

ESCUELA POLITÉCNICA NACIONAL

INGENIERÍA EN SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE SEGURIDAD PARA CONTROL DOMÓTICO BASADO EN IOT BAJO AMBIENTES DE DISPOSITIVOS MÓVILES CON ANDROID

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN**

DARIO WLADIMIR HERRERA CHÁVEZ
dario.herrera@epn.edu.ec

DIRECTOR: MSc. RODRIGO FABIAN CHANCUSIG CHUQUILLA
rodrigo.chancusig@epn.edu.ec

CO-DIRECTOR: MSc. MARÍA MONSERRATE INTRIAGO PAZMIÑO
monserrate.intriago@epn.edu.ec

Quito, febrero 2020

DECLARACIÓN DE AUTORÍA

Yo Dario Wladimir Herrera Chávez, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

DARIO WLADIMIR HERRERA CHÁVEZ

CERTIFICACIÓN

Certificamos que el presente trabajo fue desarrollado por Dario Wladimir Herrera Chávez, bajo nuestra supervisión.

MSc. RODRIGO CHANCUSIG
DIRECTOR DEL TRABAJO DE
TITULACIÓN

MSc. MONSERRATE INTRIAGO
CO-DIRECTOR DEL TRABAJO DE
TITULACIÓN

AGRADECIMIENTOS

Agradezco en primer lugar a Dios, por las bendiciones que derrama sobre mí y mis seres queridos día tras día, por haberme cuidado y guiado durante toda mi carrera universitaria y por permitirme culminar mis estudios con éxito.

A mis padres Viviana y Santiago, por todo su amor, esfuerzo y apoyo incondicional en las distintas etapas de mi vida. A mi madre por siempre guiarme, motivarme y enseñarme a nunca rendirme ante las adversidades. A mi padre por sus enseñanzas y consejos. Gracias por creer en mí siempre.

A toda mi familia, por todo su cariño. En especial a mis abuelitas Olga y Clara, por sus cuidados, guías y consejos.

A todos mis amigos que formaron parte de esta etapa de mi vida, con los cuales he compartido muchas vivencias y buenos momentos. Gracias por sus consejos, apoyo y sobre todo por su sincera amistad.

A todos mis profesores de la Escuela Politécnica Nacional por su tiempo, esfuerzo, dedicación y sobre todo por compartir su conocimiento.

Finalmente, a mi director y co-director de proyecto de titulación, MSc. Rodrigo Chancusig y MSc. Monserrate Intriago, por su tiempo, conocimientos y por la confianza brindada para el desarrollo del proyecto hasta su éxito.

Dario Herrera

DEDICATORIA

Este proyecto de titulación va dedicado a mis padres, en especial a mi madre Viviana, por todo su esfuerzo, amor y apoyo incondicional, sin todos sus consejos, guías y valores no lo hubiese conseguido.

Dario Herrera

ÍNDICE DE CONTENIDO

ÍNDICE DE FIGURAS.....	XIII
ÍNDICE DE TABLAS.....	XVIII
RESUMEN.....	XXIII
PRESENTACIÓN	XXIV
CAPÍTULO I.....	1
1. PLANTEAMIENTO DEL PROBLEMA	1
1.1 ANTECEDENTES	1
1.2 OBJETIVOS	2
1.2.1 Objetivo General.....	2
1.2.2 Objetivos Específicos.....	2
1.3 ESTUDIO DE MERCADO.....	3
1.3.1 Análisis de resultados.....	7
1.4 PROYECTO INMOBILIARIO	7
1.4.1 Jardines Del Dean.....	7
1.4.1.1 Principales ventajas del sector.....	8
1.4.1.2 Principales ventajas del proyecto	8
1.4.1.3 Tipos de casas ofertadas por la inmobiliaria.....	9
1.4.1.3.1 Descripción del inmueble de dos plantas	9
CAPÍTULO II.....	11
2. FUNDAMENTOS TEÓRICOS.....	11
2.1 DOMÓTICA	11
2.1.1 Tecnologías de la Información y Domótica	11
2.1.2 Sistema Domótico	12
2.1.3 Gestión de la Domótica	12
2.1.3.1 Energía.....	12
2.1.3.2 Confort.....	13
2.1.3.3 Seguridad.....	13
2.1.3.3.1 Seguridad de los bienes.....	13
2.1.3.3.2 Seguridad de las personas	13
2.1.3.3.3 Incidentes y averías	13
2.1.3.4 Comunicaciones	13

2.2	TIPOS DE ARQUITECTURA.....	14
2.2.1	Arquitectura centralizada.....	14
2.2.2	Arquitectura distribuida.....	14
2.3	SENSORES.....	15
2.3.1	Sensores de temperatura.....	15
2.3.2	Sensores de movimiento.....	15
2.3.3	Sensores de calidad de aire.....	15
2.3.4	Sensores magnéticos de apertura.....	16
2.3.5	Sensores de luz.....	16
2.3.6	Sensores de rotura de cristal.....	16
2.3.7	Sensores de vibración sísmica.....	17
2.3.8	Sensores de inundación.....	17
2.4	ACTUADORES.....	17
2.4.1	Motores DC.....	17
2.4.2	Servomotores.....	17
2.4.3	Cámaras.....	18
2.4.4	Relés.....	18
2.4.5	Optoacopladores.....	18
2.5	INTERNET DE LAS COSAS.....	18
2.5.1	Internet para todo.....	19
2.5.2	Arquitectura IoT.....	20
2.5.2.1	Dispositivos IoT.....	20
2.5.2.2	Redes.....	21
2.5.2.3	Nube.....	21
2.5.3	Aplicaciones IoT.....	21
2.5.3.1	Edificios inteligentes.....	22
2.5.3.2	Ciudades inteligentes y transporte.....	22
2.5.3.3	Educación.....	22
2.5.3.4	Electrónica de consumo.....	22
2.5.3.5	Salud.....	22
2.5.3.6	Agricultura y medio ambiente.....	22
2.5.3.7	Servicios de energía.....	23
2.5.3.8	Compras inteligentes.....	23
2.6	PLATAFORMAS IOT.....	23
2.6.1	Plataformas Hardware.....	23

2.6.1.1	Arduino	23
2.6.1.1.1	Modelos de Arduino	23
2.6.1.1.2	Características de Arduino	24
2.6.1.1.2.1	Libre y extensible	24
2.6.1.1.2.2	Amplia comunidad	25
2.6.1.1.2.3	Multiplataforma	25
2.6.1.1.2.4	Lenguaje de programación intuitivo	25
2.6.1.1.2.5	Bajo costo	25
2.6.1.2	Raspberry Pi	25
2.6.1.2.1	Modelos de Raspberry Pi	26
2.6.1.2.1.1	Raspberry Pi 3 B+	27
2.6.1.2.1.1.1	Características técnicas	28
2.6.1.2.1.1.2	Pines de propósito general (GPIO)	28
2.6.1.2.1.1.2.1	Voltajes	29
2.6.1.2.1.1.2.2	Entradas	29
2.6.1.2.1.1.2.3	Salidas	29
2.6.1.2.1.1.2.4	GPIO especiales	29
2.6.1.2.1.1.2.5	Nomenclatura	29
2.6.1.2.1.1.2.5.1	Modo BOARD	29
2.6.1.2.1.1.2.5.2	Modo BCM	30
2.6.2	Plataformas Software	30
2.6.2.1	Firestore	30
2.6.2.1.1	Servicios	30
2.6.2.1.1.1	Cloud Firestore	30
2.6.2.1.1.2	Kit de AA	31
2.6.2.1.1.3	Cloud Functions	31
2.6.2.1.1.4	Authentication	31
2.6.2.1.1.5	Cloud Storage	32
2.6.2.1.1.6	Cloud Messaging	32
2.7	SISTEMA OPERATIVO ANDROID	32
2.7.1	Características	32
2.7.2	Versiones	33
2.7.3	Arquitectura	35
2.7.3.1	Kernel Linux	35
2.7.3.2	Bibliotecas	35

2.7.3.3	Marco de trabajo de aplicaciones	36
2.7.3.4	Aplicaciones	37
2.8	METODOLOGÍA DE DESARROLLO DE SOFTWARE.....	37
2.8.1	Marco de Trabajo Scrum.....	38
2.8.1.1	Artefactos De Scrum.....	38
2.8.1.1.1	Product Backlog.....	38
2.8.1.1.2	Sprint Backlog.....	39
2.8.1.1.3	Incremento.....	39
2.8.1.2	Eventos de Scrum	39
2.8.1.2.1	Sprint	39
2.8.1.2.2	Reunión de planificación del Sprint (Sprint Planning Meeting)	39
2.8.1.2.3	Scrum diario (Daily Scrum)	39
2.8.1.2.4	Revisión del Sprint (Sprint Review)	39
2.8.1.2.5	Retrospectiva del Sprint (Sprint Retrospective).....	40
2.8.1.3	Roles en Scrum	40
2.8.1.3.1	Propietario del Producto (Product Owner).....	40
2.8.1.3.2	Equipo de Desarrollo (Development Team).....	40
2.8.1.3.3	Scrum Master.....	40
CAPÍTULO III.....		42
3.	DISEÑO DEL PROTOTIPO DE SEGURIDAD	42
3.1	PROTOTIPO DEL INMUEBLE	42
3.1.1	Prototipo 2D del inmueble.....	42
3.1.2	Prototipo del inmueble en 3D	43
3.1.2.1	Vista superior del inmueble prototipo.....	43
3.1.2.2	Vista superior lateral derecha del inmueble prototipo.....	43
3.1.2.3	Vista superior lateral izquierda del inmueble prototipo.....	44
3.2	ELICITACIÓN DE REQUERIMIENTOS.....	44
3.2.1	Requerimientos iniciales.....	44
3.2.2	Pila de producto (Product Backlog)	46
3.3	PLANIFICACIÓN DE SPRINTS	53
3.3.1	Sprint 1.....	54
3.3.1.1	Objetivo del Sprint 1	54
3.3.2	Sprint 2.....	55
3.3.2.1	Objetivo del Sprint 2.....	55

3.3.3	Sprint 3.....	55
3.3.3.1	Objetivo del Sprint 3.....	56
3.3.4	Sprint 4.....	56
3.3.4.1	Objetivo del Sprint 4.....	56
3.3.5	Sprint 5.....	57
3.3.5.1	Objetivo del Sprint 5.....	57
3.3.6	Sprint 6.....	58
3.3.6.1	Objetivo del Sprint 6.....	58
3.3.7	Sprint 7.....	59
3.3.7.1	Objetivo del Sprint 7.....	59
3.3.8	Sprint 8.....	60
3.3.8.1	Objetivo del Sprint 8.....	60
3.3.9	Sprint 9.....	61
3.3.9.1	Objetivo del Sprint 9.....	61
3.3.10	Sprint 10.....	62
3.3.10.1	Objetivo del Sprint 10.....	62
3.3.11	Sprint 11.....	63
3.3.11.1	Objetivo del Sprint 11.....	63
3.3.12	Sprint 12.....	64
3.3.12.1	Objetivo del Sprint 12.....	64
3.3.13	Sprint 13.....	65
3.3.13.1	Objetivo del Sprint 13.....	65
3.4	DISEÑO DE LA ARQUITECTURA DEL SISTEMA DE SEGURIDAD	66
3.4.1	Diseño de la arquitectura	66
3.5	DISEÑO DE LA ESTRUCTURA PARA LA BASE DE DATOS	67
3.5.1	Diseño de la estructura NoSQL.....	68
3.6	DISEÑO DE LAS INTERFACES DEL APLICATIVO MÓVIL.....	68
3.6.1	Diseño de la interfaz de inicio de sesión.....	69
3.6.2	Diseño de la interfaz para creación de cuenta.....	69
3.6.3	Diseño de la interfaz para restablecer contraseña.....	70
3.6.4	Diseño del menú lateral.....	70
3.6.5	Diseño de la interfaz para mostrar el historial de eventos	71
3.6.6	Diseño de la interfaz para control de cámara y visualización de video	71
3.6.7	Diseño de la interfaz para control de garaje.....	72
3.6.8	Diseño de la interfaz para control de iluminación.....	72

3.6.9	Diseño de la interfaz para monitoreo de temperatura	73
3.6.10	Diseño de la interfaz para control del simulador de presencia.....	73
3.6.11	Diseño de la interfaz para configuraciones	74
3.7	DISEÑO DE LOS CIRCUITOS ELECTRÓNICOS	74
3.7.1	Definición de pines del encabezado GPIO.....	74
3.7.2	Diagrama del circuito para lectura de temperatura	75
3.7.3	Diagrama del circuito para detección de gases.....	75
3.7.4	Diagrama del circuito para detección de movimiento	76
3.7.5	Diagrama del circuito para control del ventilador	76
3.7.6	Diagrama del circuito para control del calefactor y buzzer	76
3.7.7	Diagrama del circuito para detección de apertura de puertas.....	77
3.7.8	Diagrama del circuito para control de servomotores	77
3.7.9	Diagrama del circuito para detección de luminosidad	77
3.7.10	Diagrama del circuito de potencia para iluminación	78
3.7.10.1	Diseño del circuito impreso para potencia de iluminación	78
3.7.11	Diagrama del circuito para control de iluminación.....	79
3.7.11.1	Diseño del circuito impreso para control de iluminación	79
CAPÍTULO IV		81
4.	CONSTRUCCIÓN DEL PROTOTIPO DE SEGURIDAD	81
4.1	INSTALACIÓN DE RASPBIAN EN RASPBERRY PI	81
4.1.1	Descarga del sistema operativo Raspbian	81
4.1.2	Descarga e instalación del programa balenaEtcher	82
4.1.3	Instalación de Raspbian en Raspberry Pi.....	82
4.1.4	Configuración y habilitación de interfaces en Raspberry Pi	83
4.2	HERRAMIENTAS	84
4.2.1	Control de versiones	85
4.2.2	Lenguajes de programación y base de datos	85
4.2.3	Entornos de desarrollo para hardware y software.....	86
4.2.4	Librerías	86
4.2.5	Recursos y diagramación.....	87
4.3	CONSTRUCCIÓN DEL APLICATIVO MÓVIL	88
4.3.1	Compatibilidad del aplicativo móvil.....	88
4.3.2	Creación de proyecto e integración del SDK de Firebase	88
4.3.3	Métodos de acceso para autenticación de usuarios en Firebase.....	92

4.3.4	Estructura del proyecto en Android Studio	93
4.3.5	Construcción de la interfaz para inicio de sesión	97
4.3.6	Construcción de la interfaz para creación de cuenta.....	98
4.3.7	Construcción de la interfaz para restablecimiento de contraseña.....	99
4.3.8	Construcción del Navigation Drawer	100
4.3.9	Construcción de la interfaz para mostrar el historial de eventos	102
4.3.10	Construcción de la interfaz para control y visualización de video	103
4.3.11	Construcción de la interfaz para control de garaje.....	105
4.3.12	Construcción de la interfaz para control de iluminación.....	106
4.3.13	Construcción de la interfaz para control y monitoreo de temperatura	107
4.3.14	Construcción de la interfaz para control del simulador de presencia.....	108
4.3.15	Construcción de la interfaz para configuraciones	109
4.3.16	CONSTRUCCIÓN DE LOS CUADROS DE DIÁLOGO.....	110
4.3.16.1	Cuadro de diálogo en la interfaz de historial de eventos	110
4.3.16.2	Cuadro de diálogo en la interfaz de simulador de presencia	111
4.3.16.3	Cuadro de diálogo en la interfaz de temperatura	111
4.3.16.4	Cuadros de diálogo en la interfaz de configuración	112
4.3.16.4.1	Cuadro de diálogo para establecer niveles de activación.....	112
4.3.16.4.2	Cuadro de diálogo para establecer niveles de intensidad	113
4.3.17	Construcción de la interfaz que notifica la no existencia de conexión.....	113
4.4	CONSTRUCCIÓN DEL HARDWARE	114
4.4.1	Construcción del circuito para lectura de temperatura	114
4.4.2	Construcción del circuito para la detección de gases.....	115
4.4.3	Construcción del circuito para la detección de movimiento.....	116
4.4.4	Construcción del circuito para el control del ventilador	117
4.4.5	Construcción del circuito para control del calefactor y buzzer	118
4.4.6	Construcción del circuito para detección de apertura de puertas.....	118
4.4.7	Construcción del circuito para el control de servomotores	119
4.4.8	Construcción del circuito para detección de luminosidad	121
4.4.9	Construcción del circuito de potencia para iluminación.....	121
4.4.10	Construcción del circuito para control de iluminación.....	122
4.4.11	Instalación de cámara en Raspberry Pi	123
4.5	CONSTRUCCIÓN DE LOS CONTROLADORES DE HARDWARE	124
4.6	Integración del SDK de Firebase a Python	124
4.6.1	Estructura del proyecto en Python.....	125

4.7	IMPLEMENTACIÓN DEL SISTEMA DE SEGURIDAD	129
CAPÍTULO V		131
5.	PRUEBAS Y RESULTADOS.....	131
5.1	Pruebas funcionales.....	131
5.1.1	Creación y autenticación de cuentas a través de correo electrónico	131
5.1.2	Creación y autenticación de cuentas a través de un proveedor de acceso... ..	132
5.1.3	Validación de campos de los formularios	133
5.1.4	Restablecimiento de contraseña.....	135
5.1.5	Control de iluminación	136
5.1.6	Visualización de video en tiempo real.....	137
5.1.7	Control de la rotación de cámara.....	138
5.1.8	Control de apertura y cierre del garaje.....	139
5.1.9	Control del simulador de presencia	140
5.1.10	Monitoreo y control de temperatura.....	142
5.1.11	Detección de movimiento	143
5.1.12	Detección de apertura de puertas.....	144
5.1.13	Control automático del alumbrado externo.....	145
5.1.14	Detección de gases nocivos	146
5.1.15	Visualización del historial de eventos.....	147
5.2	Resultados y Discusión	148
CAPÍTULO VI		150
6.	CONCLUSIONES Y RECOMENDACIONES	150
6.1	Conclusiones.....	150
6.2	Recomendaciones.....	152
ANEXOS.....		160

ÍNDICE DE FIGURAS

Figura 1. Proyectos inmobiliarios por sector.....	3
Figura 2. Número de proyectos por tipo de producto inmobiliario.....	4
Figura 3. Número de proyectos por rango de precio	4
Figura 4. Número de inmuebles reservados por sector	5
Figura 5. Número de inmuebles reservados por tipo de producto	5
Figura 6. Número de inmuebles reservados por rango de precio	5
Figura 7. Preferencia de los asistentes por el tipo de inmueble.....	6
Figura 8. Preferencia de los asistentes por el sector del inmueble.....	6
Figura 9. Preferencia de los asistentes por el precio del inmueble.....	6
Figura 10. Tiempo estimado de compra de los inmuebles por parte de los asistentes	7
Figura 11. Mapa de acceso al proyecto inmobiliario.....	8
Figura 12. Vista aérea del proyecto inmobiliario.....	9
Figura 13. Plano referencial del inmueble de dos plantas	10
Figura 14. Arquitectura centralizada.....	14
Figura 15. Arquitectura distribuida.....	14
Figura 16. Arquitectura IoT en bloques	21
Figura 17. Pines de propósito general GPIO	28
Figura 18. Cloud Firestore	31
Figura 19. Kit de AA.....	31
Figura 20. Cloud Functions	31
Figura 21. Firebase Authentication.....	31
Figura 22. Cloud Storage	32
Figura 23. Firebase Cloud Messaging.....	32
Figura 24. Pila de arquitectura del sistema operativo Android.....	37
Figura 25. Marco de Desarrollo Scrum.....	41
Figura 26. Modelo prototipo de la vivienda.....	43

Figura 27. Vista superior del inmueble prototipo	43
Figura 28. Vista superior lateral derecha.....	43
Figura 29. Vista superior lateral izquierda	44
Figura 30. Arquitectura del prototipo de seguridad para control domótico	67
Figura 31. Estructura NoSQL para Firestore	68
Figura 32. Interfaz para inicio de sesión.....	69
Figura 33. Interfaz para creación de cuenta	69
Figura 34. Interfaz para restablecimiento de contraseña.....	70
Figura 35 - Diseño del menú lateral	70
Figura 36. Interfaz para mostrar el historial de eventos.....	71
Figura 37. Interfaz para control y visualización de video en tiempo real.....	71
Figura 38. Interfaz para control de garaje.....	72
Figura 39. Interfaz para control de iluminación.....	72
Figura 40. Interfaz para monitoreo de temperatura	73
Figura 41. Interfaz para control del simulador de presencia	73
Figura 42. Interfaz para configuraciones	74
Figura 43. Definición de pines del encabezado GPIO	75
Figura 44. Diagrama del circuito para lectura de temperatura	75
Figura 45. Diagrama del circuito para la detección de gases	75
Figura 46. Diagrama del circuito para detección de movimiento	76
Figura 47. Diagrama del circuito para control del ventilador	76
Figura 48. Diagrama del circuito para control del calefactor y buzzer.....	76
Figura 49. Diagrama del circuito para detección de apertura de puertas.....	77
Figura 50. Diagrama del circuito para control de servomotores.....	77
Figura 51. Diagrama del circuito para detección de luminosidad.....	77
Figura 52. Diagrama del circuito de potencia para iluminación.....	78
Figura 53. Diseño del PCB para potencia de iluminación	78

Figura 54. Previsualización del PCB para potencia de iluminación	79
Figura 55. Diagrama del circuito para control de iluminación	79
Figura 56. Diseño del PCB para control de iluminación.....	80
Figura 57. Previsualización del PCB para control de iluminación.	80
Figura 58. Imagen del sistema operativo Raspbian.....	81
Figura 59. Sitio oficial de la herramienta balenaEtcher	82
Figura 60. Interfaz del software balenaEtcher	82
Figura 61. Conexión de periféricos Raspberry Pi	83
Figura 62. Interfaz del sistema operativo Raspbian.....	83
Figura 63. Interfaces Raspberry Pi.....	84
Figura 64. Habilitación de interfaces Raspberry Pi.....	84
Figura 65. Porcentaje de dispositivos móviles compatibles con la app.....	88
Figura 66. Sitio oficial de Firebase	88
Figura 67. Creación de proyectos en Firebase.....	89
Figura 68. Asignación de nombre del proyecto	89
Figura 69. Adición de Google Analytics.....	89
Figura 70. Selección de zona y condiciones del servicio.....	90
Figura 71. Consola de Firebase	90
Figura 72. Plataformas disponibles	90
Figura 73. Ingreso de información del aplicativo móvil	91
Figura 74. Generación de archivo google-services.json	91
Figura 75. Adición del archivo google-services.json al árbol del proyecto	91
Figura 76. Dependencias a nivel de proyecto	92
Figura 77. Dependencias a nivel de aplicación	92
Figura 78. Habilitación de proveedores de acceso.....	93
Figura 79. Estructura del proyecto en Android Studio	93
Figura 80. Estructura lógica de la aplicación móvil.....	95

Figura 81. Interfaz para inicio de sesión.....	97
Figura 82. Interfaz para creación de cuenta	98
Figura 83. Interfaz para restablecimiento de contraseña.....	99
Figura 84. Interfaz del Navigation Drawer	100
Figura 85. Interfaz para mostrar el historial de eventos.....	102
Figura 86. Interfaz que notifica la no existencia de eventos	102
Figura 87. Interfaz para rotar cámara y visualizar video en tiempo real.....	103
Figura 88. Interfaz que notifica que la opción de video está deshabilitada	104
Figura 89. Interfaz que notifica que algo ha salido mal en el despliegue de video.....	104
Figura 90. Interfaz para la apertura y cierre de garaje.....	105
Figura 91. Interfaz para el control de iluminación	106
Figura 92. Interfaz para el monitoreo de temperatura.....	107
Figura 93. Interfaz para el control del simulador de presencia	108
Figura 94. Interfaz que permite la configuración de parámetros.....	109
Figura 95. Cuadro de diálogo confirmar eliminación de evento.	110
Figura 96. Cuadro de diálogo para selección de ambientes.	111
Figura 97. Cuadro de diálogo confirmación de eliminación de gráfico de temperatura. .	111
Figura 98. Cuadro de diálogo para establecer nivel de activación del ventilador.....	112
Figura 99. Cuadro de diálogo para establecer nivel de activación del calefactor.....	112
Figura 100. Cuadro de diálogo para establecer nivel de intensidad	113
Figura 101. Interfaz que notifica la usencia de una red de internet.....	113
Figura 102. Circuito electrónico para la lectura de temperatura	115
Figura 103. Circuito electrónico para la detección de gases.....	115
Figura 104. Circuito electrónico para la detección de movimiento.....	116
Figura 105. Circuito electrónico para el control del ventilador	117
Figura 106. Circuito electrónico para la activación del calefactor y buzzer	118
Figura 107. Circuito electrónico para la detección de apertura de puertas	119

