recuperar el contexto desde la base de datos (mantiene las relaciones generadas a partir del modelo y nos permite realizar todas las acciones de un CRUD).
- **APIW/Migrations/:** Directorio que contiene las migraciones que se obtienen a partir de los modelos ya relacionados y validados (registro de cambios de la base de dato).
- **APIW/Models/:** Directorio que contiene los modelos y sus respectivas validaciones. Estos son utilizados por los controladores. También, con el uso de Entity Framework Core (ORM), los modelos se pueden relacionar con sus respectivas cardinalidades.
- **APIW/Services/:** Directorio utilizado para contener las clases e interfaces que será utilizadas mediante inyección de dependencia.
- **APIW/Test/:** Directorio que tiene como finalidad almacenar las pruebas realizadas al web API.
- **APIW/appsettings.json:** Archivo en formato JSON que sirve para establecer atributos que pueden ser llamados desde el código (en este proyecto almacena el enlace de conexión a la base de datos).

- **APIW/libman.json:** Archivo autogenerado que no se ha utilizado en esta versión del proyecto, libman.json sirve para agregar las bibliotecas que se usarán en el frontend.
- **APIW/Program.cs:** Corresponde a la clase que contiene la función principal, para la ejecución del web API.
- **APIW/Startup.cs:** Corresponde a la clase en donde se establecen los middlewares, configuraciones iniciales, y en donde se configuran las inyecciones de dependencias.

8.2.2. Servidor MQTT

La **ilustración 8-6** muestra la estructura de directorios del servicio multiplataforma (Servidor MQTT). Posterior a esta se detallará cada directorio o archivo.

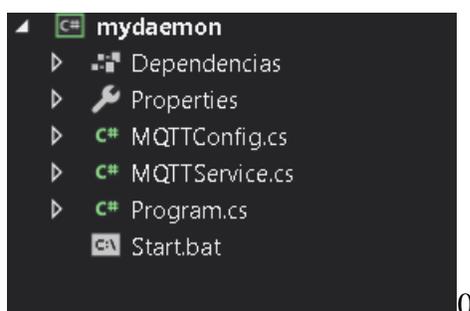


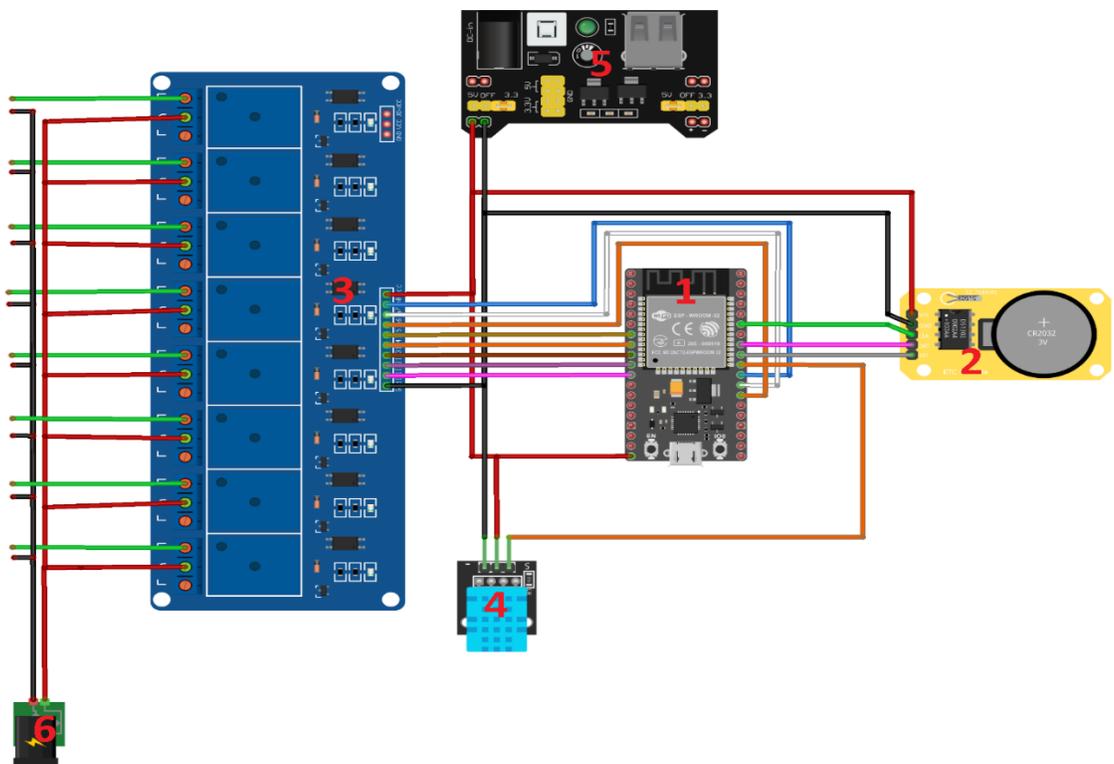
Ilustración 8-6 Estructura de directorios servidor MQTT

- **mydaemon/Dependencias:** Directorio en donde se almacenan las dependencias externas, importadas mediante el administrador de paquetes NuGet.
- **mydaemon/Properties:** Directorio que contiene archivos con configuraciones de lanzamiento del web API (puerto, URL, SSL, etc.).
- **mydaemon/MQTTConfig.cs:** Clase que contiene un único atributo llamado Port, el valor de este atributo será ingresado por parámetro durante el lanzamiento del servidor MQTT.

- **mydaemon/MQTTService.cs:** Clase que contiene las opciones de arranque del servicio MQTT y también los métodos (eventos de publicación, suscripción, conexión, desconexión y de arribo de mensajes).
- **mydaemon/Program.cs:** Corresponde a la clase que contiene la función principal, para la ejecución del servidor MQTT.
- **mydaemon/Start.bat:** Archivo que contiene el comando de ejecución del servidor MQTT.