Figura 108. Circuito electrónico para control de servomotor para rotara cámara.....	120
Figura 109. Circuito electrónico para control de servomotor de garaje	120
Figura 110. Circuito electrónico para la detección de luminosidad	121
Figura 111. Placa de circuito impreso PCB para potencia de iluminación	122
Figura 112. Placa de circuito impreso para control de lámparas	123
Figura 113. Raspberry Pi cámara con visión nocturna	123
Figura 114. Instalación de la cámara en Raspberry Pi	124
Figura 115. Generación de clave privada en Firebase para Python	125
Figura 116. Generación del archivo que contendrá la clave privada	125
Figura 117. Estructura del proyecto para los controladores de hardware en Python	126
Figura 118. Vivienda prototipo para la implementación del sistema de seguridad para control domótico	129
Figura 119. Implementación del sistema de seguridad en la vivienda prototipo	130
Figura 120. Resultados pruebas funcionales	148

ÍNDICE DE TABLAS

Tabla 1. Modelos de tarjetas electrónicas Arduino	24
Tabla 2. Modelos de tarjetas Raspberry Pi.....	26
Tabla 3. Características técnicas de Raspberry Pi 3 Modelo B+.....	28
Tabla 4. Versiones del sistema operativo Android.....	33
Tabla 5. Roles bajo el marco de desarrollo Scrum para el diseño y construcción del prototipo de seguridad	44
Tabla 6. R01_Diseñar la arquitectura del prototipo de seguridad para control domótico..	46
Tabla 7. R02_Centralizar toda la información en un dispositivo móvil	46
Tabla 8. R03_Gestión de usuarios para el acceso al aplicativo móvil.....	46
Tabla 9. R04_Creación y autenticación de cuentas para el acceso a través de correo electrónico	47
Tabla 10. R05_Creación y autenticación de cuentas para el acceso a través de un proveedor de acceso	47
Tabla 11. R06_Confirmación de email antes de otorgar acceso al aplicativo móvil	47
Tabla 12. R07_Opción de restablecimiento de contraseña para el acceso al aplicativo móvil	47
Tabla 13. R08_Validación en campo email.....	47
Tabla 14. R09_ Validación y aplicación de una expresión regular al campo contraseña .	48
Tabla 15. R10_Validación y verificación en los campos de contraseña	48
Tabla 16. R11_Control de iluminación a través del dispositivo móvil	48
Tabla 17. R12_Control de iluminación a través de placa electrónica	48
Tabla 18. R13_Sincronización de interruptores tanto digitales como físicos.....	48
Tabla 19. R14_Automatización para la activación y desactivación de toda la iluminación	49
Tabla 20. R15_Visualización de video en tiempo real.....	49
Tabla 21. R16_Generación de un grado de libertad para rotar la cámara.....	49
Tabla 22. R17_Automatización para la apertura y cierre de garaje.....	49

Tabla 23. R18_Automatización para la activación y desactivación de un simulador de presencia.....	49
Tabla 24. R19_Lectura y gráfica en tiempo real de la temperatura del hogar	50
Tabla 25. R20_Automatización para el control de ventilador y calefactor	50
Tabla 26. R21_Detección de movimiento en el hogar.....	50
Tabla 27. R22_Envío automático de email con fotografía al detectar movimiento	50
Tabla 28. R23_Posteo automático con fotografía en la red social Twitter al detectar movimiento	50
Tabla 29. R24_Detección de apertura de puertas	51
Tabla 30. R25_Control automático para la activación y desactivación del alumbrado externo	51
Tabla 31. R26_Detección de humo y gases nocivos dentro del hogar.....	51
Tabla 32. R27_Activación automática de extractor de aire y alarma sonora frente a la detección de gases.....	51
Tabla 33. R28_Recepción de notificaciones push en tiempo real frente a incidentes dentro del hogar.....	51
Tabla 34. R29_Historial de acciones e incidentes efectuados dentro del hogar.....	52
Tabla 35. R30_Apartado de configuraciones del aplicativo móvil	52
Tabla 36. R31_Control del sistema de seguridad a través del aplicativo móvil desde cualquier ubicación	52
Tabla 37. R32_Implementación del sistema de seguridad en un prototipo de vivienda ...	52
Tabla 38. Tareas a efectuarse en Sprint 1	54
Tabla 39. Tareas a efectuarse en Sprint 2.....	55
Tabla 40. Tareas a efectuarse en Sprint 3.....	56
Tabla 41. Tareas a efectuarse en Sprint 4.....	57
Tabla 42. Tareas a efectuarse en Sprint 5.....	58
Tabla 43. Tareas a efectuarse en Sprint 6.....	59
Tabla 44. Tareas a efectuarse en Sprint 7.....	60
Tabla 45. Tareas a efectuarse en Sprint 8.....	61

Tabla 46. Tareas a efectuarse en Sprint 9.....	62
Tabla 47. Tareas a efectuarse en Sprint 10.....	63
Tabla 48. Tareas a efectuarse en Sprint 11.....	64
Tabla 49. Tareas a efectuarse en Sprint 12.....	65
Tabla 50. Tareas a efectuarse en Sprint 13.....	66
Tabla 51. Herramientas para control de versiones	85
Tabla 52. Lenguajes de programación y base de datos.....	85
Tabla 53. Entornos de desarrollo para hardware y software	86
Tabla 54. Librerías para desarrollo	86
Tabla 55. Recursos y diagramación	87
Tabla 56. Descripción de los componentes de la interfaz de inicio de sesión	98
Tabla 57. Descripción de los componentes de la interfaz para crear cuenta	99
Tabla 58. Descripción de los componentes de la interfaz para restablecimiento de contraseña.....	100
Tabla 59. Descripción de los componentes del Navigation Drawer.....	101
Tabla 60. Descripción de los componentes de la interfaz para mostrar el historial de eventos	103
Tabla 61. Descripción de los componentes de la interfaz para rotación y visualización de video.....	105
Tabla 62. Descripción de los componentes de la interfaz para la apertura y cierre del garaje	106
Tabla 63. Descripción de los componentes de la interfaz para el control de la iluminación	107
Tabla 64. Descripción de los componentes de la interfaz para el monitoreo de temperatura	108
Tabla 65. Descripción de los componentes de la interfaz para activar o desactivar el simulador de presencia.....	109
Tabla 66. Descripción de los componentes de la interfaz de configuración	110

Tabla 67. Descripción del componente del cuadro de diálogo para confirmación de eliminación de evento	111
Tabla 68. Descripción del componente del cuadro de diálogo para selección de ambientes	111
Tabla 69. Descripción del componente del cuadro de diálogo para confirmación de eliminación del gráfico de temperatura	112
Tabla 70. Descripción del componente del cuadro de diálogo para establecer niveles de activación.....	112
Tabla 71. Descripción del componente del cuadro de diálogo para ajustar nivel de intensidad de las lámparas	113
Tabla 72. Descripción de los componentes para notificación de la no existencia de una conexión a internet	114
Tabla 73. Especificaciones y características del sensor DS18B20	115
Tabla 74. Especificación y características del sensor MQ-135	116
Tabla 75. Especificaciones y características del sensor PIR HC-SR501	116
Tabla 76. Especificaciones del ventilador	117
Tabla 77. Especificaciones del buzzer.....	118
Tabla 78. Especificaciones del sensor magnético MC-38.....	119
Tabla 79. Especificaciones del driver IRF520 y del micro servo SG90	120
Tabla 80. Especificaciones de la LDR	121
Tabla 81. Componentes utilizados para la construcción del PCB para potencia de iluminación.....	122
Tabla 82. Componentes utilizados para la construcción del PCB	123
Tabla 83. Componentes de hardware a instalarse en la vivienda prototipo	130
Tabla 84. Criterios de la escala cualitativa para pruebas y resultados.....	131
Tabla 85. P01_Prueba para la creación y autenticación de cuentas a través de correo electrónico	131
Tabla 86. P02_Prueba para la creación y autenticación de cuentas a través de un proveedor de acceso	132

Tabla 87. P03_Prueba para la validación de los campos del formulario para inicio de sesión	133
Tabla 88. P04_Prueba para la validación de los campos del formulario para creación de cuenta.....	134
Tabla 89. P05_Prueba para la validación de los campos del formulario para restablecimiento de contraseña	134
Tabla 90. P06_Prueba para restablecimiento de contraseña.....	135
Tabla 91. P07_Prueba para el control de iluminación.....	136
Tabla 92. P08_Prueba para la visualización de video en tiempo real	138
Tabla 93. P09_Prueba para control de la rotación de cámara	139
Tabla 94. P10_Prueba para el control de apertura y cierre del garaje	140
Tabla 95. P11_Prueba para el control del simulador de presencia	141
Tabla 96. P12_Prueba para el monitoreo y control de temperatura	142
Tabla 97. P13_Prueba para la detección de movimiento.....	143
Tabla 98. P14_Prueba para la detección de apertura de puertas	144
Tabla 99. P15_Prueba para el control automático del alumbrado externo	145
Tabla 100. P16_Prueba para la detección de gases nocivos.....	146
Tabla 101. P17_Prueba para la visualización del historial de eventos e incidentes del hogar	147

RESUMEN

La vivienda constituye un activo de suma importancia; sobre ésta, los seres humanos depositan sus bienes materiales y sirve de refugio acondicionado para el diario vivir. En su mayoría, las viviendas están expuestas a robos e incidentes, lo cual supone un peligro en contra de la integridad física de sus miembros y bienes. El mercado ofrece soluciones de seguridad para el sector de la vivienda, sin embargo, son costosas y atadas a plataformas o fabricantes. Por ello, como solución al problema, se ha planteado el diseño e implementación de un prototipo de seguridad de bajo coste para control domótico basado en internet de las cosas (IoT) bajo ambientes de dispositivos móviles con Android.

El sistema de seguridad para control domótico fue construido bajo el marco de desarrollo Scrum y está conformado por cuatro componentes esenciales. El primero hace referencia al hardware (Raspberry Pi, sensores y actuadores) que fue instalado en el inmueble prototipo. El segundo se basa en los controladores de hardware desarrollados con Python. El tercero consiste en una aplicación móvil para la plataforma Android, utilizando Kotlin. Finalmente, el cuarto consiste en la base de datos que permite el almacenamiento de toda la información generada por el usuario y los diversos elementos que coexisten en el sistema, utilizando Firebase.

Al concluir el diseño e implementación del prototipo de seguridad para control domótico, se obtuvo como resultado un sistema económico, innovador, de grandes prestaciones orientadas al confort y al elevamiento de seguridad física de los miembros del hogar y sus bienes; convirtiéndola así en una solución tecnológica al alcance de la mayor parte de sectores socioeconómicos de la ciudad de Quito.

Palabras clave: Vivienda, Seguridad Física, Domótica, Internet de las Cosas, Android, Scrum.

PRESENTACIÓN

En este documento se presenta el diseño e implementación de un prototipo de seguridad para control domótico basado en IoT bajo ambientes de dispositivos móviles con Android.

El Capítulo I expone el planteamiento del problema y los antecedentes relacionados con las viviendas, miembros y bienes materiales frente a posibles incidentes que atenten contra la integridad física. Sobre la problemática expuesta se plantea una solución basada en el uso de las tecnologías de la información.

Posteriormente se expone un estudio de mercado, análisis de resultados y elección de un proyecto inmobiliario para la implementación del sistema de seguridad prototipo que aquí se presenta.

El Capítulo II define los fundamentos teóricos relacionados con las tecnologías, dispositivos, plataformas y metodologías usadas en el diseño e implementación del sistema de seguridad para control domótico.

En este capítulo se encuentran definiciones, características, campos de aplicación, especificaciones técnicas y arquitecturas relacionadas con: domótica, internet de las cosas, sistema operativo Android, sensores, actuadores, plataformas hardware, plataformas software y el marco de trabajo con Scrum.

El Capítulo III está orientado al diseño de los diversos elementos que forman parte del sistema de seguridad. Inicia con el diseño de la vivienda prototipo, posteriormente se presenta la elicitación de requerimientos basados en características que aporten al confort, mejoramiento de la calidad de vida y el incremento de seguridad física hacia los miembros de un hogar.

En este capítulo también se definen los Sprints llevados a cabo en conjunto con sus objetivos. Además, se presenta el diseño de la arquitectura, base de datos, interfaces y circuitos que dan soporte al funcionamiento del sistema de seguridad.

El Capítulo IV se enmarca en la construcción de todos los elementos que dan soporte al sistema de seguridad diseñados en el Capítulo III. Se presenta los lenguajes, herramientas y tecnologías usadas, así como también la implementación, pruebas efectuadas y resultados obtenidos del sistema de seguridad en la vivienda prototipo.

El Capítulo V corresponde a las conclusiones y recomendaciones obtenidas a lo largo del diseño, desarrollo e implementación del proyecto.

CAPÍTULO I

1. PLANTEAMIENTO DEL PROBLEMA

1.1 ANTECEDENTES

La evolución tecnológica ha permitido cambiar el panorama del mundo actual sobre el cual se desarrolla el ser humano, se ha adherido a todas las actividades que una persona puede llevar a cabo en su diario vivir tales como: educación, trabajo, ocio o entretenimiento. Dicha evolución tecnológica ha permitido la creación de un sin número de tecnologías que en la antigüedad no se pensaba que pudieran cristalizarse.

La humanidad en la época actual se desenvuelve y vive en un mundo donde la interacción con la tecnología es constante, habla por teléfonos celulares, realiza videoconferencias desde la comodidad de su hogar, efectúa transacciones bancarias desde dispositivos electrónicos, trabaja desde el hogar con computadores personales y un sin número más de actividades que resultaría en una lista sumamente extensa. Sin embargo, un lugar en donde aún no se ha logrado revolucionar la inmersión tecnológica en su totalidad, es la vivienda. Lugar al cual hay que brindarle suma importancia, ya que permite albergar a personas tanto a nivel individual como a nivel familiar, sobre esta los seres humanos depositan sus bienes materiales, lo cual la convierte en un refugio diario acondicionado para que un ser humano pueda vivir.

Aun cuando las viviendas son construidas con mecanismos tradicionales de seguridad, tales como: puertas, ventanas, protecciones de hierro o incluso con la incorporación de algún equipo tecnológico, los habitantes del hogar y sus bienes se encuentran frente a riesgos o incidentes que en muchos de los casos pudieron ser evitados de una manera sencilla y efectiva.

Actualmente el ser humano se encuentra en un ritmo de vida muy agitado, por lo cual es muy frecuente que los hogares permanezcan solos por largos periodos de tiempo, esto debido a que sus miembros podrían encontrarse en situaciones rutinarias tales como: estudio, trabajo o a su vez factores como viajes de negocio o simplemente vacaciones. Frente a estas situaciones es muy común que los delincuentes aprovechen la ausencia de los integrantes del hogar para cometer robos dentro del inmueble.

En los hogares también se presentan otros tipos de situaciones de riesgo, ya sea que esta se encuentre sola o habitada. En la vivienda se pueden desembocar incidentes tales como: fugas de gas, misma que puede desencadenar un incendio que atente contra la

vida de las personas que se encuentren dentro del hogar, daños y pérdidas de los bienes materiales o a su vez con la contaminación del aire que respiran los miembros del hogar.

Existe otro aspecto relevante como son las personas de la tercera edad y personas con movilidad reducida, mismas que requieren de una constante vigilancia, pero a su vez requieren de cierto nivel de independencia. Este grupo de personas son vulnerables frente posibles incidentes dentro del hogar, mismos que fácilmente pudiesen ser evitados mediante el uso de video en tiempo real, como mecanismo de monitoreo.

Frente a toda la problemática expuesta surge la domótica, misma permite una mayor calidad de vida, una reducción del trabajo doméstico, un aumento del bienestar y la seguridad de sus habitantes y una racionalización de los distintos consumos [1]. Hoy en día se cuenta con algunos sistemas domóticos avanzados que permiten mantener un control sobre los hogares; no obstante, su costo es excesivo, evitando que las personas lo visualicen como un apoyo para el mejoramiento de la calidad de vida, sino que más bien está visto como algo destinado a la vanidad.

1.2 OBJETIVOS

1.2.1 Objetivo General

Diseñar e implementar un prototipo de seguridad de bajo coste para control domótico basado en IoT haciendo uso de un aplicativo móvil desarrollado para la plataforma móvil Android.

1.2.2 Objetivos Específicos

- Analizar y definir los requerimientos para el diseño de un sistema de seguridad domótico de bajo coste.
- Diseñar un sistema de seguridad prototipo que aumente el nivel de seguridad física tanto de los miembros del hogar, así como de los bienes materiales, de bajo coste y que se adapte a los requerimientos definidos.
- Desarrollar un aplicativo para la plataforma móvil Android que permita controlar y monitorear el sistema de seguridad.
- Validar el funcionamiento del sistema de seguridad con un prototipo en un ambiente de pruebas que simule las características de un hogar.

1.3 ESTUDIO DE MERCADO

Inicialmente el presente sistema de seguridad prototipo está dirigido hacia todas las personas que deseen incorporar un nivel adicional de seguridad a sus hogares. Reforzando los elementos de seguridad tradicionales (puertas, ventanas, cercas, etc.) con la implementación de un sistema de seguridad que permitirá elevar la seguridad física de los miembros del hogar, así como de sus bienes.

En este sentido, el sistema de seguridad está ideado para ser usado en diferentes tipos de inmuebles tales como casas, departamentos, oficinas, entre otros. La diferencia radicará en el espacio físico de las mismas, en las cuales según cada caso se usará una cantidad superior o inferior de sensores y actuadores.

Frente a la diversidad de inmuebles en el territorio ecuatoriano, el sistema de seguridad puede ser acoplado a las diversas necesidades según al tipo que corresponda; sin embargo, se debe tener una orientación de mercado que permita apuntar hacia la mayor cantidad de personas que podrían hacer uso del sistema de seguridad.

Para ello se ha recabado información sobre proyectos emprendidos por el sector inmobiliario. En abril del 2018 en el Distrito Metropolitano de Quito se llevó a cabo la Feria MI CASA CLAVE, en la cual se ofertó alrededor de 542 proyectos inmobiliarios, presentados por 110 empresas [2].

En dicha feria se obtuvo indicadores sobre la distribución geográfica de los 542 proyectos obteniendo como resultado una mayor preferencia para la construcción en las zonas de Quito Norte, Valle de Los Chillos, Valles de Cumbayá y Tumbaco [2].

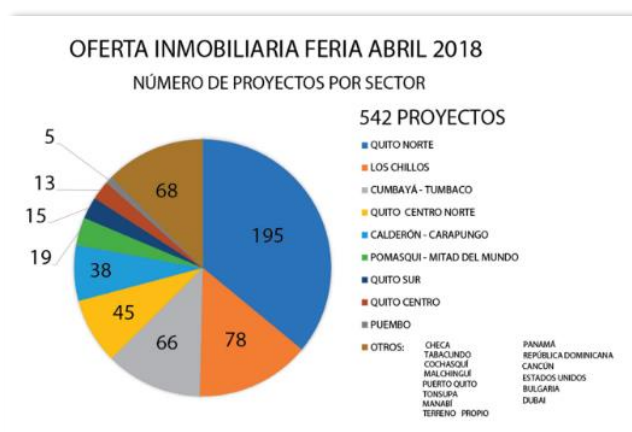


Figura 1. Proyectos inmobiliarios por sector
Feria MI CASA CLAVE abril 2018
Fuente [2]

Los proyectos que se presentaron incluían diferentes tipos de inmuebles, sin embargo, la mayor cantidad equivalente al 55% tuvieron oferta de departamentos. En la Figura 2 se presenta un indicador que refleja el número de proyectos según el tipo de producto inmobiliario ofertado.

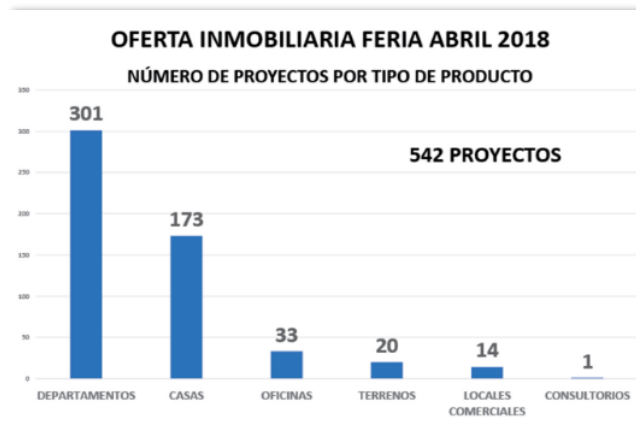


Figura 2. Número de proyectos por tipo de producto inmobiliario Feria MI CASA CLAVE abril 2018
Fuente [2]

En dicha feria también se obtuvo un índice de los diferentes rangos de precios a los que estaban sujetos los 542 proyectos ofertados, los mismos se muestran en la Figura 3.

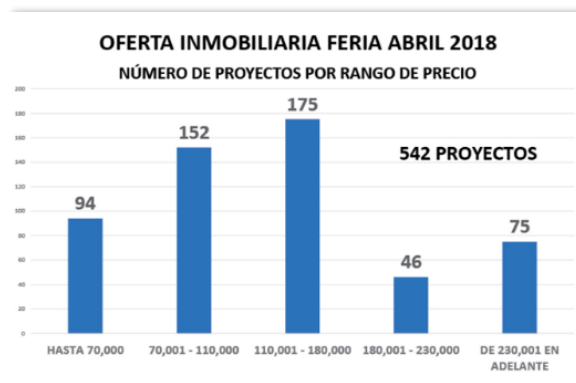


Figura 3. Número de proyectos por rango de precio Feria MI CASA CLAVE abril 2018
Fuente [2]

A su vez también se obtuvo el número de reservas realizadas por posibles compradores, en donde El Valle de Los Chillos, Quito - Norte y Calderón - Carapungo, son los sectores líderes en cuando a las apetencias del mercado, tal como se muestra en la Figura 2.



Figura 4. Número de inmuebles reservados por sector
Feria MI CASA CLAVE abril 2018
Fuente [2]

En base a las reservas realizadas se obtuvo un indicador que permitió determinar el tipo de inmueble de mayor preferencia por el potencial comprador, el mismo es presentado a continuación en la Figura 5.



Figura 5. Número de inmuebles reservados por tipo de producto
Feria MI CASA CLAVE abril 2018
Fuente [2]

El rango de precios de las unidades reservadas por tipo de producto se evidencia en la Figura 6.



Figura 6. Número de inmuebles reservados por rango de precio
Feria MI CASA CLAVE abril 2018
Fuente [2]

Dentro de la feria se realizó una encuesta al mercado comprador, con una muestra de 5.730 asistentes, hombres y mujeres mayores de edad, de diferentes segmentos económicos, todos con empleo. A ellos se les preguntó el tipo de inmueble de preferencia en conjunto con su sector, precio y tiempo estimado de compra. En las Figuras 7, 8, 9, 10 se puede apreciar los resultados de dicha encuesta, respectivamente.

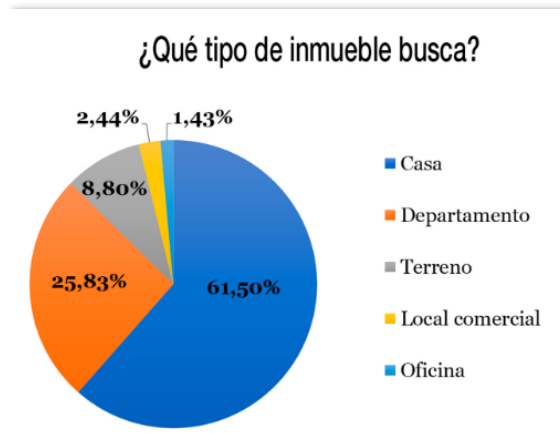


Figura 7. Preferencia de los asistentes por el tipo de inmueble Feria MI CASA CLAVE abril 2018
Fuente [2]

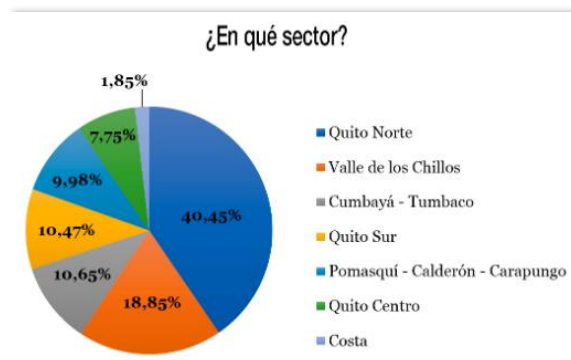


Figura 8. Preferencia de los asistentes por el sector del inmueble Feria MI CASA CLAVE abril 2018
Fuente [2]



Figura 9. Preferencia de los asistentes por el precio del inmueble Feria MI CASA CLAVE abril 2018
Fuente [2]

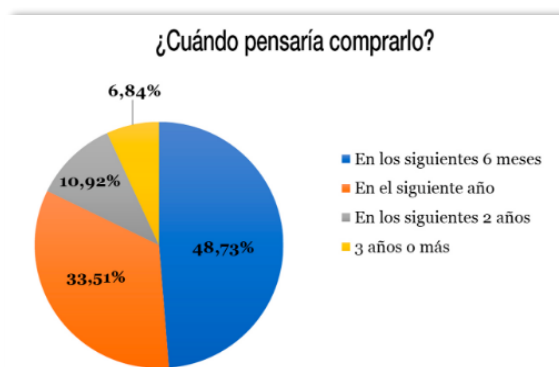


Figura 10. Tiempo estimado de compra de los inmuebles por parte de los asistentes Feria MI CASA CLAVE abril 2018
Fuente [2]

1.3.1 Análisis de resultados

Al presentar la problemática y el estudio de mercado con sus respectivos indicadores. Los resultados arrojan datos que permitirán llegar a consumidores finales claves para que los mismos hagan uso del sistema de seguridad descrito en el presente documento.

Frente a los resultados obtenidos se puede concluir que la preferencia de los consumidores de productos inmobiliarios en su mayoría prefiere el tipo de inmueble casa, con una ubicación en los sectores de *Quito-Norte* y *Los Valles*, con un precio fluctuante entre \$70.000 y \$120.000.

En base a los indicadores y resultados obtenidos se ha procedido a seleccionar un tipo de inmueble para la implantación del prototipo, que en este caso será de tipo casa. El mismo será construido en una maqueta que permita plasmar las características de uno de los diversos proyectos inmobiliarios ofertados en los sectores de *Quito-Norte* o *Los Valles*.

1.4 PROYECTO INMOBILIARIO

Frente a la preferencia de los sectores, se ha seleccionado un proyecto inmobiliario de tipo casa, ubicado en el sector de *Los Valles*.

1.4.1 Jardines Del Dean

Jardines del Dean, es un proyecto inmobiliario construido por Bellavista SC, en el sector del Dean Bajo (Valle de los Chillos). Está situado a 3 minutos del Triángulo de San Rafael [3].

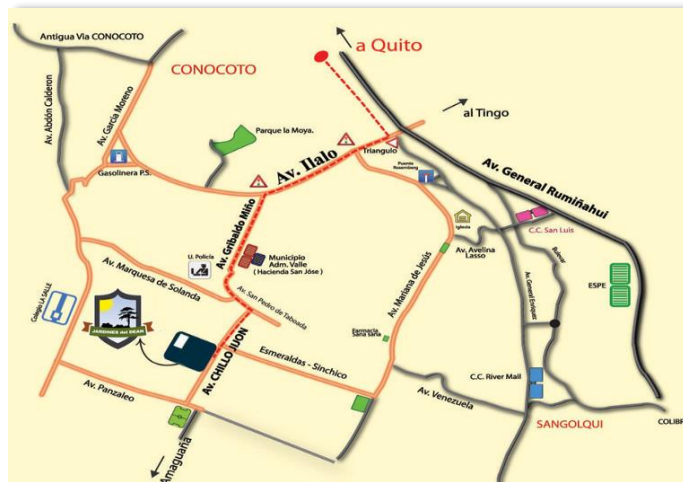


Figura 11. Mapa de acceso al proyecto inmobiliario Jardines del Dean
Fuente [3]

1.4.1.1 Principales ventajas del sector

Entre las principales ventajas del sector del proyecto inmobiliario se cuenta con [3]:

- Varias vías de acceso al conjunto desde Quito o el valle.
- Fácil acceso al transporte público, líneas de buses, servicio de taxi.
- Centros educativos cercanos: ESPE, Colegio La Salle.
- A pocos minutos de los principales centros comerciales del valle.

1.4.1.2 Principales ventajas del proyecto

- Gran sala comunal integrada al área verde.
- 16 estacionamientos para visitas.
- Área verde de gran tamaño tipo bosque.
- Casas pareadas con jardines propios muy amplios.
- Guardianía privada.
- Conjunto totalmente cerrado.
- Dos modelos diferentes de casas.
- Diseño nuevo, moderno y elegante.
- Salón comunal.
- Jardines.
- Área de juegos infantiles.



Figura 12. Vista aérea del proyecto inmobiliario
Jardines del Dean
Fuente [3]

1.4.1.3 Tipos de casas ofertadas por la inmobiliaria

Bellavista S.C oferta casas de dos y tres plantas. Para la implantación del prototipo se seleccionó el inmueble de dos plantas que cuenta con las siguientes características.

1.4.1.3.1 Descripción del inmueble de dos plantas

Área cubierta: 95 m²

Jardín propio total: desde 85 m² en adelante, incluido estacionamiento.

Planta Baja:

- Sala
- Comedor
- Baño Social
- Cocina tipo americana
- Estacionamiento lateral
- Casas pareadas todas esquineras
- Jardín y acceso
- Jardín lateral y posterior

Segunda Planta:

- Dormitorio máster con baño completo incluido y walk in closet.
- Dos dormitorios adicionales.
- Baño completo adicional.



Figura 13. Plano referencial del inmueble de dos plantas
Jardines del Dean
Fuente [3]

CAPÍTULO II

2. FUNDAMENTOS TEÓRICOS

2.1 DOMÓTICA

El avance tecnológico ha permitido revolucionar el mundo actual en el que los seres humanos se desenvuelven, se han creado nuevos conceptos, herramientas y productos que facilitan y dan soporte a las diversas tareas llevadas a cabo en el diario vivir. En este sentido “la tecnología aplicada al hogar”, denominada domótica, trae integrada consigo automatización, informática y nuevas tecnologías de comunicación; todas ellas dirigidas al confort, ocio, entretenimiento, seguridad y, en definitiva, la generación de un ambiente que consigo ofrezca bienestar dentro del hogar a sus miembros [4].

La domótica, también denominada, vivienda inteligente, vivienda del futuro, vivienda electrónica, hábitat integrado, hábitat interactivo, entre otros., puede ser definida como un conjunto de técnicas que emplean la electrónica, la informática y los automatismos industriales, con el objetivo de satisfacer las necesidades básicas de seguridad, confort, comunicación, gestión energética, del usuario y su entorno más cercano, haciendo uso de una red de comunicación que permite la interconexión de diversos dispositivos electrónicos presentes en el hogar [5] [6].

2.1.1 Tecnologías de la Información y Domótica

Dentro de la vivienda se han generado nuevas tendencias y aplicaciones gracias al uso de las *TIC (Tecnologías de la Información y las Comunicaciones)*. Las mismas, mediante el procesamiento, la administración y la compartición de la información han permitido generar una amplia gama de aplicaciones dentro de un hogar domotizado, entre las cuales destacan [7]:

- Seguridad
- Gestión de la energía
- Automatización de tareas domésticas
- Formación, cultura y entretenimiento
- Monitorización de salud
- Ocio y entretenimiento
- Operación y mantenimiento de las instalaciones

2.1.2 Sistema Domótico

Un sistema domótico básicamente está compuesto por una red de comunicaciones que permite la interconexión de una serie de equipos con el fin de obtener información sobre el entorno doméstico, y basándose en esta, realizar determinadas acciones sobre dicho entorno.

Los elementos de campo (detectores, sensores, captadores, etc.), presentes en un sistema domótico, tendrán la función de transmitir las señales del medio captadas a una unidad central inteligente, la misma que procesará la información recibida. En función de dicha información y en base a una programación establecida, la unidad central actuará en respuesta de dichas señales sobre determinados circuitos de potencia [7].

Dentro de las funciones que permite realizar un sistema domótico, se tiene:

- **Control remoto desde dentro de la vivienda** reduce la necesidad de moverse dentro de la vivienda, permitiendo ejercer tareas de forma sencilla. Hecho importante en el caso de personas de la tercera edad o discapacitadas.
- **Control remoto desde fuera de la vivienda** ejecución de tareas de forma remota, lo que supone un mejor aprovechamiento de tiempo al usuario.
- **Programabilidad** para ejecutar tareas de manera automatizada en base a una programación implantada en la unidad central, lo que supone un aumento del confort.

2.1.3 Gestión de la Domótica

La domótica a través de las distintas redes de comunicación, se encarga principalmente de gestionar cuatro aspectos del hogar, los mismo son expuestos a continuación.

2.1.3.1 Energía

En este campo la domótica se encarga de gestionar el consumo de energía eléctrica haciendo uso de temporizadores, termostatos, entre los principales. Esto con la finalidad de reducir el consumo excesivo de energía, traducándose a un ahorro económico y un mejoramiento de los recursos del medio ambiente. También se apoya de los recursos naturales, en especial de la energía solar y eólica, para las denominadas viviendas “bioclimáticas” [7].

2.1.3.2 Confort

La domótica aporta una serie de comodidades dentro del hogar, mediante esta se puede establecer un control automático de la iluminación, monitoreo del ambiente, control de puertas, ventanas, garaje, riego automático, entre otros.

Todas estas acciones se traducen a facilitar las tareas cotidianas de los miembros del hogar, permitiendo generar un ambiente confortable, ajustado a los deseos o necesidades de los mismos.

2.1.3.3 Seguridad

La seguridad es uno de los campos más importantes que nos brinda la domótica, la misma está dirigida tanto a los bienes como a los miembros que forman parte del hogar.

A continuación, se detalla los tres principales campos de acción de la domótica en cuanto a seguridad se refiere.

2.1.3.3.1 Seguridad de los bienes

En este ámbito se cuenta con simulación de presencia, alerta contra intrusos, control de acceso. Características que aportan al resguardo de los bienes materiales dentro del hogar.

2.1.3.3.2 Seguridad de las personas

Mediante el uso de controles o mandos a distancia tales como el teléfono móvil, se puede controlar y monitorear los eventos que pudieran suscitarse en el hogar. Con ello personas vulnerables de la tercera edad o discapacitados, podrán manejar situaciones y ejecutar tareas de manera sencilla y eficiente, permitiendo elevar el nivel de seguridad física frente a posibles incidentes.

2.1.3.3.3 Incidentes y averías

Mediante el uso de sensores se puede monitorear posibles incidentes tales como fugas de agua, gas, incendios o gases nocivos, de modo que, frente a dichos sucesos se activen circuitos o alarmas que contrarresten los incidentes que supondrían un peligro para la vida de los miembros del hogar.

2.1.3.4 Comunicaciones

Las comunicaciones son el pilar fundamental de la domótica. Estas permiten acceder a los diversos servicios que ofrecen las operadoras de telecomunicaciones, tales como las redes celulares, el internet, con el fin de interconectar los diversos elementos presentes en un sistema domótico.

2.2 TIPOS DE ARQUITECTURA

La arquitectura dentro de un sistema domótico, determina el modo en que los diversos componentes o elementos de control presentes en él, se conectarán o ubicarán. En seguida se presentan las arquitecturas más populares y usadas.

2.2.1 Arquitectura centralizada

Un controlador es el encargado de recibir las señales provenientes de los diversos sensores, y una vez procesadas, generar órdenes hacia los actuadores. En este tipo de arquitectura, el mal funcionamiento del controlador, derivará a que el sistema domótico no funcione en su totalidad [6].

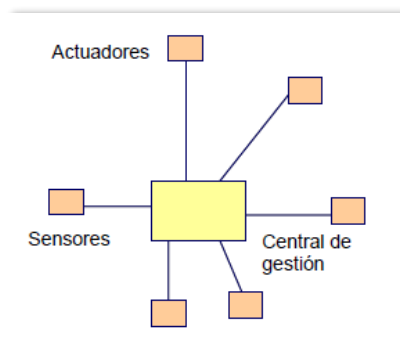


Figura 14. Arquitectura centralizada
Fuente [7]

2.2.2 Arquitectura distribuida

En la arquitectura distribuida no se maneja el concepto de controlador centralizado. A diferencia, toda la inteligencia del sistema se encuentra distribuida en sensores y actuadores que operan independientemente. En este sentido, algunas de las ventajas de contar con una arquitectura distribuida se enmarcan en que [7]:

- El mal funcionamiento de un elemento no impide el correcto funcionamiento del resto de elementos.
- Son fáciles de instalar.
- Son modulares y de fácil ampliación.

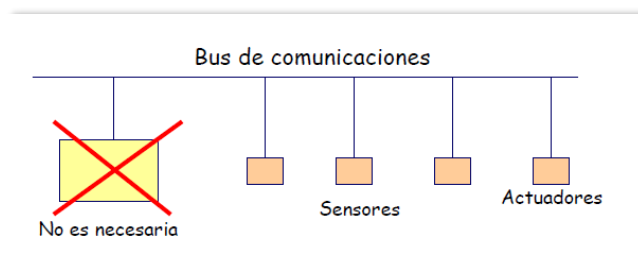


Figura 15. Arquitectura distribuida
Fuente [7]

2.3 SENSORES

Dentro de un sistema domótico, los sensores o también conocidos como transductores son un pilar fundamental, ya que permiten capturar información del entorno que lo rodea. Una vez la información es capturada, la misma es enviada a una unidad de control para su procesamiento y posterior toma de decisiones.

A continuación, se presentan diferentes tipos de sensores comúnmente usados en el desarrollo de soluciones domóticas.

2.3.1 Sensores de temperatura

Los sensores de temperatura son dispositivos electrónicos que transforman los cambios de temperatura en señales eléctricas. Por lo general un sensor de temperatura entrega al controlador la información en forma de señal analógica, aunque también es posible que su salida sea en forma digital mediante el uso de conversores analógicos/digital.

El uso de sensores de temperatura dentro de un sistema domótico, contribuye al monitoreo y automatización. Esto mediante la toma de decisiones por parte de la unidad central para la activación o desactivación de actuadores en base a las lecturas provista por el sensor.

2.3.2 Sensores de movimiento

Los detectores o sensores de movimiento también denominados *PIR (Passive Infrared)* o Pasivo Infrarrojo, son elementos electrónicos que se activan ante fuentes de energía tales como la temperatura corporal presente en seres humanos o animales. Reciben el nombre de pasivos, ya que dichos dispositivos no emiten radiaciones, sino que las reciben.

El modo en que se determina la presencia o movimiento de un individuo, radica en la diferencia emitida por el calor del cuerpo humano y el espacio que lo rodea [8].

Las aplicaciones de un sensor de movimiento son variadas, desde emitir un sonido de alarma, encender o apagar dispositivos luminosos hasta tomar fotografías cuando se detecte presencia humana en el entorno.

2.3.3 Sensores de calidad de aire

En este apartado existen gran variedad de sensores dedicados al monitoreo de calidad de aire presente en un determinado entorno. Estos sensores son capaces de obtener información del medio ambiente y entregar salidas tanto analógicas como digitales. Según el modelo en específico, existen sensores capaces de determinar la presencia de NH_3 (amoníaco), NO_x (óxidos de nitrógeno), CO_2 (dióxido de carbono), alcohol, benceno, humo, entre otros.

El uso de dichos sensores dentro de un sistema domótico, permite precautelar la integridad física de los miembros del hogar, mediante la generación de alarmas o la ejecución de acciones como la activación de un extractor, frente a una posible contaminación del aire.

2.3.4 Sensores magnéticos de apertura

Los sensores magnéticos de apertura generalmente son usados en puertas y ventanas, esto con el fin de determinar si alguien ajeno al inmueble intenta acceder a las instalaciones.

Un sensor magnético consta de dos partes para su funcionamiento. La primera se basa en un imán y la segunda en un encapsulado denominado *reed switch* (contiene un *switch* interno que se abre o se cierra). Estas dos partes en conjunto, funcionan como un interruptor tradicional, es decir, el circuito se abre o cierra frente a la presencia o ausencia del imán (cuando se abre o cierra una puerta o ventana).

Dependiendo del modelo, existen sensores *NA* (normalmente abiertos), *NC* (normalmente cerrados) y *C* (combinados *NA* y *NC*).

2.3.5 Sensores de luz

Conocidos también como sensores fotoeléctricos, fotocélulas o *LDR* (*Light-Dependent Resistor*). La función de una resistencia dependiente de luz, es aumentar o disminuir el paso de corriente frente a la presencia o ausencia de luz en el ambiente.

El uso de una *LDR* dentro de un sistema domótico, supone en la generación de ahorro energético y económico, esto debido a que puede ser usado como mecanismo para la activación (en la noche) o desactivación (en el día) automática de la iluminación interior o exterior del hogar.

2.3.6 Sensores de rotura de cristal

Muy usados en diferentes tipos de inmuebles, tales como locales comerciales y viviendas. El objetivo principal de uso del presente sensor consiste en la detección y alertas frente a posibles robos.

El modo de funcionamiento de un sensor de rotura de cristal está basado en una escucha continua de la frecuencia del ambiente. Cuando un cristal sufre una ruptura, este genera una frecuencia específica, que a su vez es captada por el sensor, para producir una respuesta frente a dicho evento.

2.3.7 Sensores de vibración sísmica

Este tipo de sensores por lo general son usados en locales comerciales, bancos o inmuebles que están sujetos a posibles incidentes de robo con explosivos u otro tipo de elementos que generen un alto grado de impacto.

Los sensores de vibración sísmica comúnmente se los relaciona con predicciones de terremotos, pero no es el caso, su función radica en la detección de vibraciones provenientes por fuentes tales como cortadores de diamantes para atravesar cristal, sopletes o lanzas térmicas para atravesar acero, hormigón, u otros.

2.3.8 Sensores de inundación

Los sensores de inundación son incorporados en lugares vulnerables a sufrir acumulaciones de agua como sótanos, garajes, lavanderías, con el fin de precautelar los bienes materiales alojados en dichas zonas.

2.4 ACTUADORES

Los actuadores también forman parte fundamental de un sistema domótico. La función principal de un actuador es responder a órdenes generadas por la unidad de control que repercuten en el mundo real. Por ello, a continuación, se exponen algunos de los actuadores más usados y populares en la implementación de soluciones domóticas.

2.4.1 Motores DC

Los motores *CC (corriente continua)* o *DC (corriente directa)*, son comúnmente usados en respuestas a eventos registrados en el mundo real. El modo de funcionamiento de un motor DC radica en la conversión de energía eléctrica en energía mecánica, lo que provoca un movimiento rotatorio, gracias a la acción de un campo magnético [9].

El uso de motores DC, puede ser aplicado a diversos entornos. Generalmente implementados en ventiladores, como ruedas de un carro a radio control o para apertura y cierre de persianas de un hogar.

2.4.2 Servomotores

Un servomotor es un elemento electrónico conformado por un motor DC, engranajes y una placa de control; todos estos elementos dentro de una carcasa de plástico. A diferencia de un motor DC tradicional, un servomotor tiene la capacidad de generar movimiento rotatorio de forma precisa. Esto quiere decir que puede adoptar posiciones generalmente entre 0° y 180° (aunque también pueden adoptar posiciones de hasta 360°).

Los servomotores son comúnmente usados en la robótica, automática y modelismo. Campos en los cuales se requiere precisión a la hora de generar movimiento rotatorio [10].

2.4.3 Cámaras

Las cámaras son otro dispositivo electrónico comúnmente usado como respuesta hacia un evento o incidente. Generalmente estos dispositivos dentro de un sistema domótico permiten el monitoreo del hogar en forma de video, o a su vez sirven como disparadores para capturar fotografías frente a una intrusión captada por un sensor de movimiento.

2.4.4 Relés

Un relé o relevador, es un dispositivo electromagnético, que funciona como interruptor, permitiendo o no el paso de corriente eléctrica. Está controlado por un circuito eléctrico, que por medio de una bobina o electroimán acciona un juego de uno o varios contactos.

Los relés por lo general son usados en circuitos de potencia, es decir funcionan como amplificadores que permiten accionar elementos que operan en voltajes de 110 VCA (*voltaje de corriente alterna*) [11].

2.4.5 Optoacopladores

Un optoacoplador al igual que un relé, permite la acción de dispositivos que operen en voltajes de 110 VCA. La diferencia radica en el funcionamiento, tamaño y no generación de ruido eléctrico.

El optoacoplador opera como un interruptor, al recibir una señal, está en su interior activa un diodo led que emite luz hacia un fototransistor, permitiendo de esta manera cerrar el circuito y generar una señal de salida.

2.5 INTERNET DE LAS COSAS

La innovación tecnológica ha permitido revolucionar los sectores de la electrónica, la informática y en general las TIC. Frente a esto, en cada uno de los sectores expuestos, se han creado nuevas tendencias y soluciones tecnológicas que han permitido generar hardware y software relativamente económico.

Se han desarrollado dispositivos electrónicos con un gran poder de procesamiento a un precio considerable y con un tamaño reducido, esto gracias al desarrollo en el campo de la microelectrónica y la nanotecnología.

Con estos antecedentes al contar con hardware de altas prestaciones, se han creado nuevos conceptos enmarcados en el ámbito tecnológico, es de esta manera que, desde

hace ya varios años, el término *Internet of Things (IoT)* o en español *Internet de las cosas*, o *Internet de los objetos (IO)*, ha generado una revolución y una evolución del Internet como hoy lo conocemos.

IoT, a diferencia del Internet tradicional, interconecta objetos del mundo real en la red. Esto supone un cambio sustancial, ya que se evoluciona de una súper red de ordenadores interconectados entre sí (Internet), a una súper red de objetos del mundo real interconexionados.

La interconexión radica no únicamente en teléfonos celulares, tablets o computadores, IoT va más allá de eso, desde ropas tecnológicas o también denominado *wereables (relojes, pulseras inteligentes, gafas de realidad aumentada)*, electrodomésticos (televisores, refrigeradores, aspiradoras, consolas de videojuegos, etc.), automóviles, hasta grandes infraestructuras como puentes, autopistas o ciudades, abriendo así un nuevo paradigma a la interacción *máquina-máquina (Machine-to-Machine, M2)* [12].

Al ofrecer un abanico de posibilidades IoT ha sido considerado como uno de los agentes de la cuarta revolución industrial (Industria 4.0), en conjunto con la inteligencia artificial, la robótica, la nanotecnología, la impresión 3D, 4D, entre otros.

En este sentido, IoT puede ser definido como la tecnología permisiva para el intercambio y procesamiento de información de objetos cotidianos del mundo real apoyándose en la red de redes (Internet) para su interconexión, ya sea de forma alámbrica o inalámbrica. Mediante esta premisa, el objetivo de IoT radica en brindar prestaciones a los usuarios finales, mediante la dotación de capacidades *inteligentes* a los objetos pasivos (o “*tontos*”) que nos rodean [13].

Frente a ello, es importante mencionar que, un objeto o dispositivo puede ser considerado *inteligente* cuando éste, al interactuar con el medio en el que “*habita*”, opera de forma interactiva o autónoma, permitiendo ejercer acciones en el mundo real en base a ordenes, patrones o señales captas, convirtiéndolo así en un ente capaz de tomar “*decisiones por si solo*”.

2.5.1 Internet para todo

En la actualidad el Internet forma parte de todas las actividades en las que se desarrolla el ser humano, desde ser usado para ocio, entreteniendo, compras o trabajo hasta como medio digital para la educación.

Hoy en día, el Internet es visto, no como una súper red de redes que interconecta varios ordenadores, sino que más bien es catalogado como un “*lugar*” en donde se dan “*cita*” los

seres humanos para ejecutar o realizar un abanico de posibilidades. Es de esta forma que surge la denominación del *Internet para todo*.

El abaratamiento y reducción de tamaño de componentes electrónicos ha permitido la evolución y desarrollo de las comunicaciones y de todos los sectores en sí. Frente a esto la cultura y el modo en que viven los seres humanos han cambiado en todos los ámbitos.

Por citar un ejemplo, en el pasado era muy común que las personas rentasen cintas o asistan al cine para poder observar una película. Hoy en día gracias al Internet, los usuarios pueden acceder a un catálogo completo desde la comodidad de su hogar.

Entre algunos de los sectores más importantes que se han visto involucrados en dicha evolución se tiene [14]:

- Medios de comunicación y periodismo
- Publicidad y marketing
- Música
- Televisión
- Lectura
- Banca
- Educación
- Ocio, cultura y viajes
- Seguridad del hogar y las personas.

2.5.2 Arquitectura IoT

Los componentes clave que forman parte de una arquitectura IoT, básicamente consisten en tres: *Dispositivos IoT*, *Redes* y *la Nube*, mismos que se presentan a continuación.

2.5.2.1 Dispositivos IoT

En este apartado se cuenta con elementos que pueden ser interconectados vía alámbrica o inalámbrica. Sensores, actuadores, centrales de procesamiento, son algunos de los múltiples elementos que forman parte de una red IoT. Estos tienen como finalidad interactuar con el medio que los rodea, de modo que permitan consumir o generar datos para realizar una tarea en particular, convirtiéndolos en objetos o dispositivos *inteligentes*, como bien se describió en la sección 2.5.

2.5.2.2 Redes

Las redes son las encargadas del transporte de la información generada por los dispositivos IoT hacia la nube. Radican en un componente sumamente importante dentro de una arquitectura IoT ya que sin ella no sería posible el intercambio de información.

2.5.2.3 Nube

La nube es uno de los nuevos términos introducidos al mundo tecnológico. Básicamente consiste en un conjunto de servidores que se encuentran distribuidos en una o diferentes partes del mundo.

La función de la nube dentro de una arquitectura IoT, se enmarca en la generación de prestaciones para el alojamiento de la información de manera segura y con la particularidad de poder ser accedida desde cualquier lugar geográfico, en donde el usuario se encuentre [15].