8.2.3. Módulo domótico

El módulo domótico carece de una estructura de directorios, pues para el desarrollo de este, se ha utilizado el paradigma de programación funcional, por tanto, se procederá a describir las funcionalidades del módulo domótico a partir del circuito presentado en la **ilustración 8-7**, en el cual se destacan todos los componentes utilizados en la confección de este.



fritzing

Ilustración 8-7 Circuito del módulo domótico

1. **Módulo ESP32:** Es el núcleo del circuito y es en donde reside el binario del código desarrollado. También posee tarjeta WIFI, lo que le permitirá conectarse a Internet sobre el protocolo MQTT (para la comunicación remota con otros dispositivos).
2. **RTS1302:** Es un reloj de tiempo real, que podrá ser configurado mediante el uso de la aplicación móvil (la aplicación también recibe la zona horaria del dispositivo cada cierto periodo de tiempo) y es usado para encender o apagar los relés a una determinada hora de forma automática (el ESP32 se programó para que sincronice el RTS con un servidor NTP).
3. **Módulo de 8 relés:** Este módulo puede ser comparado a 8 interruptores. Su función en este proyecto es que sean controlados y automatizados, mediante el uso de la aplicación móvil, lo cual permitirá encender o apagar aparatos eléctricos de forma remota (cada relé soporta 250 Volts a 10 Amperios).
4. **Módulo KY-015:** El módulo porta un sensor de temperatura y humedad (DTH11). El propósito de este módulo es medir estas variables y enviarlas a la aplicación móvil.
5. **Fuente de alimentación YwRobot:** Su función es alimentar el circuito, el voltaje de entrada debe ser de 6 a 12 volts y el de salida es de 3 o 5 volts, con una corriente máxima de 700 mA.
6. **Fuente externa de alimentación:** Es utilizada para energizar los electrodomésticos, ya que el módulo domótico solo tiene el rol de abrir o cerrar el circuito mediante los relés (la fuente externa igual puede ser de corriente continua).

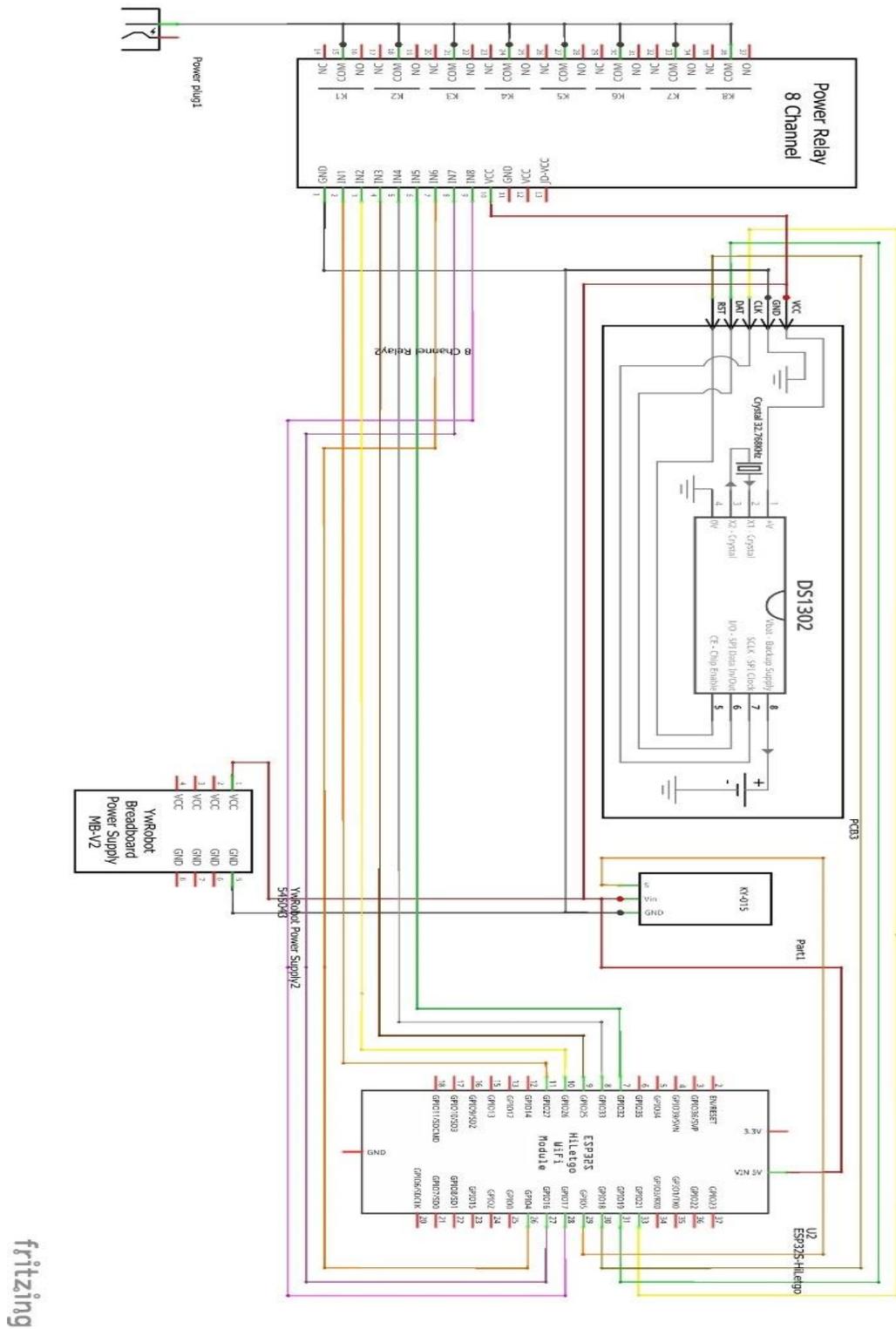


Ilustración 8-9 Circuito esquemático del módulo doméstico

fritzing

8.2.4. Aplicación móvil

La **lustración 8-10** muestra la estructura de directorios de la aplicación móvil, para control del módulo domótico. Luego, se detallará cada directorio o archivo existente.

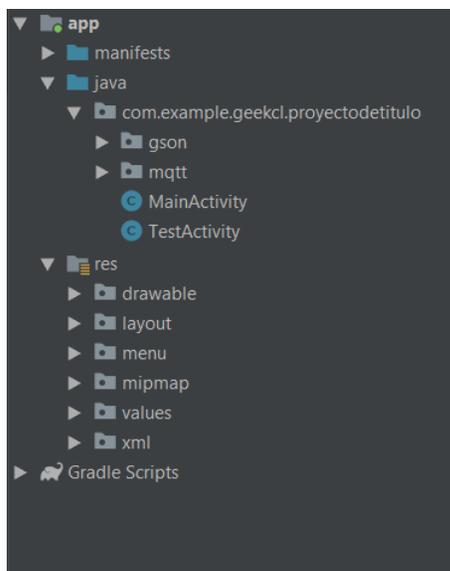


Ilustración 8-10 Estructura de directorios App Android

- **app/manifests/:** En este directorio se encuentra el archivo AndroidManifest.xml, el cual proporciona información esencial sobre la aplicación al sistema Android, información que el sistema debe tener para poder ejecutar el código de la app.
- **app/java/com.example.geekcl.proyectodetitulo/:** Directorio que contiene el código de las actividades de la aplicación Android (una actividad es un componente de la aplicación que contiene una pantalla con la que los usuarios pueden interactuar para realizar una acción).
- **app/java/com.example.geekcl.proyectodetitulo/gson/:** Directorio que contiene los modelos que serán utilizados para serializar los mensajes enviados al módulo domótico y deserializar los enviados por este.