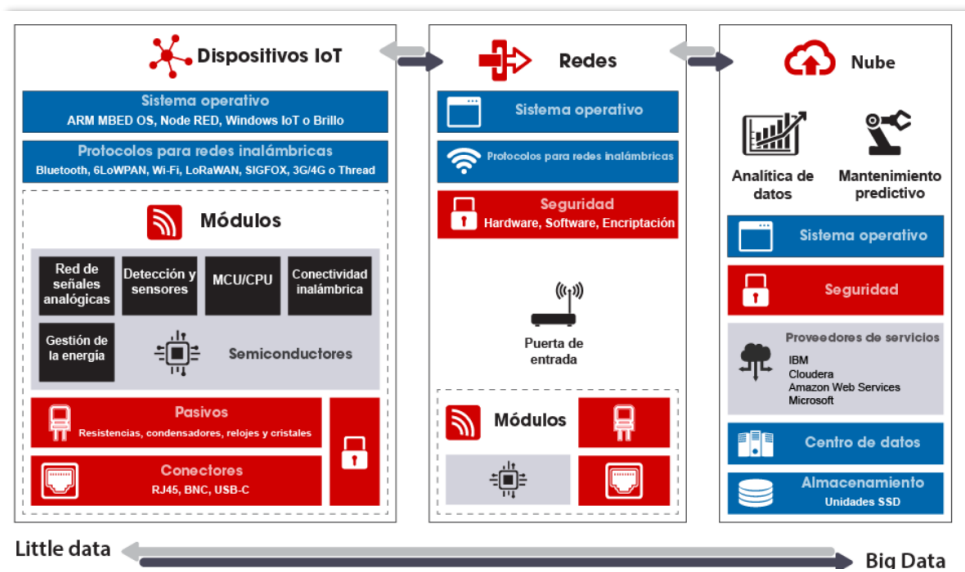


Figura 16. Arquitectura IoT en bloques
Fuente [15]

2.5.3 Aplicaciones IoT

Las aplicaciones de IoT son prácticamente ilimitadas, esto se debe a que pueden ser incorporadas y adaptadas a diferentes campos y/o sectores de la actividad humana, con la finalidad de proporcionar confort, facilidades y en general contribuir a una mejor calidad de vida.

2.5.3.1 Edificios inteligentes

Mejoras en la eficiencia, gestión y ahorro de la energía. Aplicaciones domóticas que mediante sensores y actuadores permiten monitorear, accionar y controlar electrodomésticos. Termostatos inteligentes. Servicios de cable/satélite. Apagado automático de la electrónica cuando no esté en uso. Detectores de humo y alarmas. Seguridad para todos los miembros de la familia [16].

2.5.3.2 Ciudades inteligentes y transporte

En este ámbito se tiene la optimización de transporte público y privado mediante la implementación de sensores de aparcamiento. Gestión inteligente de los servicios de estacionamiento y tráfico en tiempo real. Gestión inteligente de semáforos en función de las colas de tráfico. Administración del agua. Riego de parques y jardines. Contenedores de basura inteligentes. Controles de contaminación y movilidad. Monitoreo de accidentes y coordinación de acciones de emergencia [16].

2.5.3.3 Educación

Gestión, creación y vinculación de aulas virtuales y físicas para el aprendizaje, espacios virtuales (e-learning) más eficientes y accesibles. Servicios de acceso a bibliotecas virtuales y portales educativos. Aprendizaje permanente. Gestión de la asistencia [16].

2.5.3.4 Electrónica de consumo

Sólido posicionamiento de teléfonos, televisiones inteligentes. laptops, computadores y tablets. Refrigeradoras, lavadoras y secadoras inteligentes. Sistemas inteligentes de cine en casa [16].

2.5.3.5 Salud

Monitoreo de enfermedades crónicas. Mejora de la calidad de la atención y la calidad de vida de los pacientes. Trackers o rastreadores de actividad. Diagnóstico remoto. Cinturones interactivos. Deporte y monitoreo de actividades de fitness. Etiquetas inteligentes para fármacos. Monitoreo de hábitos alimenticios. Uso de biochips [16].

2.5.3.6 Agricultura y medio ambiente

Medición y control de la contaminación del medio ambiente (CO₂, ruido, elementos contaminantes presentes en el ambiente). Asociación de etiquetas RFID pasivas a los productos agrícolas. Gestión de residuos. Cálculos de nutrición [16].

2.5.3.7 Servicios de energía

Datos precisos sobre el consumo de energía. Mediciones y cálculos energéticos inteligentes. Redes inteligentes. Análisis y predicción de comportamientos de consumo de energía y patrones. Pronóstico de tendencias y necesidades futuras de energía [16].

2.5.3.8 Compras inteligentes

Inmersión de RFID y lectores electrónicos. Control de la procedencia geográfica de los alimentos y productos. Control de calidad de los alimentos y de la seguridad [16].

2.6 PLATAFORMAS IOT

Una plataforma IoT, es la base para que los diversos componentes que forman parte de una arquitectura IoT se interconecten y generen un ecosistema propio. Por lo general existen plataformas de dos tipos, de hardware y de software, mismas que se describen a continuación.

2.6.1 Plataformas Hardware

Para la ejecución y puesta en marcha de proyectos basado en IoT se cuenta con dos principales plataformas de hardware, relativamente económicas, sencillas de usar y de fácil adquisición. Estas dos plataformas son las conocidas placas electrónicas *Arduino* y *Raspberry Pi*.

2.6.1.1 Arduino






Arduino, consiste en una placa electrónica de hardware libre, está compuesto por un microcontrolador reprogramable (el cual varía en función de los diversos modelos) y un conjunto de pines hembra que permite la inserción de sensores y actuadores con el fin de que interactúen con el microcontrolador en cuestión [17].

2.6.1.1.1 Modelos de Arduino

Desde su lanzamiento hasta la presente fecha, la placa Arduino es una de las más utilizadas a nivel mundial, en diferentes sectores y ámbitos. Debido al gran uso y popularidad, se han desarrollado y fabricado varios modelos de placas Arduino. La diferencia radica en las características que cada una de estas ofrece (tamaño, número de pines E/S, modelo de microcontrolador) [17].

En la Tabla 1, se presenta los modelos de las diferentes tarjetas Arduino ofertadas en el mercado.

Tabla 1. Modelos de tarjetas electrónicas Arduino

MODELO	TARJETA
Arduino Nano	
Arduino Mega	
Arduino Leonardo	
Arduino Uno	
Arduino Leonardo	

2.6.1.1.2 Características de Arduino

Entre las principales características de la placa electrónica Arduino, se pueden destacar los siguientes aspectos: libre y extensible, amplia comunidad, multiplataforma, lenguaje de programación intuitivo y bajo coste.

2.6.1.1.2.1 Libre y extensible

Al ser una placa electrónica de hardware libre, Arduino es libre y extensible. Esto se traduce a que cualquier persona es libre de modificar, mejorar o ampliar el diseño del hardware.

2.6.1.1.2.2 Amplia comunidad

Arduino cuenta con una gran comunidad que brinda soporte a los diferentes modelos. Esto permite que exista una vasta documentación para comenzar a trabajar sobre cualquier plataforma.

2.6.1.1.2.3 Multiplataforma

Arduino puede ser instalado y manejado desde las diversas plataformas tradicionales como *Windows, Mac y Linux*.

2.6.1.1.2.4 Lenguaje de programación intuitivo

Arduino cuenta con su propio lenguaje de programación. El mismo está basado en C++, lo cual lo convierte en un lenguaje de fácil comprensión y con una curva de aprendizaje relativamente alta.

2.6.1.1.2.5 Bajo costo

El costo de las placas electrónicas es relativamente económico, la placa estándar (Arduino Uno) tiene un costo aproximado de \$10.

Incluso el usuario final podría construirla por sí mismo, ya que, al ser libre, existe todo tipo de documentación que contribuye a lograr dicho objetivo.

2.6.1.2 Raspberry Pi

Raspberry Pi, es otra alternativa a plataformas de hardware para implementación de proyectos basados en IoT, o de cualquier otra índole.

Raspberry Pi, es un computador de bajo coste y de tamaño reducido. Esta cuenta con periféricos de *E/S (entrada y salida)* de un computador tradicional, adicionalmente trae consigo un conjunto de pines denominados *GPIO (General Purpose Input/Output)*, los mismos que permiten la conexión de sensores y actuadores. Es capaz de realizar las mismas funciones de un computador de escritorio, desde navegar por internet, editar documentos de texto hasta ejecutar juegos de video.

Esta plataforma fue desarrollada con el propósito de que las personas tengan acceso y exploren diversos campos de la informática de una manera interactiva, aplicando el uso de dispositivos electrónicos en conjunto con lenguajes de programación como Python o Scratch [18].

Debido a su bajo coste y gran poder computacional, esta plataforma ha sido usada en diversos campos y sectores. Proyectos sujetos a ejemplos sobran, pero entre algunos de estos se tiene:

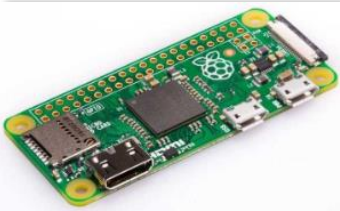



- Raspberry Pi como estación meteorológica [19]
- Raspberry Pi como estación de videojuegos [20]
- Raspberry Pi como centro multimedia [21]
- Raspberry Pi como chatbot [22]






2.6.1.2.1 Modelos de Raspberry Pi

Raspberry se ha establecido en el mercado como una solución tecnológica con un gran poder de procesamiento y de bajo coste. Por ende, se han fabricado gran variedad de modelos que se ajustan a las necesidades del usuario final.

A continuación, en la Tabla 2, se exponen algunos de los modelos con mayor demanda y usados en diversos sectores tecnológicos [23].

Tabla 2. Modelos de tarjetas Raspberry Pi

MODELO	TARJETA
Raspberry Pi Zero	
Raspberry Pi Zero W	
Raspberry Pi 1 A+	
Raspberry Pi 1 B+	

<p>Raspberry Pi 2 B</p>	
<p>Raspberry Pi 3 B</p>	
<p>Raspberry Pi 3 B+</p>	
<p>Raspberry Pi 3 A+</p>	
<p>Raspberry Pi 4 B</p>	

2.6.1.2.1.1 Raspberry Pi 3 B+

Raspberry Pi 3 B+, es la tercera generación de computadores de placa reducida. Es muy comercializada y usada a nivel mundial debido a su gran desempeño y bajo coste. Debido a dichas prestaciones, esta fue la placa seleccionada para el desarrollo del sistema de seguridad prototipo; por ello, a continuación, se presentan aspectos referentes a la misma, de modo que permita obtener un mayor contexto de las bondades, características y limitaciones que trae consigo.

2.6.1.2.1.1.1 Características técnicas

Entre las características técnicas de esta plataforma [24], se cuenta con lo expuesto en la Tabla 3.

Tabla 3. Características técnicas de Raspberry Pi 3 Modelo B+

CARACTERÍSTICAS TÉCNICAS RASPBERRY PI 3 B+	
CPU + GPU	Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz
RAM	1GB
WI-FI	2.4GHz y 5GHz IEEE 802.11.b/g/n/ac
BLUETOOTH	4.2 BLE
ETHERNET	Gigabyte Ethernet sobre USB 2.0 (300 Mbps)
GPIO	Cabecera extendida de 40 pines
HDMI	SI
USB	4 puertos 2.0
PUERTO CSI (CAMERA SERIAL INTERFACE)	SI (Permite conectar una cámara Raspberry Pi)
PUERTO DSI (DISPLAY SERIAL INTERFACE)	SI (Permite conectar una pantalla táctil Raspberry Pi)
PUERTO MICRO SD	SI (Para cargar el SO y almacenar datos)
PUERTO MICRO USB	SI (Para alimentar la placa 5V / 2.5 DC)
POE (POWER OVER ETHERNET)	SI

2.6.1.2.1.1.2 Pines de propósito general (GPIO)

El encabezado que trae consigo Raspberry Pi, es denominado *GPIO*. Dichos pines pueden ser configurados vía software, para actuar en forma de entrada o salida.

Desde el modelo Pi 1 B+, se cuenta con una cantidad considerable de pines, un total de 40, que remplazaron a un encabezado de 26 pines presente en versiones anteriores [25].

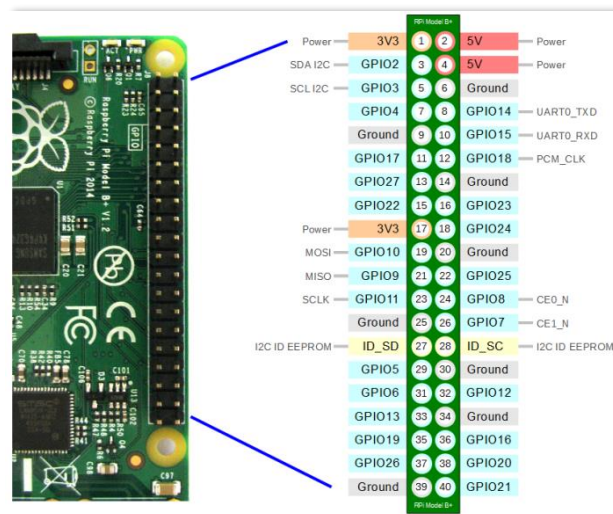


Figura 17. Pines de propósito general GPIO Raspberry Pi 3 B+

2.6.1.2.1.1.2.1 Voltajes

Dentro de los GPIO se puede encontrar pines no configurables, es decir, pines dedicados únicamente a entregar voltaje de corriente continua. GPIO brinda dos pines con salida de 5V, dos pines con salida de 3.3V y varios pines dedicados a la entrega de 0V (tierra).

2.6.1.2.1.1.2.2 Entradas

Los GPIO que son configurados como entradas, puede tomar dos valores: 0V (*low level*) y 3.3V (*high level*). Es necesario mencionar la importancia de uso de resistencias internas *pull-up* o *pull-down* (pueden ser activadas vía software) o resistencias externas, esto debido a que la Raspberry solo tolera voltajes en estado *high* de 3.3V y si por error se envía un estado de 5V, la misma puede sufrir daños.

2.6.1.2.1.1.2.3 Salidas

Un pin que es configurado como salida, puede entregar dos estados, 0V (*low level*) y 3.3V (*high level*).

2.6.1.2.1.1.2.4 GPIO especiales

Dentro de todos los GPIO presentes en el encabezado de Raspberry Pi, se cuenta con una variedad de pines que puede ejecutar funciones alternativas. Entre algunos de estos, se cuenta con:

- *PWM (Pulse-with modulation)*
- *SPI (Serial-peripheral interface)*
- *I2C (Inter-Integrated Circuit)*
- *Tx y Rx (Transmission and Reception)*

2.6.1.2.1.1.2.5 Nomenclatura

Para la lectura y numeración de los encabezados de Raspberry Pi existe gran cantidad de formas de hacerlo. Sin embargo, a continuación, se presenta las más comunes y recomendables.

2.6.1.2.1.1.2.5.1 Modo BOARD

El modo *Board* o también conocido como GPIO, es el más sencillo y recomendado, los pines se enumeran de forma física por el lugar que ocupan en la placa.

2.6.1.2.1.1.2.5.2 Modo BCM

El modo *BCM (Broadcom Mode)*, no es comúnmente usado, sin embargo, en este caso la numeración de los pines se la hace en base a la correspondencia con el chip *Broadcom (CPU de la Raspberry)*.

2.6.2 Plataformas Software

Las plataformas de software son un aspecto fundamental dentro de la puesta en marcha y ejecución de proyectos basados en IoT. Las plataformas de software tienen como finalidad la interconexión de los diversos objetos que forman parte del sistema. También ofrecen una gran variedad de prestaciones como, almacenamiento, procesamiento, alojamiento de recursos, entre otras.

Existe una gran variedad de plataformas de software que se han desarrollado y han madurado con el transcurso del tiempo y el avance de la tecnología. Entre las más conocidas y sencillas de usar se cuenta con:

- *Microsoft Azure*
- *AWS IoT*
- *Google Cloud*
- *Firebase*

2.6.2.1 Firebase

Es una plataforma ubicada en la nube, integrada con *Google Cloud*, que usa un conjunto de herramientas para la creación y sincronización de proyectos tanto web como móviles, con soporte para *Android, iOS, Javascript, Unity*, entre las principales. Firebase se caracteriza por la generación de soluciones de gran calidad y con una alta escalabilidad en el número de usuarios [26].

2.6.2.1.1 Servicios

Firebase es una plataforma que provee una gran variedad de soluciones como servicios, permitiendo hacer uso de su infraestructura con la integración de los *SDK (software development kit)* de una manera rápida y sencilla.

2.6.2.1.1.1 Cloud Firestore

Cloud Firestore es la evolución de *Firebase Real Time Database*, consiste en una base de datos *NoSQL* alojada en la nube a la que pueden acceder aplicaciones móviles y web mediante los SDK nativos de sus respectivas plataformas. Cloud Firestore al igual que

Firebase Real Time Database, proporcionan sincronización entre aplicaciones a través de agentes de escucha en tiempo real [27].

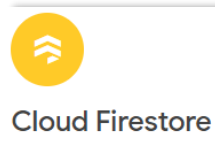


Figura 18. Cloud Firestore

2.6.2.1.1.2 Kit de AA

El Kit de AA (*Aprendizaje automático*) o Kit ML (*Machine Learning*), es un SDK desarrollado para las plataformas móviles Android y iOS, permite la incorporación de técnicas de inteligencia artificial de una manera práctica y sencilla, ya que cuenta con modelos listos para usar [28].

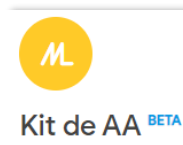


Figura 19. Kit de AA

2.6.2.1.1.3 Cloud Functions

Cloud Functions permite ejecutar de forma automática una o varias partes de código (funciones) por el lado del backend. Una vez la función se encuentra lista, los servidores de Google comienzan a administrarla, con ello ésta puede ser activada directamente desde una petición HTTP [29].

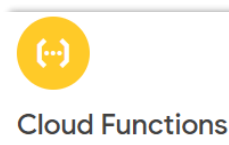


Figura 20. Cloud Functions

2.6.2.1.1.4 Authentication

Mediante los SDK desarrollados para *Firebase Authentication*, es posible administrar e identificar usuarios de las diversas plataformas. Permite también el almacenamiento de los datos de forma segura, basada en estándares como *OAuth 2.0* y *OpenID Connect*.

Permite la autenticación mediante el uso de contraseñas, números de teléfono y proveedores de identidad de terceros tales como: Facebook, Google, Twitter, entre otros [30].

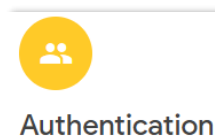


Figura 21. Firebase Authentication

2.6.2.1.1.5 Cloud Storage

Cloud Storage mediante sus SDK ofrece servicios para el almacenamiento de objetos de una manera simple y eficiente. Permite alojar contenido generado por los usuarios de aplicaciones tales como imágenes, audio, video y algunos otros tipos de contenido [31].

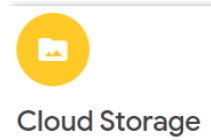


Figura 22. *Cloud Storage*

2.6.2.1.1.6 Cloud Messaging

Firebase Cloud Messaging (FCM), es una solución de mensajería instantánea, multiplataforma, que permite el envío de mensajes de forma gratuita y segura.

Mediante FCM se puede notificar a las aplicaciones de las diversas plataformas sobre eventos o cualquier otro aspecto que el desarrollador lo requiera de una manera práctica, sencilla y en tiempo real, ya sea a un dispositivo en específico, grupos de dispositivos o dispositivos suscritos a un tema [32].

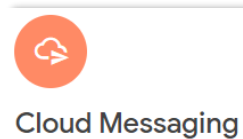


Figura 23. *Firebase Cloud Messaging*

2.7 SISTEMA OPERATIVO ANDROID

Android es un sistema operativo adquirido, desarrollado y mantenido por *Google*, inicialmente creado únicamente para dispositivos móviles, sin embargo, hoy en día es posible encontrarlo en televisores inteligentes (*Android TV*), *smartwatches (Wear OS)* y automóviles (*Android Auto*).

Está basado en el Kernel de Linux, con una plataforma abierta, lo que conlleva a que está puede estar sujeta a cambios, ajustes, mejoras, por parte de cualquier persona e incluso de la competencia [33].

2.7.1 Características

Android cuenta con un abanico de características que lo convierten en un sistema operativo atractivo, entre estas se puede destacar [33]:

- Plataforma totalmente libre que permite desarrollar aplicaciones y modificar las ya existentes con el lenguaje Java.

- Es multitarea, ya que permite ejecutar distintos procesos que corren al mismo tiempo.
- Compatible en su gran mayoría, con el hardware ofertado en el mercado, permitiendo elegir al usuario, el dispositivo que mejor se ajusta a sus necesidades.
- Cuenta con una biblioteca (Google Play) con una enorme cantidad de aplicaciones listas para descargar y usar.
- Actualmente es el sistema operativo más popular y usado a nivel mundial.


2.7.2 Versiones

Android, el sistema operativo más popular del medio, diseñado para diferentes plataformas, ha ido evolucionando y adecuándose tanto a las necesidades del usuario como a los diferentes dispositivos ofertados en el mercado. Dicha demanda ha permitido que el sistema operativo evolucione, generando así una gran cantidad de versiones con el pasar del tiempo.

En la Tabla 4 se presenta los diversos build versión del sistema operativo desarrollados por Google hasta la fecha [34].

Tabla 4. Versiones del sistema operativo Android

NOMBRE COMERCIAL	VERSIÓN	NIVEL DE API	FECHA DE LANZAMIENTO	LOGO
N. A	1.0	API Level 1	23 de septiembre de 2008	
Petit Four	1.1	API Level 2	09 de febrero de 2009	
Cupcake	1.5	API Level 3	27 de abril de 2009	
Donut	1.6	API Level 4	15 de septiembre de 2009	
Eclair	2.0	API Level 5	26 de octubre de 2009	
Eclair	2.0.1	API Level 6	4 de diciembre de 2009	
Eclair	2.1	API Level 7	12 de enero de 2010	
Froyo	2.2	API Level 8	20 de mayo de 2010	

Froyo	2.2.1	API Level 8	18 de enero de 2011	
Froyo	2.2.2	API Level 8	22 de enero de 2011	
Froyo	2.2.3	API Level 8	21 de noviembre de 2011	
Gingerbread	2.3	API Level 9	6 de diciembre de 2010	
Gingerbread	2.3.3	API Level 10	9 de febrero de 2011	
Gingerbread	2.3.4	API Level 10	28 de abril de 2011	
Gingerbread	2.3.5	API Level 10	26 de julio de 2011	
Gingerbread	2.3.6	API Level 10	2 de septiembre de 2011	
Gingerbread	2.3.7	API Level 10	21 de septiembre de 2011	
Honeycomb	3.0	API Level 11	22 de febrero de 2011	
Honeycomb	3.1	API Level 12	10 de mayo de 2011	
Honeycomb	3.2.x	API Level 13	20 de septiembre de 2011	
Ice Cream Sandwich	4.0.1 – 4.0.2	API Level 14	18 de octubre de 2011	
Ice Cream Sandwich	4.0.3 – 4.0.4	API Level 15	16 de diciembre de 2011	
Jelly Bean	4.1.x	API Level 16	9 de julio de 2012	
Jelly Bean	4.2.x	API Level 17	13 de noviembre de 2012	
Jelly Bean	4.3.x	API Level 18	24 de julio de 2013	
KitKat	4.4 – 4.4.4	API Level 19	3 de septiembre de 2013	

Lollipop	5.0	API Level 21	25 de junio de 2014	
Lollipop	5.1	API Level 22	09 de marzo de 2014	
Marshmallow	6.0	API Level 23	5 de octubre de 2015	
Nougat	7.0	API Level 24	9 de marzo de 2016	
Nougat	7.1	API Level 25	4 de octubre de 2016	
Oreo	8.0.0	API Level 26	21 de agosto de 2017	
Oreo	8.1.0	API Level 27	5 de diciembre de 2017	
Pie	9.0	API Level 28	6 de agosto de 2018	
Android10	10	API Level 29	3 de septiembre de 2019	android 10

2.7.3 Arquitectura

El sistema operativo Android está compuesto por cuatro capas: Kernel Linux, bibliotecas, marco de trabajo de aplicaciones y aplicaciones, mismas que se describen a continuación.

2.7.3.1 Kernel Linux

Android hace uso del Kernel de Linux para gestionar los servicios base como son la administración de memoria y procesos, pila de red, modelo de controladores y seguridad. Sirve también como intermediador para interactuar con el hardware y el resto de la pila de software. Todo este proceso permite acceder a los componentes sin necesidad de conocer el modelo o características precisas que tiene instalado cada dispositivo [35].

2.7.3.2 Bibliotecas

La segunda capa está compuesta por un conjunto de bibliotecas escritas en C/C++, las mismas son usadas por varios componentes del sistema. Todas estas bibliotecas se ofrecen al desarrollador a través del marco de trabajo de Android, entre estas se tiene [35]:

- Gestor de superficies (Surface Manager)
- SGL (Scalable Graphics Library)

- OpenGL ES (OpenGL for Embedded Systems)
- Bibliotecas multimedia
- WebKit
- SSL (Secure Socket Layer)
- FreeType
- SQLite
- Biblioteca C de sistema

Al mismo nivel de las bibliotecas se tiene el Entorno de Ejecución (Runtime Android), compuesta por:

- Bibliotecas Android
- Máquina virtual Dalvik

2.7.3.3 Marco de trabajo de aplicaciones

El marco de trabajo ha sido diseñado para reducir la carga en la reutilización de componentes de una aplicación. Los desarrolladores tienen acceso total a las APIs del Framework usados por las aplicaciones bases.

En esta capa, la mayor parte de componentes son bibliotecas Java que acceden a los recursos a través de la máquina virtual Dalvik. Entre las más importantes se cuenta con [35]:

- Administrador de actividades (Activity Manager)
- Administrador de ventanas (Windows Manager)
- Proveedor de contenidos (Content Provider)
- Vistas (Views)
- Administrador de notificaciones (Notification Manager)
- Administrador de paquetes (Package Manager)
- Administrador de telefonía (Telephony Manager)
- Administrador de recursos (Resource Manager)
- Administrador de ubicaciones (Location Manager)
- Administrador de sensores (Administration Manager)
- Cámara
- Multimedia

2.7.3.4 Aplicaciones

Aplicaciones base que trae consigo el sistema, incluyen navegador, cliente de correo electrónico, mensajería, mapas, contactos y uno de los más importantes el launcher, que tiene como tarea fundamental “lanzar” todas las aplicaciones instaladas en el sistema [35].

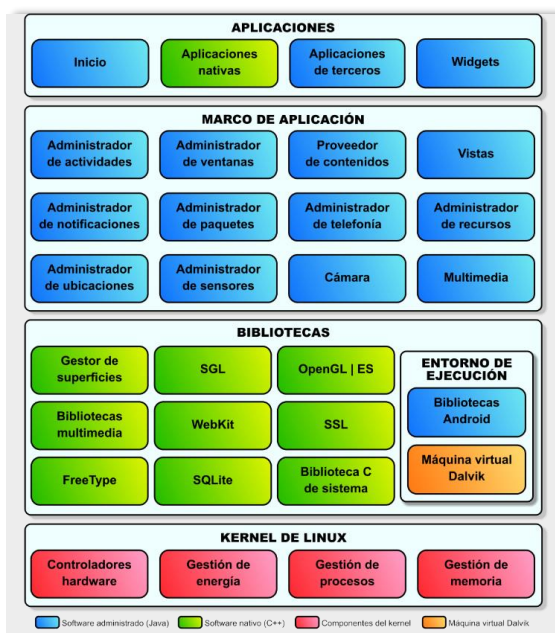


Figura 24. Pila de arquitectura del sistema operativo Android

2.8 METODOLOGÍA DE DESARROLLO DE SOFTWARE

Una metodología de desarrollo hace referencia a un framework o marco de trabajo para la creación de productos, servicios o software, convirtiéndola así en un marco sumamente adaptable para diversos campos y sectores.

El uso de una metodología para el desarrollo de software contribuye a elevar los estándares y el aseguramiento de éxito al momento de la finalización del proyecto en cuestión, esto debido a que mediante la ejecución de una secuencia ordenada de pasos permite estructurar, planear y controlar los diversos procesos que se llevan a cabo en el desarrollo del producto de software.

Dentro de las metodologías de desarrollo se cuenta con dos grandes grupos; las *tradicionales* y *ágiles*. La elección vendrá dada por las características del proyecto a llevar a cabo.

Las metodologías de desarrollo tradicionales por lo general son mucho más rígidas y estrictas que las ágiles, llevan consigo mecanismos que no dan soporte al despliegue de funcionalidad de forma continua, lo que se traduce en entregas sumamente extensas hacia el usuario final.

Por su parte las metodologías ágiles radican en todo lo contrario, son sumamente adaptables, permiten la entrega continua de funcionalidad, convirtiéndola en un marco de desarrollo muy usado en el mundo del software. Tal es el caso del marco de trabajo ágil Scrum.

Scrum es usado y puesto en práctica a nivel mundial. Un gran número de organizaciones dedicadas al desarrollo de software adoptan dicho marco debido a prestaciones tales como: flexibilidad, simplicidad y sobre todo a la adaptabilidad a cambios y entrega continua; factores que convierten a este marco de trabajo en el más idóneo para llevar a cabo el desarrollo e implementación del prototipo de seguridad que aquí se expone, puesto a que el mismo está sujeto tanto a cambios como entrega continua.

2.8.1 Marco de Trabajo Scrum

Es una de las metodologías ágiles más usadas en el desarrollo de software, caracterizada fundamentalmente por adoptar una estrategia de desarrollo incremental, entregando funcionalidad de manera continua, en periodos relativamente cortos denominados Sprints [36].

Scrum estaba basado en principios ágiles que permiten su adaptabilidad a diversos campos. Entre algunos de dichos principios se tiene [37]:

- Colaboración estrecha con el cliente.
- Predisposición y respuesta al cambio.
- Desarrollo incremental con entregas frecuentes de funcionalidad.
- Comunicación verbal directa.
- Simplicidad, solo con los artefactos necesarios.
- Motivación, compromiso y responsabilidad del equipo por la autogestión, auto organización.

2.8.1.1 Artefactos De Scrum

Scrum define una pequeña cantidad de artefactos que permiten asegurar el seguimiento del proyecto y el control de las actividades asociadas a cada sprint [37].

2.8.1.1.1 Product Backlog

El Product Backlog es un documento dinámico que incorpora un listado ordenado de los requisitos del producto a realizar [37].

2.8.1.1.2 Sprint Backlog

El Sprint Backlog es un listado de tareas que son extraídas del Product Backlog para ser ejecutadas dentro de un Sprint, las mismas que se convertirán en un incremento de funcionalidad [37].

2.8.1.1.3 Incremento

Un Incremento es el resultado de la ejecución de un Sprint.

2.8.1.2 Eventos de Scrum

Para la ejecución de las tareas dentro del marco de desarrollo Scrum, se cuenta con eventos o reuniones que permiten que se lleve a cabo un incremento mediante la planificación y el uso de bloques de tiempos relativamente cortos.

2.8.1.2.1 Sprint

Un Sprint es la base de Scrum. Un Sprint tiene un periodo de tiempo de alrededor de un mes, tres, dos o una semana, tiempo durante el cual se crea un incremento del producto [38].

2.8.1.2.2 Reunión de planificación del Sprint (Sprint Planning Meeting)

El Sprint Planning consiste en una reunión previa a la ejecución de un Sprint. El objetivo fundamental de esta reunión consiste en la planificación y la delegación de tareas para todos los miembros del equipo. Por lo general si el Sprint es de un mes, la reunión tiene un tiempo de duración de ocho horas, por otro lado, si el Sprint a ejecutarse es de menor tiempo, la duración de la reunión suele ser más corta [38].

2.8.1.2.3 Scrum diario (Daily Scrum)

Es una reunión que se lleva a cabo todos los días, en ella se expone las tareas que cada uno de los miembros del equipo realizaron el día previo a celebrarse el Daily Scrum, así también las tareas que llevarán a cabo el día que se ejecuta el Scrum diario y posibles problemas que atenten contra el progreso del Sprint Global.

Por lo general tiene una duración de 15 minutos, se lo hace de pie, a la misma hora y en el mismo lugar, de forma que resulte sencillo recordar para todos los miembros del Development Team [39].

2.8.1.2.4 Revisión del Sprint (Sprint Review)

Esta reunión es llevada a cabo al finalizar el Sprint, en ella el Product Owner revisa el Product Backlog y evalúa las tareas que pasaron a un estado "Done", con el objetivo de

priorizar los elementos sobre los cuales se trabajará en el próximo Sprint. Para un Sprint de un mes, la revisión tiene una duración de 4 horas [39].

2.8.1.2.5 Retrospectiva del Sprint (Sprint Retrospective)

La Retrospectiva del Sprint es ejecutada después de la Revisión del Sprint y antes de llevar a cabo el nuevo Sprint Planning. Este evento tiene la finalidad de que los miembros del equipo se inspeccionen así mismo en base a lo ocurrido en los eventos efectuados anteriormente, de modo que permitan mejorar en áreas que así lo requieran [39].

2.8.1.3 Roles en Scrum

Scrum se compone de tres perfiles o roles, que se describen a continuación:

2.8.1.3.1 Propietario del Producto (Product Owner)

El Product Owner es la persona encargada de relacionarse con los stakeholders, es aquella que nombra claramente los ítems del Product Backlog, así como también de dar la prioridad adecuada a los mismos.

Una vez que el Product Owner ha creado la lista de Backlog, es fundamental que esta transmita toda la información necesaria al Development Team, de modo que se asegure la comprensión de todos los miembros del equipo, para la ejecución de las diversas tareas a llevarse a cabo [40].

2.8.1.3.2 Equipo de Desarrollo (Development Team)

Compuesto por un conjunto de profesionales que tienen la responsabilidad de generar incrementos al final de cada Sprint. Es un equipo auto organizado y multifuncional, que trabaja en conjunto con el Scrum master para la generación de funcionalidad continua.

Para la definición de un tamaño óptimo de un Development Team se debe cumplir dos condiciones [40]:

- Debe ser lo suficientemente pequeño como para seguir siendo ágil.
- Debe ser lo suficientemente grande como para completar un trabajo significativo dentro de un Sprint.

2.8.1.3.3 Scrum Master

El Scrum Master es la persona encargada de promulgar el enfoque de Scrum, de modo que sea entendido por todos los miembros del equipo y estos lleven a cabo las prácticas y reglas marcadas. Se encarga también de generar apoyo tanto interna como externamente con la finalidad de maximizar el valor generado por el Scrum Team [40].

En la Figura 25 se presenta el modelo o flujo de trabajo a adoptar por una organización cuando se hace uso del marco de desarrollo con Scrum.

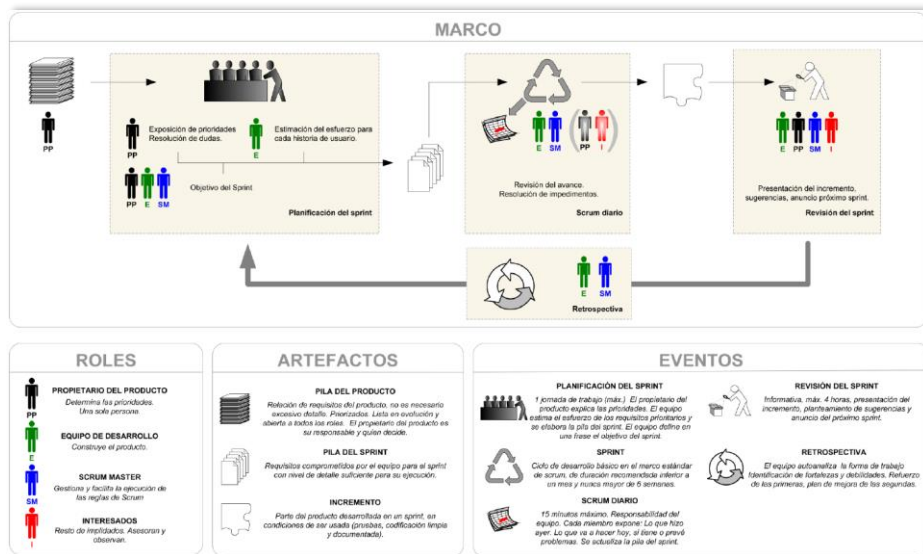


Figura 25. Marco de Desarrollo Scrum
Fuente [32]

CAPÍTULO III

3. DISEÑO DEL PROTOTIPO DE SEGURIDAD

En base a los antecedentes presentados en el Capítulo I, Sección 1.1, se procederá a la realización la definición del prototipo de seguridad que brinde confort, eleve el nivel de seguridad física tanto de las personas, así como también de los bienes materiales, traduciéndose esto a una mejor calidad de vida de los miembros del hogar.

Antes de iniciar con la etapa de diseño, es importante definir el tipo de inmueble donde será implementado el prototipo, así como sus características. Por ello en el Capítulo I, Sección 1.2 se efectuó un estudio de mercado con el fin de obtener una visión y panorama amplio que permitiese apuntar a un mercado objetivo acertado para la implementación del sistema de seguridad que aquí se presenta.

De este modo con base en el proyecto inmobiliario descrito en el Capítulo I, Sección 1.4, se ha elaborado un prototipo de vivienda con algunas modificaciones, esto con el fin de obtener una mejor apreciación del funcionamiento del sistema.

Tomando en cuenta los antecedentes expuestos, como punto de partida se elaborarán prototipos 2D y 3D, posteriormente se procederá con el levantamiento de requerimientos, para establecer el alcance de la solución.

3.1 PROTOTIPO DEL INMUEBLE

En este apartado se presenta el prototipo del inmueble tanto en 2D como en 3D, esto con la finalidad de generar una perspectiva de las zonas a ser cubiertas por el sistema de seguridad, permitiendo de esta manera generar un acertado levantamiento de requerimientos.

3.1.1 Prototipo 2D del inmueble

El presente prototipo de vivienda, está basado en el modelo de casas ofertado por el proyecto inmobiliario Jardines del Dean. La vivienda está modificada y adecuada en un diseño únicamente de un piso a diferencia del proyecto real que cuenta con dos. Esto se lo ha realizado con el fin de poder apreciar de mejor manera el funcionamiento del sistema.

La vivienda diseñada cuenta con áreas tradicionales de todo hogar, entre estas se tiene:

- Comedor
- Cocina
- Sala
- Baño
- Pasillo
- Habitación principal

- Habitación secundaria
- Área externa

A continuación, se presenta el diseño del inmueble prototipo en dos dimensiones (2D).

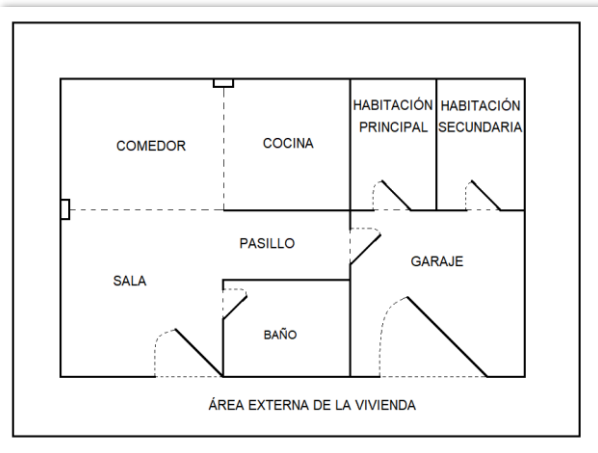


Figura 26. Modelo prototipo de la vivienda

3.1.2 Prototipo del inmueble en 3D

En esta sección se presenta el diseño del inmueble prototipo en tres dimensiones (3D).

3.1.2.1 Vista superior del inmueble prototipo

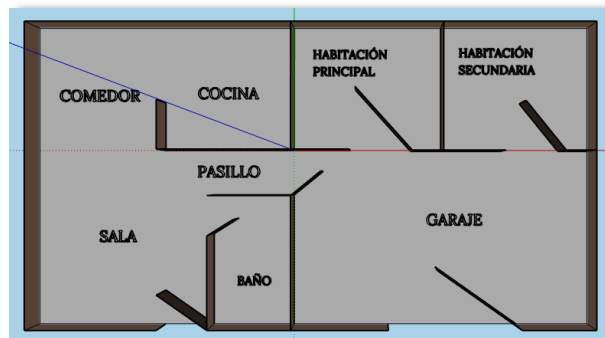


Figura 27. Vista superior del inmueble prototipo

3.1.2.2 Vista superior lateral derecha del inmueble prototipo

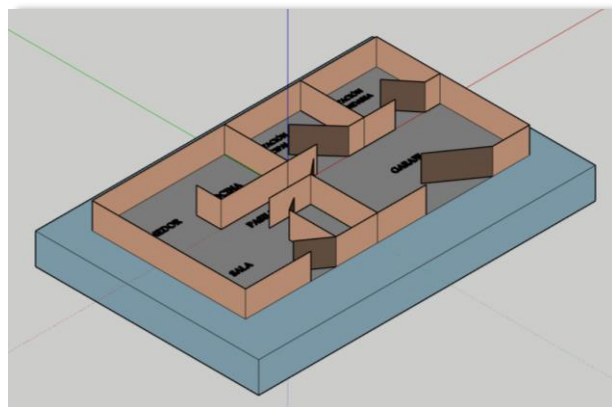


Figura 28. Vista superior lateral derecha

3.1.2.3 Vista superior lateral izquierda del inmueble prototipo

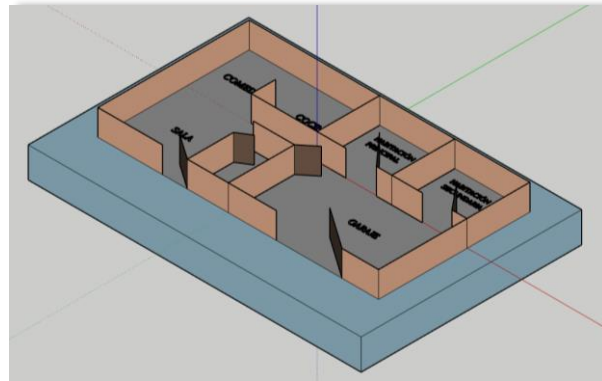


Figura 29. Vista superior lateral izquierda

3.2 ELICITACIÓN DE REQUERIMIENTOS

En base a características que den soporte al confort, mejoramiento de calidad de vida y el incremento de seguridad física hacia los miembros del hogar, se procederá a realizar la elicitación de requerimientos, misma permitirá definir las funcionalidades que ofrecerá el prototipo de seguridad a diseñarse.

En este sentido es importante mencionar que se incorporará la mayor cantidad de características y funcionalidades existentes en los diversos sistemas de seguridad ofertados en el mercado, teniendo como eje principal el obtener como resultado final un sistema innovador, robusto, de bajo coste y asequible para la mayor parte de sectores económicos.

Antes de efectuar la elicitación de requerimientos es importante definir los roles según el marco de trabajo Scrum como se muestra en la Tabla 5.

Tabla 5. Roles bajo el marco de desarrollo Scrum para el diseño y construcción del prototipo de seguridad

ROL	PERSONA
Propietario del Producto (Product Owner)	Dario Herrera
Scrum Master	MSc. Rodrigo Chancusig
Equipo de Desarrollo (Development Team)	Dario Herrera

3.2.1 Requerimientos iniciales

Los requerimientos iniciales que formarán parte de la pila de producto (Product Backlog) se los enlista a continuación:

- **R01.** Diseñar la arquitectura del prototipo de seguridad para control domótico.
- **R02.** Centralizar toda la información en un dispositivo móvil.
- **R03.** Gestión de usuarios para el acceso al aplicativo móvil.

- **R04.** Creación y autenticación de cuentas para el acceso a través de correo electrónico.
- **R05.** Creación y autenticación de cuentas para el acceso a través de un proveedor de acceso.
- **R06.** Confirmación de email antes de otorgar acceso al aplicativo móvil.
- **R07.** Opción de restablecimiento de contraseña para el acceso al aplicativo móvil.
- **R08.** Validación en campo email.
- **R09.** Validación y aplicación de una expresión regular al campo contraseña.
- **R10.** Validación y verificación en los campos de contraseña.
- **R11.** Control de iluminación a través del dispositivo móvil.
- **R12.** Control de iluminación a través de placa electrónica.
- **R13.** Sincronización de interruptores tanto digitales como físicos.
- **R14.** Automatización para la activación y desactivación de toda la iluminación.
- **R15.** Visualización de video en tiempo real.
- **R16.** Generación de un grado de libertad para rotar la cámara.
- **R17.** Automatización para la apertura y cierre de garaje.
- **R18.** Automatización para la activación y desactivación de un simulador de presencia.
- **R19.** Lectura y gráfica en tiempo real de la temperatura interior del hogar.
- **R20.** Automatización para el control de ventilador y calefactor.
- **R21.** Detección de movimiento en el hogar.
- **R22.** Envío automático de email con fotografía al detectar movimiento.
- **R23.** Posteo automático con fotografía en la red social Twitter al detectar movimiento.
- **R24.** Detección de apertura de puertas.
- **R25.** Control automático para la activación y desactivación del alumbrado público cuando las condiciones de luz natural así lo requieran.
- **R26.** Detección de humo y gases nocivos dentro del hogar.
- **R27.** Activación automática de extractor de aire y alarma sonora frente a la detección de gases nocivos.
- **R28.** Recepción de notificaciones push en tiempo real frente a incidentes dentro del hogar.
- **R29.** Historial de acciones e incidentes efectuados dentro del hogar.
- **R30.** Apartado de configuraciones dentro de la aplicación móvil para el control de sensores y dispositivos del hogar.

- **R31.** Acceso y control del sistema de seguridad a través del aplicativo móvil desde cualquier parte que el usuario se encuentre.
- **R32.** Implementación del sistema de seguridad en un prototipo de vivienda que simule las características de un hogar.

3.2.2 Pila de producto (Product Backlog)

Una vez obtenidos todos los requerimientos iniciales, estos pasarán a formar parte de la pila de producto, misma que contendrá una lista de tareas a ejecutarse durante todo el ciclo de vida del proyecto, clasificadas por grados de prioridad (Alta, Media, Baja).

Cada requerimiento será plasmado en una tabla individual con la finalidad de describir de mejor manera cada tarea a ejecutarse, obteniendo como resultado un conjunto de tablas pertenecientes al Product Backlog.

Tabla 6. R01_Diseñar la arquitectura del prototipo de seguridad para control domótico

IDENTIFICADOR	R01
Nombre	Diseñar la arquitectura del prototipo de seguridad para control domótico.
Descripción	Definir las tecnologías y herramientas de las cuales hará uso el prototipo de seguridad, tanto a nivel de software como hardware para la construcción del aplicativo móvil, base de datos y el manejo de sensores y actuadores respectivamente.
Prioridad	Alta

Tabla 7. R02_Centralizar toda la información en un dispositivo móvil

IDENTIFICADOR	R02
Nombre	Centralizar toda la información en un dispositivo móvil.
Descripción	El dispositivo contará con un aplicativo móvil multilenguaje que permitirá gestionar tareas y notificar sobre todos los eventos e incidentes que se produzcan dentro del hogar.
Prioridad	Alta

Tabla 8. R03_Gestión de usuarios para el acceso al aplicativo móvil

IDENTIFICADOR	R03
Nombre	Gestión de usuarios para el acceso al aplicativo móvil.
Descripción	El aplicativo móvil permitirá el acceso a las funciones únicamente a usuarios registrados en el sistema.
Prioridad	Alta

Tabla 9. R04_Creación y autenticación de cuentas para el acceso a través de correo electrónico

IDENTIFICADOR	R04
Nombre	Creación y autenticación de cuentas para el acceso a través de correo electrónico.
Descripción	El usuario podrá crear un perfil a través del uso de una cuenta personal de correo electrónico con el fin de poder autenticarse al tratar de ingresar al aplicativo.
Prioridad	Alta

Tabla 10. R05_Creación y autenticación de cuentas para el acceso a través de un proveedor de acceso

IDENTIFICADOR	R05
Nombre	Creación y autenticación de cuentas para el acceso a través de un proveedor de acceso.
Descripción	El usuario podrá crear un perfil de manera rápida y sencilla mediante el uso de un proveedor de acceso, en este caso una cuenta de Google. Con ello el usuario ahorrará tiempo al tratar de autenticarse al aplicativo.
Prioridad	Media

Tabla 11. R06_Confirmación de email antes de otorgar acceso al aplicativo móvil

IDENTIFICADOR	R06
Nombre	Confirmación de email antes de otorgar acceso al aplicativo móvil.
Descripción	Si el usuario opta por crear un perfil con una cuenta de correo electrónico personal, el sistema enviará de forma automática un email a la cuenta de correo electrónico proporcionada, con el fin de verificar que la información suscrita es válida. De no confirmar la recepción del email enviado, el usuario no podrá acceder a las funciones del sistema.
Prioridad	Alta

Tabla 12. R07_Opción de restablecimiento de contraseña para el acceso al aplicativo móvil

IDENTIFICADOR	R07
Nombre	Opción de restablecimiento de contraseña para el acceso al aplicativo móvil.
Descripción	Si el usuario opta por crear un perfil con una cuenta de correo electrónico personal y está ha sido verificada y validada, el sistema permitirá restablecer la contraseña proporcionada en caso de que está fuese olvidada.
Prioridad	Baja

Tabla 13. R08_ Validación en campo email

IDENTIFICADOR	R08
Nombre	Validación en campo email.
Descripción	Los campos para ingresar correos electrónicos serán validados para que admitan únicamente el formato de un email.
Prioridad	Baja

Tabla 14. R09_ Validación y aplicación de una expresión regular al campo contraseña

IDENTIFICADOR	R09
Nombre	Validación y aplicación de una expresión regular al campo contraseña.
Descripción	El campo para ingresar contraseña será validado para que únicamente acepte información que cumpla con la expresión regular establecida para la creación de contraseñas.
Prioridad	Baja

Tabla 15. R10_ Validación y verificación en los campos de contraseña

IDENTIFICADOR	R10
Nombre	Validación y verificación en los campos de contraseña.
Descripción	Al momento de la creación del perfil, el aplicativo solicitará que el usuario suministre una contraseña, misma que deberá coincidir con el campo de verificación. Si las contraseñas no coinciden el sistema no permitirá la creación de la cuenta.
Prioridad	Baja

Tabla 16. R11_ Control de iluminación a través del dispositivo móvil

IDENTIFICADOR	R11
Nombre	Control de iluminación a través del dispositivo móvil.
Descripción	Mediante el uso del aplicativo móvil, el usuario podrá activar y desactivar, la iluminación de las diferentes estancias del hogar.
Prioridad	Alta