- **app/java/com.example.geekcl.proyectodetitulo/mqtt/:** Directorio contiene el código que permite la comunicación sobre MQTT.
- **app/res/drawable/:** Directorio que almacena recursos que luego puede ser invocado (contiene las imágenes utilizadas en el desarrollo de la aplicación).
- **app/res/layout/:** Directorio donde se almacenan las vistas de la aplicación Android.
- **app/res/menu/:** Directorio que contiene el menú de la barra de navegación.
- **app/res/anim/:** Directorio que contiene el icono de la aplicación móvil, en diferentes dimensiones(tamaños).
- **app/res/values/:** Directorio que contiene recurso utilizados por la aplicación Android (colores, dimensiones, dibujables, cadenas de texto, etc.).
- **app/res/xml/:** En este directorio se encuentra el archivo backup_descriptor.xml, el cual permite excluir elementos de las copias de seguridad.
- **Gradle Scripts/:** Directorio que contiene archivos para configurar dependencias externas.

8.3. Diseño de interfaz y navegación

En la ilustración 8-11 se presenta la maqueta de la aplicación móvil, la cual ha sido dividida en cuatro vistas(actividades) que serán detalladas a partir de la numeración establecida en la imagen.

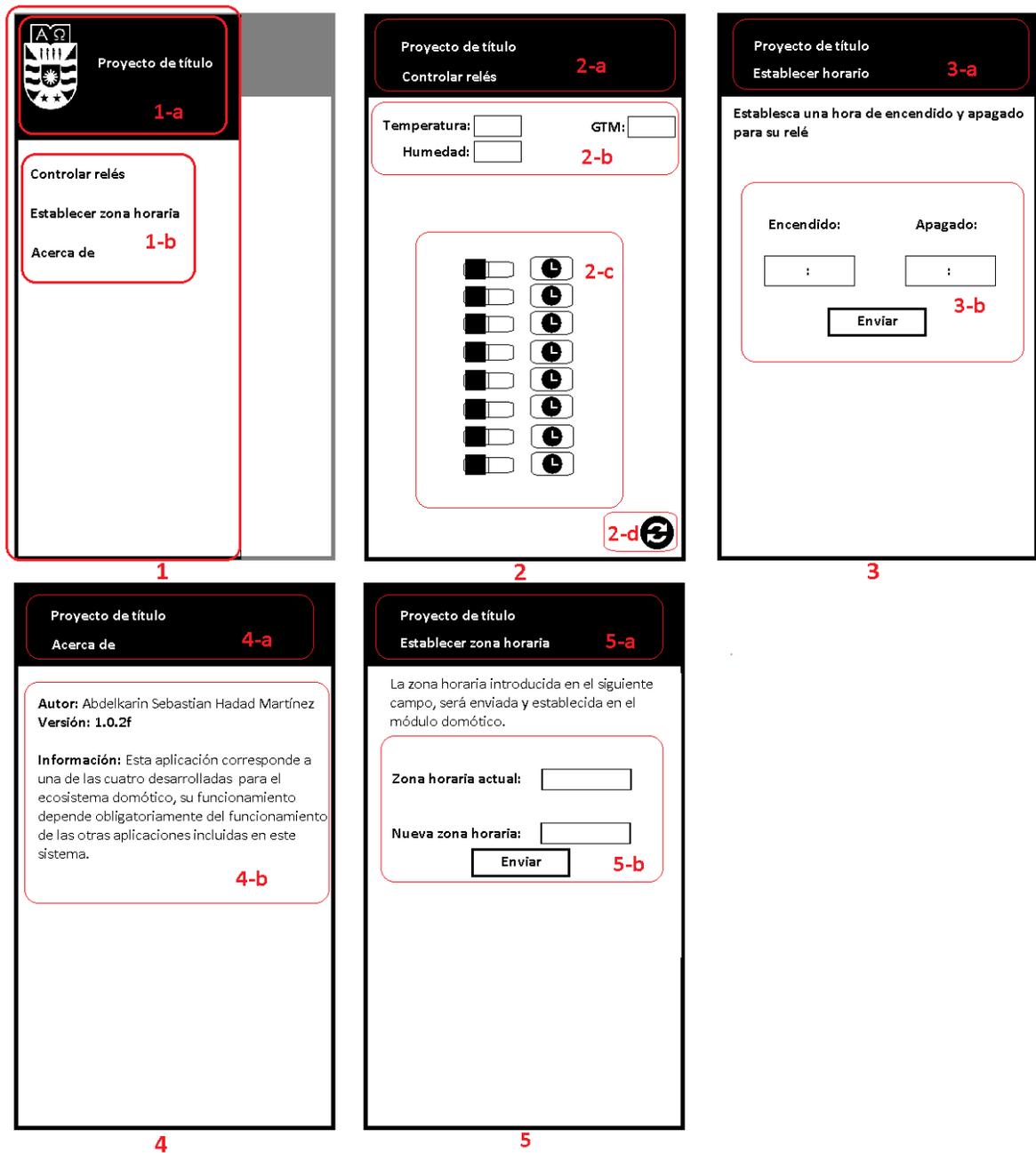


Ilustración 8-11 Diseño de interfaz y navegación

1. **Barra de navegación:** Corresponde al menú de navegación y estará presente en todas las actividades.
 - a. **Cabecera:** Presenta el logo de la Universidad del Bío Bío y el título del proyecto.
 - b. **Opciones de menú:** Contiene los enlaces a las distintas vistas de la aplicación (controlar relés, establecer zona horaria y acerca de)
2. **Actividad controlar relés:** permite visualizar los datos enviados por el módulo domótico y también permite el control de este.
 - a. **Barra de herramientas:** Muestra el título del proyecto y título de la actividad.
 - b. **Datos entrantes:** Corresponde a los datos que envía el módulo domótico cada cierto periodo de tiempo (temperatura, humedad y zona horaria).
 - c. **Control:** Corresponde a la zona de control de los relés los componentes del lado izquierdo corresponden a los interruptores de encendido y apagado y los componentes del lado derecho(relojes) abren un fragmento(panel) que permitirá establecer un encendido y apagado, de forma automática, para cada relé. Al presionar un reloj abre un fragmento que corresponde a la vista 3 de la **ilustración 8-11** (cada reloj abre un fragmento distinto para automatizar el relé que le corresponde, el orden de los switches y los relés son de 0 a 7).
3. **fragmento establecer encendido y apagado:** Como se ha mencionado en el punto 2-c esta vista corresponde a un fragmento que se abre al presionar un reloj de la actividad para controlar relés. Cada reloj establece el horario a su relé correspondiente.
 - a. **Barra de herramientas:** Muestra el título del proyecto y título de la actividad.
 - b. **Enviar horarios:** En esta área se debe ingresar un horario de encendido y otro de apagado, estos serán enviados al módulo domótico, el procederá a automatizar dicho relé.