Tabla 17. R12_ Control de iluminación a través de placa electrónica

IDENTIFICADOR	R12
Nombre	Control de iluminación a través de placa electrónica.
Descripción	-Mediante el uso de una placa electrónica para control de iluminación, el usuario podrá activar y desactivar, las lámparas de las diferentes estancias del hogar. -Mediante el uso de una placa electrónica de potencia para iluminación las lámparas se activarán y desactivarán según las órdenes efectuadas.
Prioridad	Alta

Tabla 18. R13_ Sincronización de interruptores tanto digitales como físicos

IDENTIFICADOR	R13
Nombre	Sincronización de interruptores tanto digitales como físicos.
Descripción	Tanto los interruptores de la interfaz del aplicativo móvil, como los interruptores físicos de la placa, deberán estar sincronizados, es decir, si desde el interruptor físico activo o desactivo una lámpara, la interfaz del aplicativo debe reflejar el estado de la misma y viceversa.
Prioridad	Alta

Tabla 19. R14_Automatización para la activación y desactivación de toda la iluminación

IDENTIFICADOR	R14
Nombre	Automatización para la activación y desactivación de toda la iluminación.
Descripción	Mediante el aplicativo móvil, el usuario podrá activar o desactivar todas las lámparas del hogar.
Prioridad	Baja

Tabla 20. R15_Visualización de video en tiempo real

IDENTIFICADOR	R15
Nombre	Visualización de video en tiempo real.
Descripción	-El aplicativo permitirá acceder a la visualización de video en tiempo real. -Si falla la visualización de video en tiempo, el aplicativo mostrará dicho estado en la interfaz. -Si la opción de video en tiempo real esta desactivada, el aplicativo mostrará dicho estado en la interfaz.
Prioridad	Alta

Tabla 21. R16_Generación de un grado de libertad para rotar la cámara

IDENTIFICADOR	R16
Nombre	Generación de un grado de libertad para rotar la cámara.
Descripción	Mediante el uso del aplicativo móvil, el usuario podrá rotar la cámara con un grado de libertad (izquierda, centro y derecha).
Prioridad	Media

Tabla 22. R17_Automatización para la apertura y cierre de garaje

IDENTIFICADOR	R17
Nombre	Automatización para la apertura y cierre de garaje.
Descripción	Mediante el uso del aplicativo móvil, el usuario podrá automatizar la apertura y cierre del garaje.
Prioridad	Baja

Tabla 23. R18_Automatización para la activación y desactivación de un simulador de presencia

IDENTIFICADOR	R18
Nombre	Automatización para la activación y desactivación de un simulador de presencia.
Descripción	Mediante el uso del aplicativo móvil, el usuario podrá elegir uno de los tres ambientes programados para activar y desactivar un simulador de presencia dentro del hogar.
Prioridad	Baja

Tabla 24. R19_Lectura y gráfica en tiempo real de la temperatura del hogar

IDENTIFICADOR	R19
Nombre	Lectura y gráfica en tiempo real de la temperatura del hogar.
Descripción	-El aplicativo móvil permitirá observar la temperatura interior del hogar en grados centígrados o fahrenheit (según configuración). -El aplicativo móvil permitirá generar, actualizar y borrar la gráfica de temperatura vs tiempo, basada en las lecturas recabadas.
Prioridad	Media

Tabla 25. R20_Automatización para el control de ventilador y calefactor

IDENTIFICADOR	R20
Nombre	Automatización para el control de ventilador y calefactor.
Descripción	El aplicativo móvil permitirá al usuario ajustar el nivel mínimo y máximo de temperatura para automatizar la activación o desactivación tanto del ventilador como del calefactor, según sea el caso.
Prioridad	Media

Tabla 26. R21_Detección de movimiento en el hogar

IDENTIFICADOR	R21
Nombre	Detección de movimiento en el hogar.
Descripción	El sistema de seguridad permitirá detectar movimiento mediante el uso de un sensor PIR.
Prioridad	Media

Tabla 27. R22_Envío automático de email con fotografía al detectar movimiento

IDENTIFICADOR	R22
Nombre	Envío automático de email con fotografía al detectar movimiento.
Descripción	Al detectar movimiento, el sistema de seguridad capturará una fotografía y enviará al usuario de forma automática un email de alerta con dicha información.
Prioridad	Alta

Tabla 28. R23_Posteo automático con fotografía en la red social Twitter al detectar movimiento

IDENTIFICADOR	R23
Nombre	Posteo automático con fotografía en la red social Twitter al detectar movimiento.
Descripción	Al detectar movimiento, el sistema de seguridad capturará una fotografía y posteará de manera automática en la red social Twitter un mensaje de alerta con dicha información.
Prioridad	Media

Tabla 29. R24_Detección de apertura de puertas

IDENTIFICADOR	R24
Nombre	Detección de apertura de puertas.
Descripción	-El sistema de seguridad permitirá la detección de apertura de puertas mediante el uso de un sensor magnético. -El sistema emitirá una alarma sonora en caso de detectar la apertura de una puerta.
Prioridad	Media

Tabla 30. R25_Control automático para la activación y desactivación del alumbrado externo

IDENTIFICADOR	R25
Nombre	Control automático para la activación y desactivación del alumbrado público cuando las condiciones de luz natural así lo requieran.
Descripción	El sistema de seguridad permitirá la automatización del alumbrado público mediante el uso de una LDR. La iluminación se activará o desactivará en base a las condiciones de luz natural.
Prioridad	Baja

Tabla 31. R26_Detección de humo y gases nocivos dentro del hogar

IDENTIFICADOR	R26
Nombre	Detección de humo y gases nocivos dentro del hogar.
Descripción	El sistema de seguridad permitirá la detección de humo y gases nocivos dentro del hogar mediante el uso del sensor MQ-135.
Prioridad	Media

Tabla 32. R27_Activación automática de extractor de aire y alarma sonora frente a la detección de gases

IDENTIFICADOR	R27
Nombre	Activación automática de extractor de aire y alarma sonora frente a la detección de gases nocivos.
Descripción	Al detectar humo o gases nocivos dentro del hogar, el sistema de seguridad activará de forma automática un extractor de aire y emitirá una alarma sonora.
Prioridad	Media

Tabla 33. R28_Recepción de notificaciones push en tiempo real frente a incidentes dentro del hogar

IDENTIFICADOR	R28
Nombre	Recepción de notificaciones push en tiempo real frente a incidentes dentro del hogar.
Descripción	El aplicativo móvil permitirá la recepción de notificaciones push en tiempo real frente a todos los eventos e incidentes que se generen dentro del hogar.
Prioridad	Alta

Tabla 34. R29_Historial de acciones e incidentes efectuados dentro del hogar

IDENTIFICADOR	R29
Nombre	Historial de acciones e incidentes efectuados dentro del hogar.
Descripción	-El aplicativo móvil contará con una opción que permitirá observar en tiempo real, un historial de todos los eventos e incidentes suscitados en el hogar. -El aplicativo móvil permitirá borrar el historial de acciones e incidentes suscitados en el hogar.
Prioridad	Alta

Tabla 35. R30_Apartado de configuraciones del aplicativo móvil

IDENTIFICADOR	R30
Nombre	Apartado de configuraciones dentro de la aplicación móvil para el control de sensores y dispositivos del hogar.
Descripción	El aplicativo móvil contará con una opción para configuración de diversos parámetros entre los cuales se tendrá: -Ver temperatura en grados Centígrados. -Ver temperatura en grados Fahrenheit. -Ajuste de temperatura para activación de ventilador. -Ajuste de temperatura para activación de calefactor. -Activación de detección de puertas. -Activación de video en tiempo real. -Activación de detección de movimiento.
Prioridad	Media

Tabla 36. R31_Control del sistema de seguridad a través del aplicativo móvil desde cualquier ubicación

IDENTIFICADOR	R31
Nombre	Acceso y control del sistema de seguridad a través del aplicativo móvil desde cualquier parte que el usuario se encuentre.
Descripción	El usuario podrá acceder a todas las funcionalidades del aplicativo móvil y del sistema de seguridad desde cualquier parte que se encuentre, los componentes de hardware y software deberán estar conectados a una red de datos celular o un punto de acceso Wi-Fi para lograr dicha premisa.
Prioridad	Alta

Tabla 37. R32_Implementación del sistema de seguridad en un prototipo de vivienda

IDENTIFICADOR	R32
Nombre	Implementación del sistema de seguridad en un prototipo de vivienda que simule las características de un hogar.
Descripción	El sistema de seguridad será implementado en un prototipo de vivienda que permitirá simular las características de un hogar.
Prioridad	Baja

3.3 PLANIFICACIÓN DE SPRINTS

Una vez finalizado el Product Backlog, es necesario continuar con el siguiente paso bajo el marco de desarrollo con Scrum, mismo que se basa en la ejecución del Sprint Planning Meeting o Reunión de Planificación de Sprint. Esta reunión tiene como objetivo fundamental la planificación y elección de tareas que formarán parte del Sprint Backlog a desarrollarse en cada Sprint activó en cuestión.

Cada Sprint cuenta con una estimación denominada Story Points, este indicador permite reflejar el esfuerzo o complejidad del incremento o Sprint que se lleva a cabo en cada iteración. Para la estimación de Story Points de cada tarea se hizo uso de una técnica ágil denominada Planning Poker.

Planning Poker consiste en una técnica comúnmente usada en los Sprint Planning Meeting, la misma está basada en la participación activa de todos los miembros del Scrum Team. Planning Poker hace uso de un conjunto de cartas con la particularidad de que estas están enumeradas bajo la serie de Fibonacci (0, 1, 2, 3, 5, 8, 13) en donde se incluye el signo de interrogación (?) para representar la falta de información para estimar la tarea y el símbolo infinito (∞) para indicar que la tarea es demasiada grande.

El modo de operación de la presente técnica se basa en la dinámica presentada en [41]:

- Todo el Team Scrum asiste al Sprint Planning Meeting.
- Una vez que todos los involucrados estén presentes en la reunión, el Scrum Master presenta y brinda una explicación general de cada tarea a realizarse en el Sprint a efectuarse. De ser necesario el Scrum Master brindará mayor información si alguna parte de la tarea no fue comprendida.
- Una vez que las tareas están claras, el Scrum Master pide únicamente a cada miembro del Team Develop que elija una carta. Una vez seleccionada la carta se realiza una cuenta regresiva en donde todos mostrarán al mismo tiempo su elección.
- En caso de existir estimaciones que supongan un valor relativamente alto en comparación con el resto, el Scrum Master deberá hacer exponer a los involucrados, las razones que lo llevaron a tomar dicha decisión.
- Una vez expuesto los puntos de vista de los involucrados, se realiza nuevamente la votación.
- Finalmente, el Scrum Master asigna la media de estimaciones denominada Story Point a la tarea en cuestión.

Por cada iteración o Sprint efectuado, se llevó a cabo un Sprint Planning Meeting para la planificación de tareas, mismas que fueron obtenidas de la pila de pendientes (Sprint Backlog) para su posterior estimación de Story Points.

Para la realización del prototipo de seguridad para control domótico se requirió un total de 13 Sprints, con una duración de 10 días cada uno.

Una vez finalizado cada sprint, de acuerdo con SCRUM se debe realizar la presentación del avance funcional. Esto mediante los eventos *Sprint Review* y *Retrospective* que son descritos en el siguiente capítulo a lo largo de la descripción de la construcción del sistema.

3.3.1 Sprint 1

El primer Sprint contó con una duración de 10 días, inició el 01/07/2019 y finalizó el 12/07/2019, la tarea a llevarse a cabo en este Sprint fue R01, misma que se basa en el diseño de la arquitectura para el prototipo de seguridad para control domótico.

R01 es una tarea de prioridad alta y de suma importancia, pues supone la base fundamental sobre la cual reposará el sistema propuesto. En la Tabla 38 se presentan las subtareas a efectuarse conjuntamente con los respectivos Story Points.

3.3.1.1 Objetivo del Sprint 1

El objetivo fundamental que persigue el primer Sprint está basado en el diseño de la arquitectura del sistema de seguridad, específicamente definir las tecnologías, lenguajes, herramientas, hardware y modo de comunicación de los diversos componentes que de soporte a las características y requerimientos definidos con anterioridad.

Tabla 38. Tareas a efectuarse en Sprint 1

REQUERIMIENTO	SUBTAREAS	STORY POINTS
R01 - Diseñar la arquitectura del prototipo de seguridad para control domótico.	1. Elección y estudio de la plataforma software que brinde soporte al alojamiento de datos en tiempo real.	8
	2. Elección y estudio de la plataforma hardware que brinde soporte al manejo de sensores y actuadores.	8
	3. Elección y estudio del lenguaje de programación a ser usado para el desarrollo del aplicativo móvil.	8
	4. Definición y elaboración de diseño del modo de comunicación de los componentes hardware y software.	8
	5. Diseño e implementación de la estructura NoSQL para la base de datos.	8
TOTAL, STORY POINTS SPRINT 1		40

3.3.2 Sprint 2

El segundo Sprint contó con una duración de 10 días, inició el 15/07/2019 y finalizó el 26/07/2019, las tareas que se efectuaron fueron R02, R03 y R04. Todas con un grado de prioridad Alta. En la Tabla 39 se presentan las subtareas a efectuarse conjuntamente con los respectivos Story Points.

3.3.2.1 Objetivo del Sprint 2

El objetivo del Sprint 2 tiene como finalidad la integración del proyecto con la plataforma software Firebase, así como también la construcción e implementación de las interfaces que permitirán al usuario el inicio de sesión y creación de cuentas con una dirección de correo electrónico. Todo esto con la finalidad de generar un sistema robusto que únicamente permita el acceso a usuarios registrados en el sistema.

Tabla 39. Tareas a efectuarse en Sprint 2

REQUERIMIENTO	SUBTAREAS	STORY POINTS
R02 - Centralizar la información en un dispositivo móvil.	1. Creación del proyecto móvil en Android Studio.	3
	2. Creación del archivo para internacionalización del aplicativo.	3
	3. Creación e integración del proyecto móvil con los SDK de Firebase	5
R04 - Creación y autenticación de cuentas para el acceso a través de correo electrónico.	1. Diseño, integración con Firebase Authentication e implementación de la interfaz para iniciar sesión.	8
	2. Diseño, integración con Firebase Authentication e implementación de la interfaz para registrarse con una cuenta de correo electrónico.	8
R03 - Gestión de usuarios para el acceso al aplicativo móvil.	1. Solo usuarios registrados podrán ingresar al aplicativo.	3
TOTAL, STORY POINTS SPRINT 2		30

3.3.3 Sprint 3

El tercer Sprint contó con una duración de 10 días, inició el 29/07/2019 y finalizó el 09/08/2019, las tareas que se efectuaron fueron R05, R06 y R07. Tienen un grado de prioridad Alta, Media y Baja, respectivamente. Fueron seleccionadas con el fin de aportar a la finalización de la implementación de todo el apartado de autenticación al tratar de ingresar al aplicativo. En la Tabla 40 se presentan las subtareas a efectuarse conjuntamente con los respectivos Story Points.

3.3.3.1 Objetivo del Sprint 3

En el Sprint 3 el objetivo consistió en tareas que aporten a la finalización de todos los apartados relacionados con la seguridad y autenticación de usuarios al momento de iniciar sesión. Se generará la opción para autenticarse mediante un proveedor de acceso (Google), confirmación de email para acceder al aplicativo y opción para restablecimiento de contraseña. Esto con la finalidad de crear una base sólida sobre la cual se generará el resto de funcionalidades.

Tabla 40. Tareas a efectuarse en Sprint 3

REQUERIMIENTO	SUBTAREAS	STORY POINTS
R05 - Creación y autenticación de cuentas para el acceso a través de un proveedor de acceso.	1. Diseño, integración con Firebase Authentication e implementación de la opción para autenticarse con una cuenta de Google.	5
R06 - Confirmación de email antes de otorgar acceso al aplicativo móvil.	1. Implementación para generar el envío de confirmación de email a la hora de crear una cuenta con una dirección de correo electrónico por parte del usuario.	5
R07 - Opción de restablecimiento de contraseña para el acceso al aplicativo móvil.	1. Diseño e implementación de la interfaz que permitirá al usuario restablecer contraseña en caso de que esta haya sido olvidada.	8
TOTAL, STORY POINTS SPRINT 3		18

3.3.4 Sprint 4

El cuarto Sprint contó con una duración de 10 días, inició el 12/08/2019 y finalizó el 23/08/2019, las tareas que se efectuaron fueron R08, R09, R10 y R11. Todas en su mayoría con un grado de prioridad Media y Baja. En la Tabla 41 se presentan las subtareas a efectuarse conjuntamente con los respectivos Story Points.

3.3.4.1 Objetivo del Sprint 4

El Sprint 4 tiene como objetivo la finalización de todas las tareas relacionadas a la seguridad y autenticación de usuarios. Con la culminación del presente Sprint se genera un entregable robusto y totalmente funcional, de este modo el usuario podrá acceder y mantener toda la información de forma segura.

Tabla 41. Tareas a efectuarse en Sprint 4

REQUERIMIENTO	SUBTAREAS	STORY POINTS
R08 - Validación en campo email.	1. Validación en tiempo real para los campos de tipo email. El usuario solo podrá ingresar datos que cumplan con el formato de un email.	3
R09 - Validación y aplicación de patrón en campo contraseña.	1. Validación en tiempo real para que el aplicativo móvil solo acepte contraseñas con una longitud mínima de 8 caracteres, 1 minúscula, 1 mayúscula y un carácter especial.	3
R10 - Validación y verificación en los campos de contraseña.	1. Validación en tiempo real para asegurar la coincidencia de contraseñas antes de su asignación.	3
R11 - Control de iluminación a través del dispositivo móvil.	1. Diseño e implementación del componente Navigation Drawer.	5
	2. Diseño, conexión con base de datos e implementación de la interfaz del fragmento que controlará la iluminación del hogar.	8
	3. Integración de la placa electrónica Raspberry Pi mediante Python a los SDK de Firebase.	5
	4. Conexión con base de datos e implementación del script en Python que permitirá el control del hardware de iluminación.	5
	5. Diseño del circuito electrónico que permitirá activar las lámparas del hogar.	8
	6. Construcción del circuito electrónico para iluminación del hogar en una tarjeta de circuito impreso (PCB).	10
TOTAL, STORY POINTS SPRINT 4		50

3.3.5 Sprint 5

El quinto Sprint contó con una duración de 10 días, inició el 26/08/2019 y finalizó el 06/09/2019, las tareas que se efectuaron fueron R12 y R13. Todas con un grado de prioridad Alta. En la Tabla 42 se presentan las subtareas a efectuarse conjuntamente con los respectivos Story Points.

3.3.5.1 Objetivo del Sprint 5

En el Sprint 5 el objetivo que se desea alcanzar consiste en controlar la iluminación no solo haciendo uso del aplicativo, sino que también se desea poder controlar la iluminación con

interruptores físicos. Bajo esta premisa se pretende que tanto la interfaz para el encendido y apagado de lámparas esté sincronizada con la placa de control.

Tabla 42. Tareas a efectuarse en Sprint 5

REQUERIMIENTO	SUBTAREAS	STORY POINTS
R12 - Control de iluminación a través de placa electrónica.	1. Implementación del script en Python que permitirá controlar los interruptores para activar o desactivar las lámparas del hogar.	10
	2. Diseño del circuito electrónico que permitirá a los interruptores activar o desactivar la iluminación del hogar.	8
	3. Construcción del circuito electrónico que permitirá a los interruptores activar o desactivar la iluminación del hogar en una tarjeta de circuito impreso (PCB).	10
R13 - Sincronización de interruptores tanto digitales como físicos.	1. Implementación de la lógica en el aplicativo móvil para que la interfaz se sincronice frente a los cambios efectuados desde la placa física de control.	8
	2. Implementación de la lógica en el script de Python para que se sincronice frente a los cambios efectuados desde el aplicativo móvil.	10
TOTAL, STORY POINTS SPRINT 5		46

3.3.6 Sprint 6

El sexto Sprint contó con una duración de 10 días, inició el 09/09/2019 y finalizó el 20/09/2019, las tareas que se efectuaron fueron R15 y R16. Con un grado de prioridad Alta y Media respectivamente. En la Tabla 43 se presentan las subtarear a efectuarse conjuntamente con los respectivos Story Points.

3.3.6.1 Objetivo del Sprint 6

En el Sprint 6 el objetivo se basó en la recepción de video en tiempo real desde el aplicativo móvil, así como también la generación de un grado de libertad para rotar la cámara que se apoya sobre un servomotor. Este grado de libertad se implementó con la finalidad de que la cámara pueda captar la mayor parte de zonas del hogar.

Tabla 43. Tareas a efectuarse en Sprint 6

REQUERIMIENTO	SUBTAREAS	STORY POINTS
R15 - Visualización de video en tiempo real.	1. Diseño e implementación de la interfaz del fragmento que permitirá acceder a la opción de video en tiempo real.	10
	2. Investigación acerca de video streaming server.	3
	3. Investigación acerca de video streaming con Flask.	3
	4. Instalación de Flask en Raspberry Pi e implementación de scripts en Python para video streaming.	10
R16 - Generación de un grado de libertad para rotar la cámara.	1. Impresión de un modelo 3D para soporte de cámara para Raspberry Pi.	3
	2. Acople de servomotor al modelo impreso en 3D.	5
	3. Diseño de circuito electrónico para control de servomotor.	3
	4. Implementación del script en Python para controlar el movimiento del servomotor.	8
TOTAL, STORY POINTS SPRINT 6		45

3.3.7 Sprint 7

El séptimo Sprint contó con una duración de 10 días, inició el 23/09/2019 y finalizó el 04/10/2019, las tareas que se efectuaron fueron R19 y R21, con un grado de prioridad Media. En la Tabla 44 se presentan las subtareas a efectuarse conjuntamente con los respectivos Story Points.

3.3.7.1 Objetivo del Sprint 7

El objetivo del Sprint 7 está enmarcado en tareas relacionadas a la lectura de temperatura del hogar y la detección de movimiento. La información de temperatura será presentada en el aplicativo móvil en tiempo real y además generará una gráfica sobre como varia la temperatura en el transcurso del tiempo. Finalmente se deberá sensor la detección de movimiento dentro del hogar.

Tabla 44. Tareas a efectuarse en Sprint 7

REQUERIMIENTO	SUBTAREAS	STORY POINTS
R19 - Lectura y gráfica en tiempo real de la temperatura interior del hogar.	1. Diseño e implementación de la interfaz del fragmento que permitirá observar la temperatura del hogar en tiempo real.	8
	2. Diseño e implementación de la interfaz que permitirá graficar la temperatura que se registra en el hogar.	10
	3. Diseño de circuito electrónico para lectura de temperatura.	2
	4. Implementación del script en Python que permitirá realizar la lectura del ambiente del hogar.	8
R21 - Detección de movimiento en el hogar.	1. Diseño de circuito electrónico para sensor movimiento en el hogar.	2
	2. Implementación del script en Python que permitirá sensor movimiento en el hogar	5
TOTAL, STORY POINTS SPRINT 7		35

3.3.8 Sprint 8

El octavo Sprint contó con una duración de 10 días, inició el 07/10/2019 y finalizó el 18/10/2019, las tareas que se efectuaron fueron R22, R23 y R24, con un grado de prioridad Alta y Media respectivamente. En la Tabla 45 se presentan las subtareas a efectuarse conjuntamente con los respectivos Story Points.

3.3.8.1 Objetivo del Sprint 8

En el Sprint 8 el objetivo se enmarca en la construcción de acciones basadas en la detección de movimiento. Se deberá implementar respuestas cuando se detecte a intrusos en el hogar. En este sentido se deberá capturar una imagen la cual será enviada vía email y también será posteada en la cuenta de Twitter del usuario.

Dentro de este Sprint también se deberá implementar la detección de apertura de puertas basado en las señales del sensor magnético.

Tabla 45. Tareas a efectuarse en Sprint 8

REQUERIMIENTO	SUBTAREAS	STORY POINTS
R22 - Envío automático de email con fotografía al detectar movimiento.	1. Creación de una cuenta de correo electrónico para el envío de emails.	1
	2. Implementación de un script en Python que permita el envío de correos electrónicos.	3
	3. Implementación de un script que permita enviar un correo electrónico cuando se ha detectado movimiento.	5
R23 - Posteo automático con fotografía en la red social Twitter al detectar movimiento.	1. Creación, habilitación y configuración de una cuenta de desarrollador en la red social Twitter.	3
	2. Implementación de un script en Python que permita postear en la red social Twitter cuando se detecte movimiento.	5
R24 - Detección de apertura de puertas.	1. Implementación de un script en Python que permita detectar la apertura de puertas.	5
	2. Diseño del circuito electrónico que permita detectar la apertura de puertas del hogar.	3
TOTAL, STORY POINTS SPRINT 8		25

3.3.9 Sprint 9

El noveno Sprint contó con una duración de 10 días, inició el 21/10/2019 y finalizó el 31/10/2019, las tareas que se efectuaron fueron R28 y R29, con un grado de prioridad Alta. En la Tabla 46 se presentan las subtareas a efectuarse conjuntamente con los respectivos Story Points.

3.3.9.1 Objetivo del Sprint 9

El objetivo del presente Sprint tiene como finalidad la integración de notificaciones push al aplicativo móvil, así como también la implementación de un apartado dedicado al historial de eventos suscitados dentro del hogar, con ello lo que se busca es que el usuario siempre sepa lo que sucede en su vivienda.

Tabla 46. Tareas a efectuarse en Sprint 9

REQUERIMIENTO	SUBTAREAS	STORY POINTS
R28 - Recepción de notificaciones push en tiempo real frente a incidentes dentro del hogar.	1. Implementación de una clase en el aplicativo móvil que permita suscribirse a un "Topic" en Firebase Cloud Messaging para la recepción de notificaciones push en el dispositivo.	3
	2. Implementación de un script en Python que permita enviar notificaciones push a todos los dispositivos suscritos al "Topic".	5
	3. Modificación de colores e iconos en el archivo de manifiesto para la recepción de notificaciones en los dispositivos móviles.	3
R29 - Historial de acciones e incidentes efectuados dentro del hogar.	1. Diseño e implementación de interfaz para fragmento de historial de acciones.	5
	2. Implementación de template y lógica de Recyclerview para el historial de acciones.	13
TOTAL, STORY POINTS SPRINT 9		29

3.3.10 Sprint 10

El décimo Sprint contó con una duración de 10 días, inició el 04/11/2019 y finalizó el 15/11/2019, las tareas que se efectuaron fueron R25, R26 Y R31. En este punto ya quedan únicamente tareas de prioridad Media y Baja, como es el caso de las tareas mencionadas. En la Tabla 47 se presentan las subtareas a efectuarse conjuntamente con los respectivos Story Points.

3.3.10.1 Objetivo del Sprint 10

El principal objetivo del Sprint consiste en ofrecer aspectos de confort y seguridad relacionada con el ambiente del hogar, así como también el acceso al aplicativo y sistema desde cualquier parte que el usuario se encuentre, mediante el uso de una red celular un punto de acceso Wi-Fi.

Tabla 47. Tareas a efectuarse en Sprint 10

REQUERIMIENTO	SUBTAREAS	STORY POINTS
R25 - Control automático para la activación y desactivación del alumbrado público cuando las condiciones de luz natural así lo requieran.	1. Implementación de un script en Python que permita sensor los niveles de luminosidad del medio ambiente, para activar o desactivar el alumbrado público.	5
	2. Diseño del circuito electrónico que permita la lectura de los niveles de luminosidad del medio ambiente.	3
R26 - Detección de humo y gases nocivos dentro del hogar.	1. Implementación de un script en Python que permita sensor la calidad de aire del hogar.	5
	2. Diseño del circuito electrónico que permita sensor la calidad del ambiente del hogar.	3
R31 - Acceso y control del sistema de seguridad a través del aplicativo móvil desde cualquier parte que el usuario se encuentre.	1. Configuración del archivo de manifiesto del aplicativo móvil para que pueda conectarse a la red.	3
	2. Diseño e implementación de la lógica para que el aplicativo verifique el acceso a una conexión de internet.	5
TOTAL, STORY POINTS SPRINT 10		24

3.3.11 Sprint 11

El décimo primero Sprint contó con una duración de 10 días, inició el 18/11/2019 y finalizó el 29/11/2019, las tareas que se efectuaron fueron R14, R17 y R18. Todas ellas de prioridad Baja. En la Tabla 48 se presentan las subtareas a efectuarse conjuntamente con los respectivos Story Points.

3.3.11.1 Objetivo del Sprint 11

El objetivo del Sprint continúa con la adición de características que generen confort al usuario. Se automatizará la iluminación de modo que se pueda activar o desactivar todas a la vez, también se automatizará la apertura del garaje y finalmente la opción para activar o desactivar un simulador de presencia.

Tabla 48. Tareas a efectuarse en Sprint 11

REQUERIMIENTO	SUBTAREAS	STORY POINTS
R14 - Automatización para la activación y desactivación de toda la iluminación.	1. Incorporación de opciones para activar y desactivar toda la iluminación del hogar en el fragmento de iluminación del aplicativo móvil.	3
	2. Implementación de un script en Python que permita activar o desactivar toda la iluminación del hogar a la vez.	5
R17 - Automatización para la apertura y cierre de garaje.	1. Diseño e implementación de la interfaz del fragmento que permitirá la apertura y cierre del garaje desde al aplicativo móvil.	8
	2. Implementación de un script en Python que permita el control de un servomotor.	8
	3. Diseño del circuito electrónico que permita el control de un servomotor.	5
R18 - Automatización para la activación y desactivación de un simulador de presencia.	1. Diseño e implementación de la interfaz del fragmento que permitirá la activación y desactivación de un simulador de presencia.	8
	2. Implementación de un script en Python que permita el control de las lámparas para genera un simulador de presencia.	5
TOTAL, STORY POINTS SPRINT 11		42

3.3.12 Sprint 12

El décimo segundo Sprint contó con una duración de 10 días, inició el 02/12/2019 y finalizó el 13/12/2019, las tareas que se efectuaron fueron R20, R27 y R30. Todas ellas de prioridad Media y Baja. En la Tabla 49 se presentan las subtareas a efectuarse conjuntamente con los respectivos Story Points.

3.3.12.1 Objetivo del Sprint 12

El objetivo del Sprint consiste en la creación de un apartado para configurar diversos aspectos del aplicativo móvil y el modo de operación de los sensores. A su vez también se busca la incorporación de mecanismo que permitan ejecutar acciones frente a ciertos eventos, tales como ejecutar una alarma sonora y un extractor de aire cuando se ha

detectado contaminación en el ambiente, así como también la activación o desactivación de un ventilador o calefactor según la temperatura y las configuraciones del usuario.

Tabla 49. Tareas a efectuarse en Sprint 12

REQUERIMIENTO	SUBTAREAS	STORY POINTS
R30 - Apartado de configuraciones dentro de la aplicación móvil para el control de sensores y dispositivos del hogar.	1. Diseño e implementación del fragmento que permitirá al usuario realizar configuraciones sobre el aplicativo, activación y desactivación de sensores.	10
	2. Implementación de un script en Python que permita manejar los eventos de configuración efectuados desde el aplicativo móvil.	5
R27 - Activación automática de extractor de aire y alarma sonora frente a la detección de gases nocivos.	1. Implementación de un script en Python que permita la activación automática de un extractor de aire y alarma sonora frente a contaminaciones en el ambiente.	5
	2. Diseño del circuito electrónico que permita la activación automática de un extractor de aire y alarma sonora frente a contaminaciones en el ambiente.	3
R20 - Automatización para el control de ventilador y calefactor.	1. Implementación de un script en Python que permita activar o desactivar tanto el ventilador como el calefactor según los valores de temperatura establecidos por el usuario.	5
	2. Diseño del circuito electrónico que permita la activación y desactivación del ventilador y calefactor.	3
TOTAL, STORY POINTS SPRINT 12		31

3.3.13 Sprint 13

El décimo tercero Sprint contó con una duración de 10 días, inició el 16/12/2019 y finalizó el 27/12/2019, la última tarea que se efectuó fue R32. En la Tabla 49 se presentan las subtareas a efectuarse conjuntamente con los respectivos Story Points.

3.3.13.1 Objetivo del Sprint 13

El objetivo del último Sprint consiste en la implementación de todo el sistema de seguridad (sensores, actuadores y placas electrónicas) en el prototipo de vivienda construida, esto con el fin de ejecutar el sistema en un ambiente que simule las características de un hogar.

Tabla 50. Tareas a efectuarse en Sprint 13

REQUERIMIENTO	SUBTAREAS	STORY POINTS
R32 - Implementación del sistema de seguridad en un prototipo de vivienda que simule las características de un hogar.	1. Construcción del prototipo de la vivienda presentada.	13
	2. Implementación del sistema de seguridad en el prototipo de la vivienda.	13
TOTAL, STORY POINTS SPRINT 13		26

3.4 DISEÑO DE LA ARQUITECTURA DEL SISTEMA DE SEGURIDAD

En esta sección se presenta el diseño de la arquitectura con la que fue construido el prototipo de seguridad para control domótico, *mismo que satisface el requerimiento con identificador R01.*

La arquitectura está basada en plataformas y componentes tanto a nivel de hardware como de software, las cuales se comunican y coexisten entre si frente a señales y órdenes que se generan en el medio, de una forma bidireccional y en tiempo real.

La arquitectura dentro del desarrollo de software o cualquier otro contexto, constituyen el pilar sobre el cual reposará el sistema en cuestión. En este sentido, es fundamental y de suma importancia conocer todos los elementos, componentes y partes que intervienen, así como también el modo en que trabajan e interaccionan unos con otros, con la finalidad de generar una base sólida a la hora de la puesta en marcha del sistema como un todo.

3.4.1 Diseño de la arquitectura

Para el diseño de la arquitectura que se presenta la Figura 30, se analizaron diversos elementos, tecnologías y herramientas que respalden a las características con las que se desea que cuente el sistema de seguridad. Para ello se ha generado dos componentes de software y uno de hardware.

El primer componente de software es Firebase, mismo que permitirá el alojamiento de datos y el establecimiento de comunicación entre los diversos elementos y partes del sistema. En síntesis, la presente plataforma actuara como agente de escucha de forma bidireccional para los diversos elementos que forman parte del sistema de seguridad.

El segundo componente de software, a su vez, proporcionara los mecanismos para él envío, recepción, monitoreo y ejecución de tareas de manera remota desde un aplicativo

móvil desarrollado para la plataforma Android, el cual básicamente servirá de interfaz para que el usuario pueda interactuar con los diversos elementos del hogar.

Finalmente, el tercer componente de hardware es Raspberry Pi, mismo que constituye un pilar fundamental del sistema ya que es el encargado del envío, procesamiento, recepción y manejo de información proveniente de sensores y para la puesta en marcha de actuadores.

En este punto, tanto para el procesamiento de las señales captadas del medio como para la ejecución de acciones sobre el mismo, se ha utilizado a Python como lenguaje de programación, mismo que a su vez se encuentra embebido en la placa Raspberry Pi, permitiendo así la ejecución de tareas.

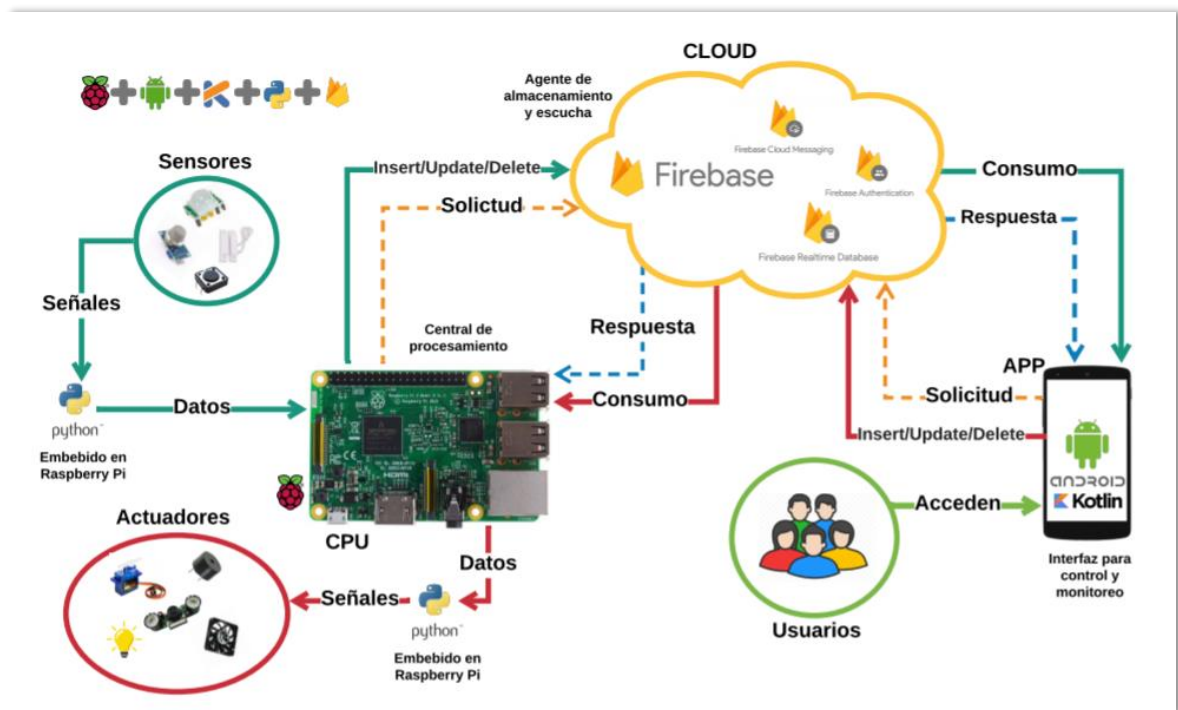


Figura 30. Arquitectura del prototipo de seguridad para control doméstico

3.5 DISEÑO DE LA ESTRUCTURA PARA LA BASE DE DATOS

Para el alojamiento de información generada por el aplicativo móvil y los diversos sensores y actuadores, se hará uso de la plataforma software Firebase, específicamente Firestore. Firestore es una base de datos en tiempo real albergada en la nube, desarrollada y mantenida por Google. La particularidad de la base que se describe, es que no maneja estructuras, ni comandos SQL como las tradicionales bases de datos, está a su vez maneja estructuras NoSQL, específicamente representada por objetos JSON (JavaScript Object Notation).

3.5.1 Diseño de la estructura NoSQL

Firestore se enmarca en los conceptos claves de colecciones y documentos para la creación de modelos NoSQL. Los documentos permiten el almacenamiento de datos mediante la creación de campos a los cuales se les asignan valores. Posteriormente dichos documentos son almacenados en colecciones que básicamente se convierten en contenedores que mantienen organizada la información y hacen posible la compilación y ejecución de consultas.

En la Figura 31, se presenta la estructura diseñada para el almacenamiento de información del sistema de seguridad.

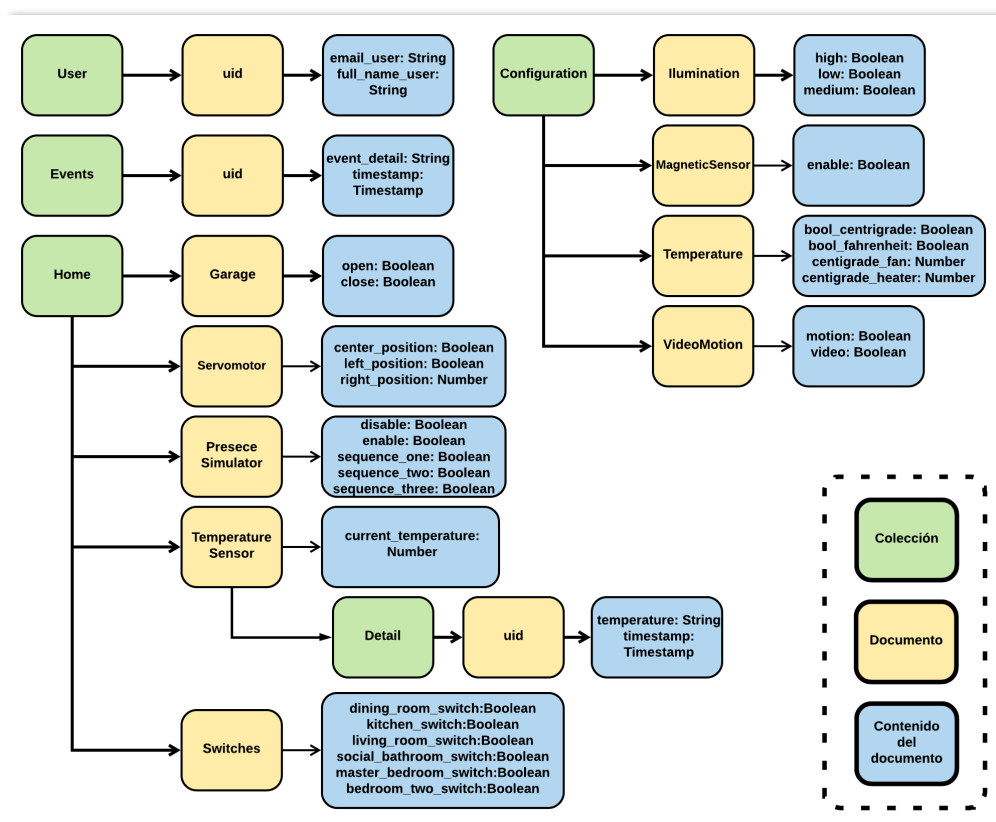


Figura 31. Estructura NoSQL para Firestore

3.6 DISEÑO DE LAS INTERFACES DEL APLICATIVO MÓVIL

En esta sección se presentan los diseños de las diversas pantallas que formarán parte del aplicativo móvil. Para la elaboración del diseño se emplearán mockups con la finalidad de replicarlos en la etapa de construcción, mediante el lenguaje de etiquetas XML, mismo del cual se apoya Android para el desarrollo de aplicativos móviles.

3.6.1 Diseño de la interfaz de inicio de sesión

El aplicativo móvil contará con un apartado destinado a la seguridad y autenticación, con el objetivo de que solo usuarios registrados en el sistema puedan acceder a sus funciones.

En la Figura 32, se presenta el diseño de la interfaz para el inicio de sesión.



Figura 32. Interfaz para inicio de sesión

3.6.2 Diseño de la interfaz para creación de cuenta

En la Figura 33, se puede observar el diseño de la pantalla para creación de cuenta, misma que se desarrollará con la finalidad de que el usuario pueda crear una cuenta con una dirección de correo electrónico.



Figura 33. Interfaz para creación de cuenta

3.6.3 Diseño de la interfaz para restablecer contraseña

La interfaz que se observa en la Figura 34, se ha diseñado con el objetivo de que el usuario pueda restablecer la contraseña de su cuenta en caso de que esta haya sido olvidada.



Figura 34. Interfaz para restablecimiento de contraseña

3.6.4 Diseño del menú lateral

En la Figura 35, se observa el diseño del componente *Navigation Drawer*, el cual consiste en un menú lateral desplegable que es capaz de contener y ejecutar múltiples funciones.



Figura 35 - Diseño del menú lateral

3.6.5 Diseño de la interfaz para mostrar el historial de eventos

El diseño de la interfaz para mostrar el historial de eventos, se la puede observar en la Figura 36, misma que hará uso del componente Recyclerview para su construcción.



Figura 36. Interfaz para mostrar el historial de eventos

3.6.6 Diseño de la interfaz para control de cámara y visualización de video

La interfaz presentada en la Figura 37, permitirá al usuario observar el interior de su vivienda en tiempo real, así como también permitirá la rotación de la cámara para obtener una visión en diferentes posiciones.

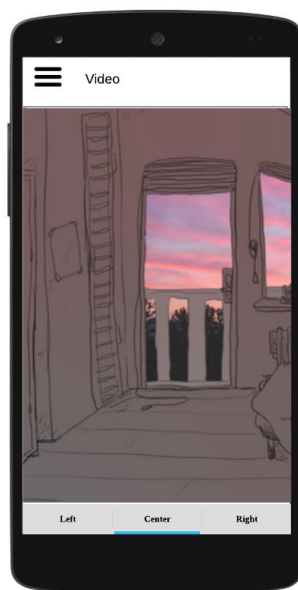


Figura 37. Interfaz para control y visualización de video en tiempo real

3.6.7 Diseño de la interfaz para control de garaje

El diseño de la interfaz que se observa en la Figura 38, permitirá al usuario abrir y cerrar el garaje de su hogar.

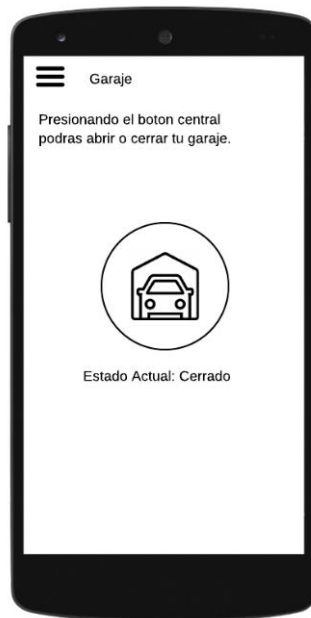


Figura 38. Interfaz para control de garaje

3.6.8 Diseño de la interfaz para control de iluminación

La interfaz diseñada para el control de iluminación puede ser apreciada en la Figura 39. Misma permitirá al usuario ejecutar acciones sobre las lámparas del hogar.



Figura 39. Interfaz para control de iluminación

3.6.9 Diseño de la interfaz para monitoreo de temperatura

La Figura 40, presenta la interfaz que permitirá el monitoreo y despliegue de información concerniente a la temperatura del hogar.

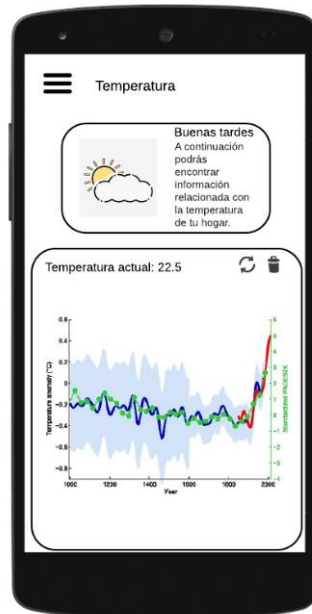


Figura 40. Interfaz para monitoreo de temperatura

3.6.10 Diseño de la interfaz para control del simulador de presencia

La interfaz para controlar el simulador de presencia puede ser observada en la Figura 41. Esta pantalla permitirá al usuario activar o desactivar un ambiente que simulará presencia en el inmueble cuando esté se encuentre deshabitado.



Figura 41. Interfaz para control del simulador de presencia

3.6.11 Diseño de la interfaz para configuraciones

Finalmente, en la Figura 42, se observa la interfaz que dará soporte a la ejecución de configuraciones relacionadas con el sistema de seguridad.



Figura 42. Interfaz para configuraciones

3.7 DISEÑO DE LOS CIRCUITOS ELECTRÓNICOS

En esta sección se presenta el diagrama y diseño *PCB (Printer Circuito Board)* de los diferentes circuitos electrónicos a ser usados en el sistema de seguridad prototipo.

Como primer apartado se tiene la definición de pines tanto para sensores como actuadores, elementos que forman parte del componente hardware del sistema. Para ello se elaborará una representación de los pines GPIO de Raspberry Pi, con la finalidad de conocer el trabajo que efectuará cada pin seleccionado del encabezado GPIO.

Una vez que se definan los pines a ser usados, se procederá a diseñar los circuitos electrónicos que permitan establecer comunicación entre Raspberry Pi y los dispositivos electrónicos tanto de entrada como de salida (sensores y actuadores).

Finalmente, con los circuitos diseñados, se procederá con la elaboración de las placas de circuito impreso, en caso de ser requerido por el sistema de seguridad.

3.7.1 Definición de pines del encabezado GPIO

A continuación, en la Figura 43, se presenta la definición de pines del encabezado GPIO a ser usados para la interacción tanto con sensores, actuadores y módulos electrónicos.

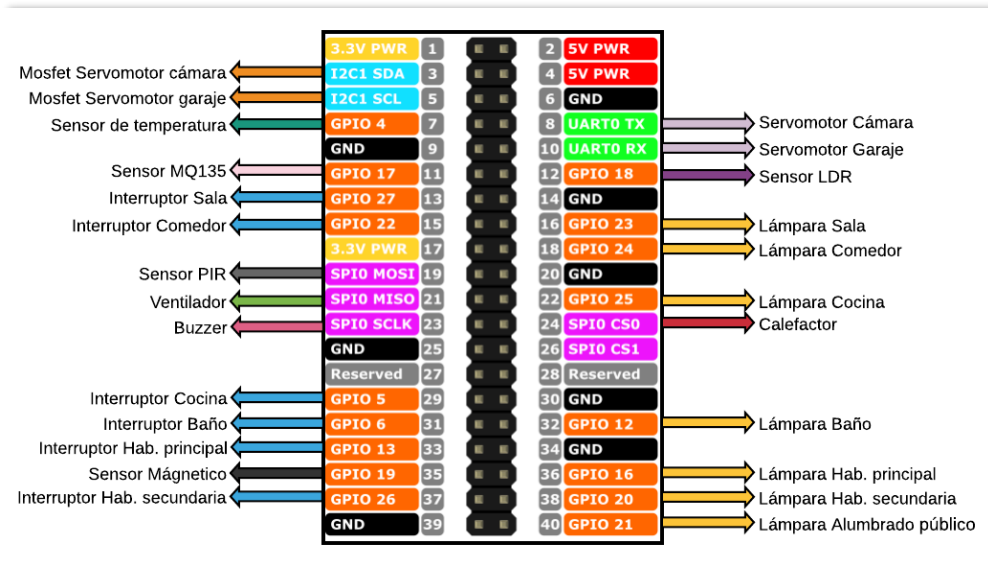


Figura 43. Definición de pines del encabezado GPIO

3.7.2 Diagrama del circuito para lectura de temperatura

En la Figura 44, se presenta el diagrama electrónico que permitirá sensar la temperatura del hogar. Para ello se hará uso del módulo DS18B20 y del pin 4 de Raspberry Pi para la lectura de los datos provenientes del sensor.