4. **Actividad acerca de:** En esta actividad se muestra el nombre del autor, versión de la aplicación y un pequeño mensaje informativo.
5. **Actividad establecer zona horaria:** Esta actividad permite visualizar la zona horaria que reside en el módulo domótico y también permite cambiarla.
 - a. **Barra de herramientas:** Muestra el título del proyecto y título de la actividad.
 - b. **Enviar zona horaria:** Área que muestra la zona horaria establecida actualmente y también que muestra el campo para ingresar un nuevo valor GTM. Al presionar enviar el módulo domótico cambiará su hora interna.

CAPÍTULO 9: PRUEBAS

Para comprobar la integridad y el correcto funcionamiento del sistema, se le hará una serie de pruebas siguiendo una adaptación basada en IEEE Software Test Documentation Std 829-1998 y como resultado se obtendrá información objetiva e independiente sobre la calidad del producto.

9.1. Elementos de prueba

A continuación, se definen los elementos de prueba para el ecosistema domótico, que se consideran indispensables para su funcionamiento.

Elementos de prueba para el web API:

- **Ingresar al sistema:** se comprobará que solo puedan acceder usuario registrados, que cuenten con un token de acceso.
- **Maestro usuarios:** Se comprobará el correcto funcionamiento de la operación para crear usuarios.

Elementos de prueba para el módulo domótico:

- **Interactuar con los relés:** Se comprobará el correcto encendido o apagado de los relés.
- **Automatizar relés:** Se comprobará el correcto funcionamiento de la automatización de los relés.
- **responder temperatura y humedad:** El módulo publicara la temperatura y humedad en un tópico del servidor MQTT.
- **recibir zona horaria:** El módulo domótico recibirá una nueva zona horaria y la establecerá.
- **Responder estado de los componentes:** El ESP32 responderá el estado de todos los componentes que este posee (configuraciones también).

Elementos de prueba para el servidor MQTT:

- **Publicación de mensajes en tópicos:** Se verificará que el servidor MQTT permita realizar publicaciones a los clientes.

- **Suscripción a tópicos:** Se comprobará que el servidor MQTT que los clientes sean capaces de suscribirse a tópicos.

9.2. Especificación de las pruebas

Se ha optado por realizar **pruebas de unidades de caja negra**, estas se centran en probar la funcionalidad de los componentes o unidades individuales de la aplicación para garantizar que se comporten correctamente y satisfagan una especificación definida o un caso de uso. Tener casos de prueba individuales para cada unidad de una base de código permite una rápida identificación, aislamiento y resolución de errores. Las pruebas unitarias permiten identificar y resolver los problemas de funcionalidad tan pronto como se presentan, lo que es más sencillo y menos costoso que dejarlos hasta el final del ciclo de desarrollo.

La técnica seleccionada para las pruebas, es la de **particiones equivalentes**, esta consta en dividir el dominio de entrada de un programa en clases de datos de las que se pueden derivar casos de prueba. El diseño de casos de prueba para la partición equivalente se basa en una evaluación de las clases de equivalencia para una condición de entrada. Una clase de equivalencia representa un conjunto de estados válidos o inválidos para las condiciones de entrada.

| Id | Nombre del requerimiento funcional | Salida esperada | Salida obtenida | Resultados de las pruebas | |
|----|--|---------------------|---------------------|---------------------------|-----------------------------|
| | | | | Éxito/Fracaso | Características del fracaso |
| 01 | Crear usuario | CreateUser_OK | CreateUser_OK | Éxito | |
| 02 | Ingresar al sistema | LoginUser_OK | LoginUser_OK | Éxito | |
| 03 | Interactuar con relés | Relay_ON/Relay_OF | Relay_ON/Relay_OF | Éxito | |
| 04 | Automatizar relés | Relay_ON/Relay_OF | Relay_ON/Relay_OF | Éxito | |
| 05 | Establecer zona horaria | TimeZone_OK | TimeZone_OK | Éxito | |
| 06 | Responder el estado de los componentes | TimeZoneResponse_OK | TimeZoneResponse_OK | Éxito | |

| | |
|---------------------------------------|--|
| <p>Interactuar con relés</p> | <ul style="list-style-type: none"> • Relé n°1 on: <ul style="list-style-type: none"> ○ Acción: relay ○ Id: 0 ○ Estado: true • Relé n°1 off: <ul style="list-style-type: none"> ○ Acción: relay ○ Id: 0 ○ Estado: false • Relé n°2 on: <ul style="list-style-type: none"> ○ Acción: Relay ○ Id: 1 ○ Estado: true • Relé n°2 off: <ul style="list-style-type: none"> ○ Acción: Relay ○ Id: 1 ○ Estado: false |
| <p>Automatizar relés</p> | <ul style="list-style-type: none"> • Automatizar relé n°1: <ul style="list-style-type: none"> ○ Acción: organizador ○ Id: 0 ○ Inicio:1314 ○ Fin:1344 • Automatizar relé n°2: <ul style="list-style-type: none"> ○ Acción: organizador ○ Id: 0 ○ Inicio: 1550 ○ Fin: 1540 |
| <p>Establecer zona horaria</p> | <ul style="list-style-type: none"> • Zona horaria n°1: <ul style="list-style-type: none"> ○ Acción: gtm ○ GTM: -3 • Zona horaria n°2: <ul style="list-style-type: none"> ○ Acción: gtm ○ GTM: -4 |

| | |
|---|---|
| Responder el estado de los componentes | <ul style="list-style-type: none"> • Notificación: <ul style="list-style-type: none"> ○ Acción: notificación |
| Suscripción a tópicos | <ul style="list-style-type: none"> • Tópico: TópicoTEST |
| Publicación de mensajes en tópicos | <ul style="list-style-type: none"> • Mensaje a publicar: Este es un mensaje publicado en el tópico de pruebas |

Tabla 9-2 Detalle de las pruebas - Particiones equivalentes

9.3. Responsables de las pruebas

- Abdelkarin Sebastián Hadad Martínez.

9.4. Conclusiones de Prueba

Teniendo en consideración que para el desarrollo del web API se hizo uso del ORM Entity Framework Core (de ASP.NET Core), la idea de que un valor de entrada rompa las reglas del modelo se va dejando de lado, por tanto, solo se evaluó el módulo de usuarios y se prosiguió con las pruebas para el módulo domótico, para este propósito se utilizó un cliente MQTT con el cual se le envió una serie mensajes (**ver anexo 1**), de los cuales se esperaba una respuesta o acción. El resultado de dichas pruebas fue positivo dado a que todas respondieron de la manera que se esperaba.

CONCLUSIONES

Realizar un proyecto de software está lleno de complejidades que pueden ser superadas con la lectura de una buena documentación. Al haber desarrollado este proyecto pasé de saberlo todo a no saber nada. El mundo del Internet de las Cosas es muy amplio y está lleno de tecnologías de las cuales se conoce muy poco. Para establecer un punto fijo en este mar de múltiples tecnologías y prácticas, se seleccionó el módulo ESP32 y el protocolo MQTT, que es lo más común en el mundo del IoT y abunda su documentación, sin embargo, hay que dedicar muchas horas de lectura para desarrollar un buen módulo.

Inicialmente el proyecto no contemplaba el desarrollo de un servidor MQTT propio, pero las circunstancias se dieron y se desarrolló un proceso demonio multiplataforma que tiene ese rol.

El conocimiento que se ha logrado con el presente proyecto dio como resultado un módulo que puede ser controlado remotamente, mediante una aplicación móvil. Además, sincroniza un reloj de tiempo real con un servidor NTP (servidor horario) y como resultado el módulo puede automatizar múltiples relés.

En síntesis, el proyecto cumple el objetivo de crear un dispositivo domótico de bajo costo con control remoto, lo cual es un buen plus para el desarrollo de futuros proyectos IoT, debido a que la domótica y el Internet de las Cosas son áreas que están tomando un peso significativo en las áreas de Informática, Arquitectura, Electrónica y Telecomunicaciones.

Este proyecto servirá como base a aquellos informáticos que quieran mezclar la programación con la electrónica y otras nuevas tecnologías. Solo hace falta que pongan en marcha su imaginación y elaboren un nuevo módulo que pueda ser integrado a este sistema, o incluso podrán modificarlo completamente si así gustan, aun así, nunca olviden la fuente.

BIBLIOGRAFÍA

- Yeeply. (2013). Usabilidad en aplicaciones móviles. Madrid, España: Yeeply. Recuperado de <https://www.yeeply.com>
- Flabio Beltrán, Carla Aguirre. (2011). Programación modular. La Paz, Bolivia: n/a.
- Isaías Bonilla, Arturo Tavizon, Melisa Morales. (2016). IoT, El Internet de las cosas y la innovación de sus aplicaciones. San Nicolás de los Garza, N.L., México: n/a.
- Jordi Salazar y Santiago Silvestre. (2016). Internet de las cosas. n/a: European Virtual Learning Platform for Electrical and Information Engineering.
- Ashton, K (2009) That “Internet of Things” Thing: In the Real World Things Matter More than Ideas. RFID Journal. Recuperado de <http://www.rfidjournal.com/articles/view?4986>

ANEXO 1: FORMATOS DE COMUNICACIÓN APP – MÓDULO DOMÓTICO

Los mensajes enviados desde la aplicación móvil al módulo domótico pueden ser enviados desde un cliente MQTT haciendo uso de una cuenta de usuario registrada en el sistema. Cada mensaje es enviado bajo una acción que es reconocida por el módulo domótico.

1.1. Establecer zona horaria (aplicación móvil a módulo)

Mensaje enviado por la aplicación móvil para establecer una nueva zona horaria en el módulo domótico. El nuevo valor GTM debe ser enviado en el campo gtm.

```
{
  "action": "gtm",
  "gtm": -3
}
```

1.2. Encender o apagar relés (aplicación móvil a módulo)

Mensaje enviado para controlar los relés del módulo domótico. **El array control** contiene objetos con los campos id (identificador del relé) y estatus (estado para encender o apagar el relé).

```
{
  "action": "relay",
  "control": [
    {"id": 0, "status": true},
    {"id": 1, "status": true},
    {"id": 2, "status": true},
    {"id": 3, "status": true},
    {"id": 4, "status": true},
    {"id": 5, "status": true},
    {"id": 6, "status": true},
    {"id": 7, "status": true}
  ]
}
```

1.3. Automatizar encendido o apagado de los relés (aplicación móvil a módulo)

Mensaje que tiene como propósito automatizar los relés del módulo domótico. **El array relay** contiene objetos con los campos relay, para identificar el relé; inicio, para establecer un horario de encendido y fin, para establecer un horario de apagado.

```

{
  "action": "organizador",
  "relay": [
    {"relay": 0, "inicio": 1830, "fin": 1831},
    {"relay": 1, "inicio": 1930, "fin": 1931},
    {"relay": 2, "inicio": 2030, "fin": 2031},
    {"relay": 3, "inicio": 2130, "fin": 2131},
    {"relay": 4, "inicio": 2230, "fin": 2231},
    {"relay": 5, "inicio": 1730, "fin": 1731},
    {"relay": 6, "inicio": 1630, "fin": 1631},
    {"relay": 7, "inicio": 2330, "fin": 2331}
  ]
}

```

1.4. Petición de sincronización (aplicación móvil a módulo)

Mensaje que tiene como objetivo realizar una petición del estado de los componentes del módulo domótico.

```

{
  "action": "notificacion"
}

```

1.5. Eliminar horarios de los relés (aplicación móvil a módulo)

Mensaje que tiene como propósito realizar un borrado lógico de los horarios de encendido y apagado de los relés. **El array relay** contiene objetos con el campo relay, este especifica el id del módulo al que se le quitara el horario.

```

{
  "action": "desorganizador",
  "relay": [
    {"relay": 0},
    {"relay": 1},
    {"relay": 2},
    {"relay": 3},
    {"relay": 4},
    {"relay": 5},
    {"relay": 6},
    {"relay": 7}
  ]
}

```

1.6. Enviar zona horaria (módulo a aplicación móvil)

Mensaje enviado por el módulo domótico, que tiene como propósito notificar a los usuarios de la aplicación móvil el valor de la zona horaria establecida (zona horaria por defecto GTM-3).

```
{
  "gtm": -3
}
```

1.7. Enviar variables de ambiente (módulo a aplicación móvil)

Mensaje que es enviado cada un minuto desde el módulo domótico a la aplicación móvil y notifica el ambiente del lugar en donde se encuentre (temperatura y humedad).

```
{
  "sensor": {
    "temperature": 21.0,
    "humidity": 77.9
  }
}
```

1.8. Enviar estado de los componentes

Mensaje enviado por el módulo domótico y corresponde a una respuesta de sincronización. El mensaje contiene la zona horaria, la humedad, la temperatura, el estado de los relés (encendido o apagado), y los horarios establecidos para el encendido y apagado de los relés (los horarios son enviadas como número entero; por ejemplo: 18:40 sería 1840).

```
{
  "gtm": -3,
  "sensors": {
    "humidity": 71,
    "temperature": 21
  },
  "relesActivos": [
    0,
    0,
    0,
    1,
    0,
    0,
    0,
    0
  ],
  1,
}
```

```
"horarios": [  
  {  
    "relay": 0,  
    "inicio": 1832,  
    "fin": 1833  
  },  
  {  
    "relay": 1,  
    "inicio": 1835,  
    "fin": 1836  
  },  
  {  
    "relay": 2,  
    "inicio": 1840,  
    "fin": 1841  
  },  
  {  
    "relay": 3,  
    "inicio": 1850,  
    "fin": 1851  
  }  
]  
}
```

ANEXO 2: MANUAL DE USO PARA ADMINISTRADOR – WEB API

A continuación, se presentará el manual de uso para el web API. La estructura del web API está diseñada para que sea navegable, por tanto, si en la URL aparece una entidad seguida de un dígito o código, este corresponderá al id de dicha entidad; por ejemplo: `https://IP:5000/api/modules/1/topics/50`, el id del módulo sería 1 y el id del tópico sería 50.

Como se mencionó con anterioridad, para interactuar con el API REST se necesita un cliente HTTP y un token de acceso (la cuenta administradora por defecto se aprecia en el punto 13.1). Se recomienda el uso de Postman, para así importar la una colección que viene configurada para interactuar con el servidor. La colección es la siguiente:

- <https://www.getpostman.com/collections/6eae84f80278c98164d5>

2.1. Ingresar al sistema (iniciar sesión)

URL: `https://IP:5000/api/account/login`

Método: **POST**

Cuerpo de la solicitud: `raw application/json`

```
{
  "Email": "AbdelkarinH@gmail.com",
  "Password": "Kteimporta8#"
}
```

Respuesta del servidor:

```
{
  "Id": "21a2eb58-c0ac-4a5c-85c0-761696f5fe48",
  "UserName": "AbdelkarinH@gmail.com",
  "Email": "AbdelkarinH@gmail.com",
  "AccessToken":
    "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1bmR5IjoiAbdelkarinH@gmail.com",
    "Expiration": "2019-03-17T06:49:06.3720504Z"
}
```

2.2. Módulo - Usuarios

2.2.1. Crear usuario

URL: <https://IP:5000/api/users/>

Método: **POST**

Cuerpo de la solicitud: raw application/json

```
{
  "Email": "AbdelkarinH@gmail.com",
  "Password": "Kteimporta8#"
  "RoleId": [1,2] //1-> Administrador 2->Usuario normal
}
```

Respuesta del servidor:

```
{
  "Id": "21a2eb58-c0ac-4a5c-85c0-761696f5fe48",
  "UserName": "AbdelkarinH@gmail.com",
  "Email": "AbdelkarinH@gmail.com",
  "NormalizedUserName": "ABDELKARINH@GMAIL.COM"
}
```

2.2.2. Modificar usuario

URL: <https://IP:5000/api/users/21a2eb58-c0ac-4a5c-85c0-761696f5fe48>

Método: **PUT**

Cuerpo de la solicitud: raw application/json

```
{
  "Id": "21a2eb58-c0ac-4a5c-85c0-761696f5fe48",
  "UserName": "AbdelkarinH@gmail.com",
  "Email": "Micorreo2@gmail.com"
}
```

Respuesta del servidor:

```
{
  "Id": "21a2eb58-c0ac-4a5c-85c0-761696f5fe48",
  "UserName": "AbdelkarinH@gmail.com",
  "Email": "Micorreo2@gmail.com",
}
```

2.2.3. Buscar usuario

URL: <https://IP:5000/api/users/1>

Método: **GET**

Respuesta del servidor:

```
{
  "Id": "21a2eb58-c0ac-4a5c-85c0-761696f5fe48",
  "UserName": "AbdelkarinH@gmail.com",
  "Email": "Micorreo2@gmail.com",
}
```

2.2.4. Listar usuarios

URL: <https://IP:5000/api/users/>

Método: GET

Respuesta del servidor:

```
[
  {
    "Id": "21a2eb58-c0ac-4a5c-85c0-761696f5fe48",
    "UserName": "AbdelkarinH@gmail.com",
    "Email": "Micorreo2@gmail.com",
  },
  {Usuario 2}, {Usuario 3}, ..., {Usuario n}
]
```

2.2.5. Eliminar usuario

URL: <https://IP:5000/api/users/1/>

Método: DELETE

Respuesta del servidor:

```
{
  "Id": "21a2eb58-c0ac-4a5c-85c0-761696f5fe48",
  "UserName": "AbdelkarinH@gmail.com",
  "Email": "Micorreo2@gmail.com",
  "NormalizedUserName": "MICORREO2@GMAIL.COM"
}
```

2.3. Módulo - Dispositivos domóticos

2.3.1. Crear dispositivo domótico

URL: <https://IP:5000/api/modules/>

Método: POST

Cuerpo de la solicitud: raw application/json

```
{
  "ModuleName": "dth11/",
  "ModuleVersion": "v1.0",
  "ModuleFirmware": "v1.0"
}
```

Respuesta del servidor:

```
{
  "ModuleId": 1,
  "ModuleName": "dth11/8relay",
  "ModuleVersion": "v1.0",
  "ModuleFirmware": "v1.0"
}
```

2.3.2. Modificar dispositivo domótico

URL: <https://IP:5000/api/modules/1>

Método: PUT

Cuerpo de la solicitud: raw application/json

```
{
  "ModuleId": 1,
  "ModuleName": "dth11/8relay",
  "ModuleVersion": "v1.0",
  "ModuleFirmware": "v1.0"
}
```

Respuesta del servidor:

```
{
  "ModuleId": 1,
  "ModuleName": "dth11/8relay",
  "ModuleVersion": "v1.0",
  "ModuleFirmware": "v1.0"
}
```

2.3.3. Buscar dispositivo domótico

URL: <https://IP:5000/api/modules/1>

Método: GET

Respuesta del servidor:

```
{
  "ModuleId": 1,
  "ModuleName": "dth11/8relay",
  "ModuleVersion": "v1.0",
  "ModuleFirmware": "v1.0"
}
```

2.3.4. Listar dispositivos domóticos

URL: **https:// IP:5000/api/modules/**

Método: **GET**

Respuesta del servidor:

```
[
  {
    "ModuleId": 1,
    "ModuleName": "dth11/8relay",
    "ModuleVersion": "v1.0",
    "ModuleFirmware": "v1.0"
  },
  {Módulo 2}, {Módulo 3}, ..., {Módulo n}
]
```

2.3.4. Eliminar dispositivo domótico

URL: **https://IP:5000/api/modules/1/**

Método: **DELETE**

Respuesta del servidor:

```
{
  "ModuleId": 1,
  "ModuleName": "dth11/8relay",
  "ModuleVersion": "v1.0",
  "ModuleFirmware": "v1.0"
}
```

2.4. Módulo - Tipos de Datos

2.4.1. Crear tipo de dato

URL: **https://IP:5000/api/datatypes/**

Método: **POST**

Cuerpo de la solicitud: raw application/json

```
{
  "DataTypeName": "flotante",
  "DataTypeType": "float"
}
```

Respuesta del servidor:

```
{
```

```
"DataTypeId": 1,  
"DataTypeName": "flotante",  
"DataTypeType": "float"  
}
```

2.4.2. Modificar tipo de dato

URL: <https://IP:5000/api/datatypes/1>

Método: **PUT**

Cuerpo de la solicitud: raw application/json

```
{  
  "DataTypeId": 1,  
  "DataTypeName": "Número flotante",  
  "DataTypeType": "float"  
}
```

Respuesta del servidor:

```
{  
  "DataTypeId": 1,  
  "DataTypeName": "Número flotante",  
  "DataTypeType": "float"  
}
```

2.4.3. Buscar tipo de dato

URL: <https://IP:5000/api/datatypes/1>

Método: **GET**

Respuesta del servidor:

```
{  
  "DataTypeId": 1,  
  "DataTypeName": "Número flotante",  
  "DataTypeType": "float"  
}
```

2.4.4. Listar tipo de dato

URL: <https://IP:5000/api/datatypes/>

Método: **GET**

Respuesta del servidor:

```
[
  {
    "DataTypeId": 1,
    "DataTypeName": "Número flotante",
    "DataTypeType": "float"
  },
  {Tipo de dato 2}, {Tipo de dato 3}, ..., { Tipo de dato n}
]
```

2.4.5. Eliminar tipo de dato

URL: <https://IP:5000/api/datatypes/1/>

Método: DELETE

Respuesta del servidor:

```
{
  "DataTypeId": 1,
  "DataTypeName": "Número flotante",
  "DataTypeType": "float"
}
```

2.5. Módulo - Tópicos

2.5.1. Crear tópico

URL: <https://IP:5000/api/topics/>

Método: POST

Cuerpo de la solicitud: raw application/json

```
{
  "TopicName": "Temperatura y humedad",
  "TopicTopic": "/temperaturehumidity/",
  "DataTypeId": 1
}
```

Respuesta del servidor:

```
{
  "TopicId": 1,
  "TopicName": "Temperatura y humedad",
  "TopicTopic": "/temperaturehumidity/",
  "DataTypeId": 1
}
```

2.5.2. Modificar tópico

URL: <https://IP:5000/api/topics/1>

Método: PUT

Cuerpo de la solicitud: raw application/json

```
{
  "TopicId": 1,
  "TopicName": "Temperatura y humedad",
  "TopicTopic": "/temperature.humidity/",
  "DataTypeId": 1
}
```

Respuesta del servidor:

```
{
  "TópicoId": 1,
  "TópicoName": "Temperatura y humedad",
  "TópicoTopic": "/temperature.humidity/",
  "DataTypeId": 1
}
```

2.5.3. Buscar tópico

URL: <https://IP:5000/api/topics/1>

Método: GET

Respuesta del servidor:

```
{
  "TopicId": 1,
  "TopicName": "Temperatura y humedad",
  "TopicTopic": "/temperature.humidity/",
  "DataTypeId": 1
}
```

2.5.4. Listar tópicos

URL: <https://IP:5000/api/topics/>

Método: GET

Respuesta del servidor:

```
[
  {
    "TopicId": 1,
    "TopicName": "Temperatura y humedad",
    "TopicTopic": "/temperature.humidity/",
    "DataTypeId": 1
  },
  {Tópico 2}, {Tópico 3}, ..., { Tópico n}
]
```

2.5.5. Eliminar tópico

URL: <https://IP:5000/api/topics/1/>

Método: DELETE

Respuesta del servidor:

```
{
  "TopicId": 1,
  "TopicName": "Temperatura y humedad",
  "TopicTopic": "/temperature.humidity/",
  "DataTypeId": 1
}
```

2.6. Asignar módulos a usuarios (módulos o dispositivos domóticos)

2.6.1. Asignar módulo a usuario

URL: <https://IP:5000/api/users/21a2eb58-c0ac-4a5c-85c0-761696f5fe48/modules>

Método: POST

Cuerpo de la solicitud: raw application/json

```
{
  "UserId": "21a2eb58-c0ac-4a5c-85c0-761696f5fe48",
  "ModuleId": 1
}
```

Respuesta del servidor: 200OK "Módulo asignado correctamente"

2.6.2. Modificar módulo asignado a usuario

URL: <https://IP:5000/api/users/21a2eb58-c0ac-4a5c-85c0-761696f5fe48/modules/1>

Método: PUT

Cuerpo de la solicitud: raw application/json

```
{
  "UserId": "21a2eb58-c0ac-4a5c-85c0-761696f5fe48",
  "ModuleId": 2
}
```

Respuesta del servidor: 200OK "Módulo modificado correctamente"

2.6.3. Listar módulos asignados a un usuario

URL: <https://IP:5000/api/users/21a2eb58-c0ac-4a5c-85c0-761696f5fe48/modules/>

Método: GET

Respuesta del servidor:

```
[
  {
    "ModuleId": 1,
    "ModuleName": "dth11/8relay",
    "ModuleVersion": "v1.0",
    "ModuleFirmware": "v1.0"
  }
  {Módulo 2}, {Módulo 3}, ..., { Módulo n}
]
```

2.6.4. Eliminar módulo asignado a usuario

URL: <https://IP:5000/api/users/21a2eb58-c0ac-4a5c-85c0-761696f5fe48/modules/1>

Método: DELETE

Respuesta del servidor: 200OK "Módulo eliminado correctamente"

2.7. Asignar tópicos a módulos

2.7.1. Asignar tópico a módulo

URL: <https://IP:5000/api/modules/1/topics>

Método: POST

Cuerpo de la solicitud: raw application/json

```
{
  "ModuleId": 1,
  "TopicId": 1
}
```

Respuesta del servidor: 200OK "Tópico asignado correctamente"

2.7.2. Modificar tópico asignado a módulo

URL: <https://IP:5000/api/modules/1/topics/1>

Método: PUT

Cuerpo de la solicitud: raw application/json

```
{
  "ModuleId": 1,
  "TopicId": 1
}
```

Respuesta del servidor: 200OK "Tópico modificado correctamente"

2.7.3. Listar tópicos asignados a un módulo

URL: <https://IP:5000/api/modules/1/topics/>

Método: GET

Respuesta del servidor:

```
[
  {
    "TopicId": 1,
    "TopicName": "Temperatura humedad",
    "TopicTopic": "/temperaturehumidity/",
    "DataTypeId": 1
  }
  {Tópico 2}, {Tópico 3}, ..., { Tópico n}
]
```

2.7.4. Eliminar tópico asignado a módulo

URL: <https://IP:5000/api/modules/1/topics/1>

Método: DELETE

Respuesta del servidor: 200OK "Tópico eliminado correctamente"

ANEXO 3: MATRIZ DE TRAZABILIDAD DE R.F. Y CASOS DE USO

En la tabla A-1 se vinculan los requerimientos funcionales con su respectivo caso de uso, el punto de aparición de los casos de uso corresponde al punto específico de aparición de este, dentro del capítulo siete. Solo se representarán los requerimientos funcionales en los que intervenga directamente un actor.

| | Id | Nombre del requerimiento | Punto de aparición en el documento (CU) | Nombre del caso de uso |
|-----------------------------|--------------|--------------------------|--|-----------------------------|
| Servidor (Administrador) | RF_01 | Iniciar sesión | 7.1.3.2 | Iniciar Sesión |
| | RF_02 | Salir del sistema | 7.1.3.5 | salir |
| | RF_06 | Crear usuario | 7.1.3.25 | Crear usuario |
| | | | 7.1.3.29 | Crear usuario normal |
| | | | 7.1.3.30 | Crear usuario administrador |
| | RF_07 | Modificar usuario | 7.1.3.26 | Actualizar usuario |
| | RF_08 | Eliminar Usuario | 7.1.3.27 | Eliminar usuario |
| | RF_09 | Listar usuario | 7.1.3.31 | Listar usuarios |
| | RF_10 | Buscar usuario | 7.1.3.28 | Buscar usuario |
| | RF_11 | Crear módulo | 7.1.3.33 | Crear módulo |
| | RF_12 | Modificar módulo | 7.1.3.34 | Actualizar módulo |
| | RF_13 | Eliminar módulo | 7.1.3.35 | Eliminar módulo |
| | RF_14 | Listar módulo | 7.1.3.37 | Listar módulos |
| | RF_15 | Buscar módulo | 7.1.3.36 | Buscar módulo |
| | RF_16 | Crear tipo de dato | 7.1.3.46 | Crear tipo de dato |

| | | | |
|--------------|--|------------------|---|
| RF_17 | Modificar tipo de dato | 7.1.3.47 | Actualizar tipo de dato |
| RF_18 | Eliminar tipo de dato | 7.1.3.48 | Eliminar tipo de dato |
| RF_19 | Listar tipo de dato | 7.1.3.50 | Listar tipos de datos |
| RF_20 | Buscar tipo de dato | 7.1.3.49 | Buscar tipo de dato |
| RF_21 | Crear tópicos | 7.1.3.39 | Crear tópico |
| RF_22 | Modificar tópicos | 7.1.3.41 | Actualizar tópico |
| RF_23 | Eliminar tópicos | 7.1.3.42 | Eliminar tópico |
| RF_24 | Listar tópicos | 7.1.3.44 | Listar tópicos |
| RF_25 | Buscar tópicos | 7.1.3. 43 | Buscar tópico |
| RF_26 | Asignar módulos a usuarios | 7.1.3.52 | Asignar módulo a usuario |
| RF_27 | Modificar módulos asignados a usuarios | 7.1.3.53 | Actualizar asignación de módulo a usuario |
| RF_28 | Eliminar módulos asignados a usuarios | 7.1.3.54 | Eliminar asignación de módulo a usuario |
| RF_29 | Listar módulos asignados a usuario | 7.1.3.55 | Listar módulos asignados a usuario |
| RF_30 | Asignar tópicos a módulo | 7.1.3.57 | Asignar tópico a módulo |
| RF_31 | Modificar tópicos asignado a módulo | 7.1.3.58 | Actualizar asignación de tópico a módulo |

| | | | | |
|--------------------|-------------------|--|--------------------------------|--|
| | RF_32 | Eliminar tópicos asignados a módulo | 7.1.3.59 | Eliminar asignación de tópico a módulo |
| | RF_33 | Listar tópicos asignado a módulo | 7.1.3.60 | Listar tópicos asignados a módulo |
| | RF_34 | Asignar tipo de dato a tópico | 7.1.3.40 | Asignar tipo de dato |
| | RF_35 | Modificar tipo de dato asignado a tópico | 7.1.3.41 | Actualizar tópico |
| Aplicación Android | RF_36 | Registrar usuario | 7.1.3.4 | Registrarse |
| | RF_38 | Iniciar sesión | 7.1.3.2 | Iniciar sesión |
| | RF_41 | Suscripción a tópicos | 7.1.3.11 | Suscribirse a tópicos |
| | RF_42 | Publicación en tópicos | 7.1.3.6 | Publicar en tópicos |
| | RF_43 | Sincronizar con módulo | 7.1.3.10 | Mensaje para la sincronización |
| | | | 7.1.3.14 | Respuesta de sincronización |
| | RF_44 | Controlar relés | 7.1.3.7 | Mensaje para controlar relés |
| | RF_45 | Obtener temperatura y humedad | 7.1.3.12 | Visualizar temperatura y humedad |
| RF_46 | Automatizar relés | 7.1.3.8 | Mensaje para automatizar relés | |

| | | | | |
|-----------------|-------------------------|---------------------------------------|-------------------------------------|--|
| | RF_47 | Desactivar horarios automáticos relés | 7.1.3.61 | Mensaje borrar horarios automáticos relés |
| | RF_48 | Establecer zona horaria | 7.1.3.9 | Mensaje para establecer zona horaria |
| | RF_49 | Ver zona horaria | 7.1.3.13 | Visualizar zona horaria |
| Módulo domótico | RF_50 | Inicio de sesión | 7.1.3.3 | Iniciar Sesión |
| | RF_51 | Suscripción a tópicos | 7.1.3.19 | Suscribirse a tópicos |
| | RF_52 | Publicación a tópicos | 7.1.3.15 | Publicar en tópicos |
| | RF_53 | Control remoto | 7.1.3.64 | Control remoto |
| | RF_54 | Automatizar relés | 7.1.3.20 | Solicitud para automatizar relés |
| | | | 7.1.3.22 | Automatizar relés |
| | RF_55 | Desactivar horarios automáticos relés | 7.1.3.62 | Solicitud para desactivar horarios automáticos relés |
| | RF_56 | Responder variables de ambiente | 7.1.3.16 | Responder temperatura y humedad |
| | RF_57 | Sincronizar hora | 7.1.3.21 | Solicitud para cambiar zona horaria |
| RF_58 | Establecer zona horaria | 7.1.3.21 | Solicitud para cambiar zona horaria | |

| | | | | |
|--|--------------|----------------------------|-----------------|-------------------------------|
| | | | 7.1.3.23 | Cambiar zona horaria |
| | RF_59 | Responder zona horaria | 7.1.3.17 | Responder zona horaria |
| | RF_60 | Responder a sincronización | 7.1.3.18 | Responder a la sincronización |

Tabla A -I Matriz de trazabilidad, requerimientos funcionales vs casos de uso