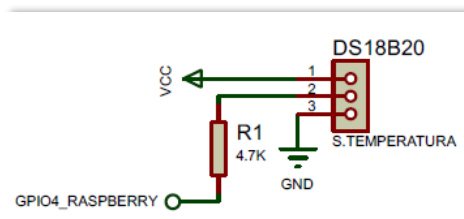


Figura 44. Diagrama del circuito para lectura de temperatura

3.7.3 Diagrama del circuito para detección de gases

Para la detección de gases nocivos dentro del hogar, se usará el sensor MQ-135 y el pin 17 que permitirá interactuar a Raspberry Pi con el sensor. En la Figura 45, se presenta el diseño del circuito.

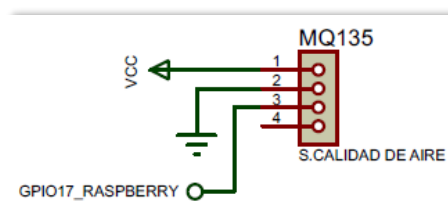


Figura 45. Diagrama del circuito para la detección de gases

3.7.4 Diagrama del circuito para detección de movimiento

Para la detección de movimiento dentro del hogar se usará el sensor PIR HC-SR501 y el pin 10 de Raspberry Pi, tal como se muestra en la Figura 46.

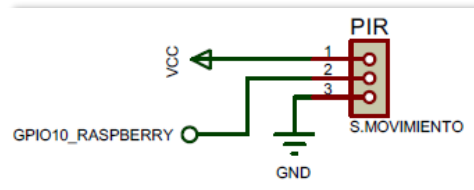


Figura 46. Diagrama del circuito para detección de movimiento

3.7.5 Diagrama del circuito para control del ventilador

Para el control del ventilador se hará uso del esquema presentado en la Figura 47. El pin 9 de Raspberry Pi será el encargado de manejar el estado del dispositivo.

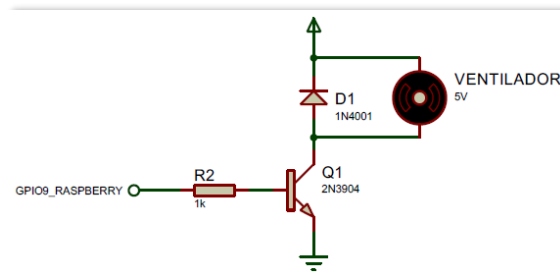


Figura 47. Diagrama del circuito para control del ventilador

3.7.6 Diagrama del circuito para control del calefactor y buzzer

En la Figura 48, se presenta el diagrama que dará soporte al control tanto del buzzer como del calefactor, que en este caso será representado por una bombilla.

El pin 8 y el pin 9 de Raspberry Pi serán los encargados del manejo y control del calefactor y buzzer, respectivamente.

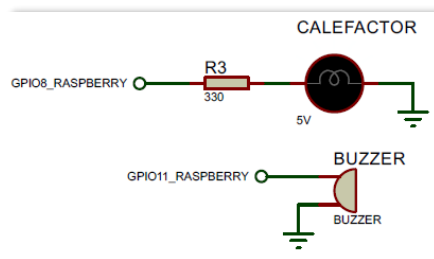


Figura 48. Diagrama del circuito para control del calefactor y buzzer

3.7.7 Diagrama del circuito para detección de apertura de puertas

Para la detección de apertura de puertas se emplearán sensores magnéticos. El pin 19 de Raspberry Pi será el encargado de manejar las señales captadas por el sensor. El diagrama implementado se presenta en el Figura 49.

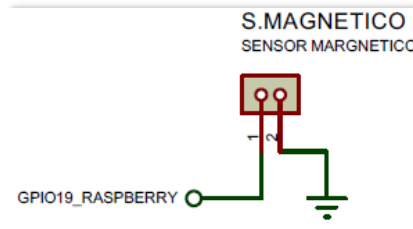


Figura 49. Diagrama del circuito para detección de apertura de puertas

3.7.8 Diagrama del circuito para control de servomotores

Para el diseño del circuito presentado en la Figura 50 que permitirá el control de servomotores tanto para la apertura y cierre del garaje como para la rotación de la cámara de seguridad, se hará uso del driver mosfet IRF520 en conjunto con los micro servos SG90.

El pin 2 y pin 3 de Raspberry Pi serán los encargados de controlar los drivers mientras que los pines 14 y 15 controlarán la señal PWM de los servomotores.

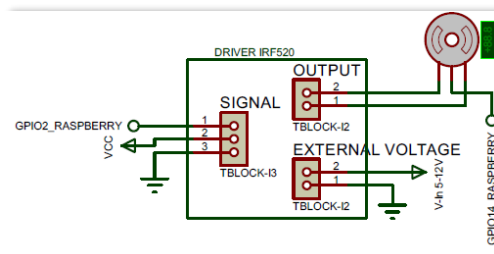


Figura 50. Diagrama del circuito para control de servomotores

3.7.9 Diagrama del circuito para detección de luminosidad

En la Figura 51, se presenta el diagrama del circuito para el control de luminosidad. El pin 18 de Raspberry Pi será el encargado de manejar los datos provenientes del sensor.

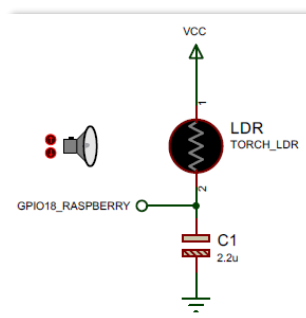


Figura 51. Diagrama del circuito para detección de luminosidad

3.7.10 Diagrama del circuito de potencia para iluminación

El diagrama del circuito presentado en la Figura 52, será el encargado de proporcionar la potencia y el voltaje (110 VCA) para la activación de las lámparas del hogar frente a la recepción de señales provenientes desde el aplicativo móvil y de la placa de circuito impreso para control de iluminación.

Los pines 23, 24, 12, 16, 20 y 21 de Raspberry Pi serán los encargados de manejar la salida de las señales hacia el circuito de potencia.

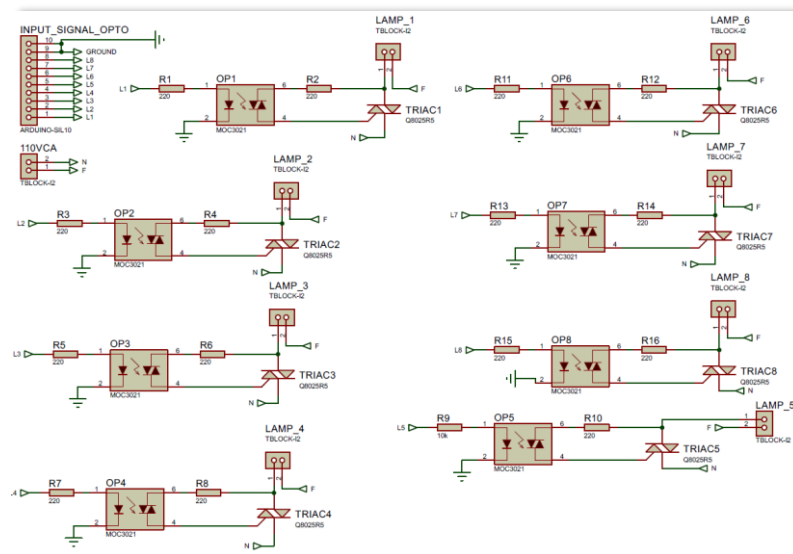


Figura 52. Diagrama del circuito de potencia para iluminación

3.7.10.1 Diseño del circuito impreso para potencia de iluminación

Ya que es necesario contar con una placa física que permita proporcionar la potencia y voltaje para la activación de las lámparas del hogar, en este apartado se presenta en la Figura 53 el diseño del circuito impreso que permitirá la creación del PCB. Así también en la Figura 54 se expone una pre visualización del resultado final del PCB basado en el diagrama electrónico diseñado.

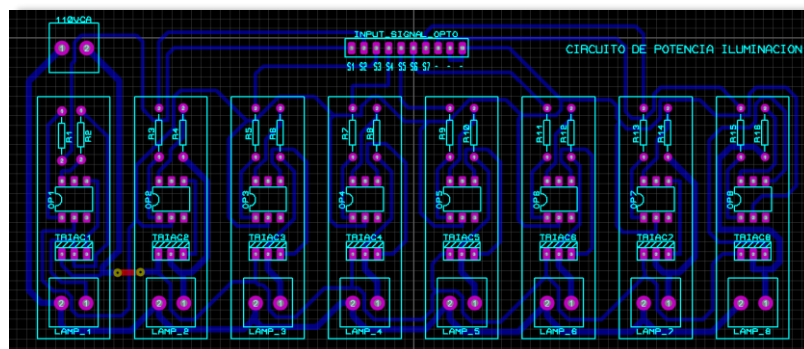


Figura 53. Diseño del PCB para potencia de iluminación

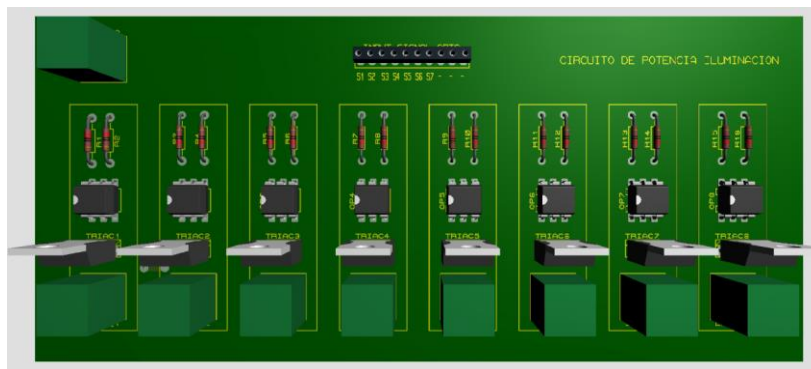


Figura 54. Previsualización del PCB para potencia de iluminación

3.7.11 Diagrama del circuito para control de iluminación

En la Figura 55, se presenta el diagrama del circuito para control de iluminación del hogar. Mediante éste, el usuario podrá activar y desactivar las lámparas del hogar de forma física.

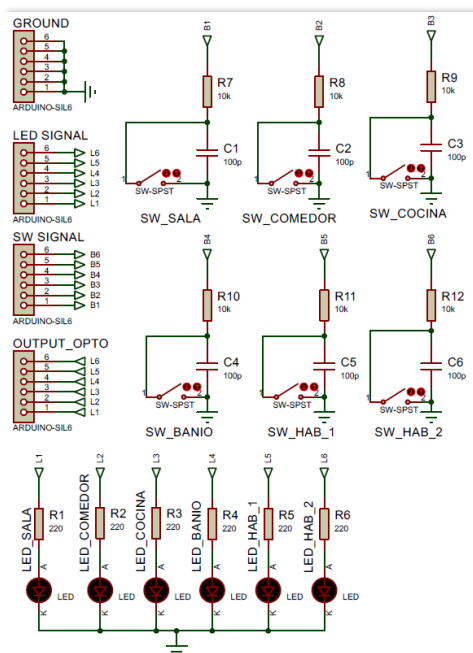


Figura 55. Diagrama del circuito para control de iluminación

3.7.11.1 Diseño del circuito impreso para control de iluminación

El diagrama presentado en 3.8.12, es necesario que sea implementado en una placa de circuito impreso, esto con la finalidad de que el usuario puede ejecutar acciones sobre los pulsadores para la activación o desactivación de las lámparas del hogar.

En la Figura 56 se presenta el diseño del circuito impreso que permitirá la creación del PCB. Así también en la Figura 57 se expone una pre visualización del resultado final del PCB basado en el diagrama electrónico diseñado.

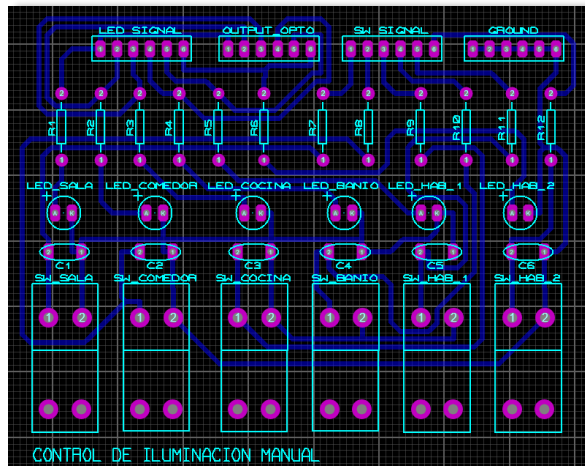


Figura 56. Diseño del PCB para control de iluminación



Figura 57. Previsualización del PCB para control de iluminación.

CAPÍTULO IV

4. CONSTRUCCIÓN DEL PROTOTIPO DE SEGURIDAD

El presente capítulo se enmarca en la descripción de las herramientas, lenguajes de programación, librerías y entornos de desarrollo que dieron soporte a la construcción del prototipo de seguridad para control domótico.

En base a los diseños y requerimientos establecidos en el Capítulo III, se presenta la solución tanto de los componentes hardware como software.

La construcción de los circuitos electrónicos, así como también de las interfaces del aplicativo móvil, serán expuestas en la presente sección. Posteriormente la implementación del sistema de seguridad para control domótico en un prototipo que permita simular las características de una vivienda y finalmente la ejecución de pruebas que permitan obtener resultados para contrastarlos con lo solicitado en la etapa de definición de requerimientos.

4.1 INSTALACIÓN DE RASPBIAN EN RASPBERRY PI

4.1.1 Descarga del sistema operativo Raspbian

La placa electrónica Raspberry Pi, constituye un elemento de hardware fundamental en el sistema de seguridad que se presenta. Por ello, el primer paso para que esté entre en funcionamiento, consiste en la instalación del sistema operativo que se encargará de la gestión de recursos y servicios.

Existe un gran número de sistemas operativos disponibles para Raspberry Pi, sin embargo, uno de los más usados y compatible con todos los modelos es Raspbian, sistema operativo oficial de Raspberry Pi, mismo que se expone en la Figura 58.

Una vez seleccionado el sistema operativo a ser usado, en nuestro caso Raspbian con una interfaz gráfica de escritorio, es necesario descargar la imagen del sistema operativo, misma que puede ser obtenida desde la página oficial de Raspberry Pi disponible en [42].



Figura 58. Imagen del sistema operativo Raspbian

4.1.2 Descarga e instalación del programa balenaEtcher

Con la imagen descargada, es necesario preparar el hardware del cual hará uso Raspberry Pi. Inicialmente es necesario contar con una tarjeta microSD, misma que albergará el sistema operativo, así como también con periféricos de entrada y salida (mouse, teclado, monitor) que permitirán la posterior configuración.

Con el hardware listo, el primer paso consiste en la instalación del sistema operativo en la microSD. Para lograr este objetivo se hará uso de la herramienta balenaEtcher, misma que puede ser descargada desde su página oficial presentada en la Figura 59 y disponible en [43].

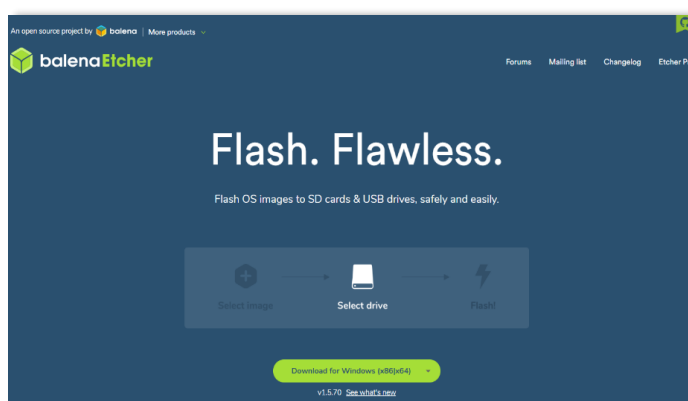


Figura 59. Sitio oficial de la herramienta balenaEtcher

Una vez descargada e instalada la herramienta, procedemos a ejecutarla. Se observará una pantalla como se muestra en la Figura 60. En ella lo único que se debe hacer es seleccionar la imagen descargada y el disco destino de instalación, en este caso la microSD. Una vez efectuada dicha configuración, únicamente resta presionar la opción Flash. El programa empezará a trabajar y notificará la finalización del proceso.

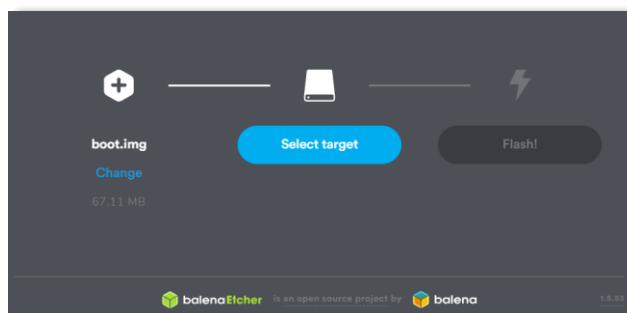


Figura 60. Interfaz del software balenaEtcher

4.1.3 Instalación de Raspbian en Raspberry Pi

Con el sistema operativo listo para ser instalado en Raspberry Pi, se procede a insertar la tarjeta microSD en la placa, así como también los periféricos y la fuente de alimentación, tal como se muestra en la Figura 61.

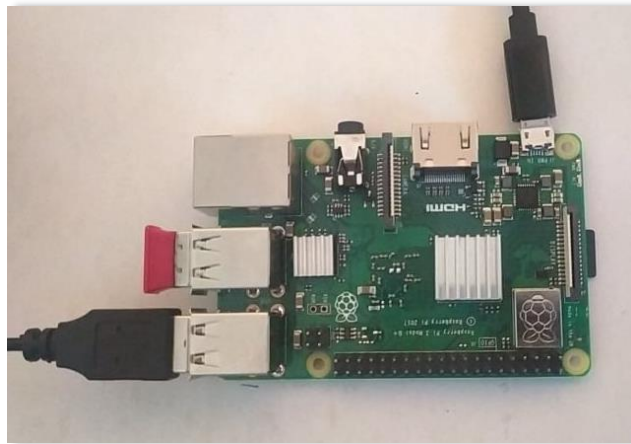


Figura 61. Conexión de periféricos Raspberry Pi

Una vez que la energía fluya por Raspberry Pi, el sistema operativo se instalará de forma automática. En el proceso de instalación el asistente solicitará al usuario una contraseña y las credenciales de acceso para establecer conexión con una red Wi-Fi (si no se hace uso de una conexión cableada). Con ello la placa queda lista para empezar a operar, desplegando su interfaz tal como se observa en la Figura 62.

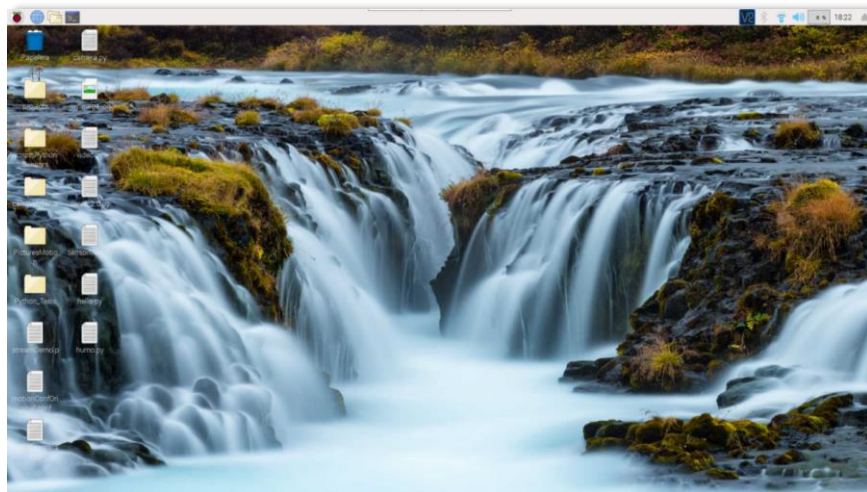


Figura 62. Interfaz del sistema operativo Raspbian

4.1.4 Configuración y habilitación de interfaces en Raspberry Pi

Raspberry Pi trae consigo interfaces que permiten la lectura, control y ejecución de diversos elementos, tales como sensores, cámaras, comunicación remota, entre otros. Para ello es necesario configurar y habilitar dichas interfaces, esto se lo realiza mediante la ejecución del comando `sudo rasp-config` en la terminal. Ejecutado el comando se desplegará la pantalla que se muestra en la Figura 63. De la cual se deberá seleccionar la opción *Interfacing Options*.

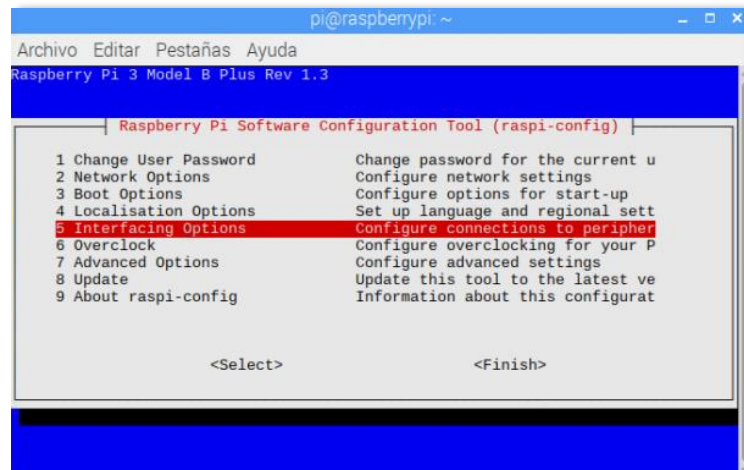


Figura 63. Interfaces Raspberry Pi

Ejecutada la orden, aparecerán todas las interfaces y protocolos disponibles. Por lo general todas las opciones por defecto están deshabilitadas, estas deberán ser modificadas y habilitadas, como se muestra en la Figura 64. Una vez se hayan habilitado todas las opciones, es necesario reiniciar Raspberry Pi, proceso que permitirá aplicar los cambios, permitiendo que las interfaces queden totalmente operables.

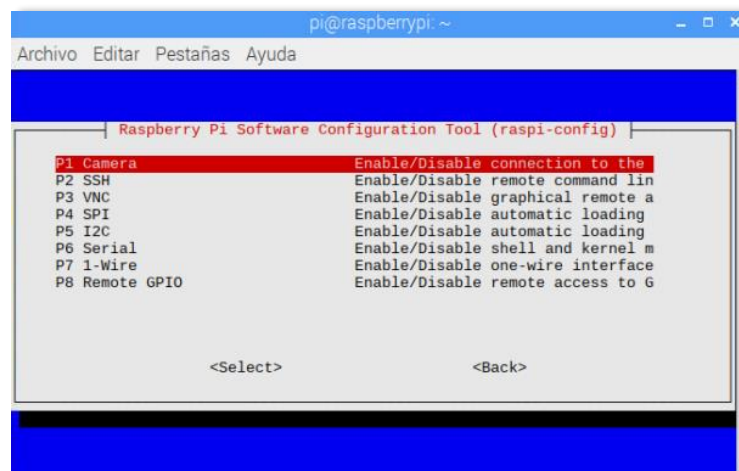


Figura 64. Habilitación de interfaces Raspberry Pi.




4.2 HERRAMIENTAS

En esta sección se presentan y describen las herramientas que dieron soporte a la construcción de los componentes hardware y software del sistema de seguridad para control doméstico.

Inicialmente se presentan herramientas relacionadas con el control de versiones, seguido de lenguajes de programación, base de datos, entornos de desarrollo (IDEs), librerías y finalmente instrumentos para la diagramación y obtención de recursos.




4.2.1 Control de versiones

Tabla 51. Herramientas para control de versiones

NOMBRE	DESCRIPCIÓN	UTILIZADO EN	LOGO
Git	Es un sistema de control de versiones, diseñado para manejar proyectos pequeños o de gran escala, totalmente gratuito y de código abierto [44].	Aplicaciones web, móviles y de escritorio.	
Github	Es una plataforma o repositorio, diseñado para el alojamiento de proyectos y trabajo colaborativo basado en el sistema de control de versiones Git [45].	Aplicaciones web, móviles y de escritorio.	
Github Desktop	Es una aplicación de escritorio desarrollada por Github, para el manejo de versiones Git de manera gráfica [46].	Aplicaciones web, móviles y de escritorio.	




4.2.2 Lenguajes de programación y base de datos

Tabla 52. Lenguajes de programación y base de datos

NOMBRE	DESCRIPCIÓN	UTILIZADO EN	LOGO
Kotlin	Es un lenguaje de programación estáticamente tipado que admite programación funcional y orientada a objetos. Desarrollado por JetBrains y soportado por Google. Comúnmente usado para el desarrollo móvil de la plataforma Android [47].	Aplicaciones web móviles y de escritorio.	
Python	Es un lenguaje de programación multiparadigma, de libre uso y distribución. Se caracteriza por su sencillez y legibilidad a la hora de escribir programas, es uno de los lenguajes de programación más poderosos y populares a nivel mundial [48].	Aplicaciones web, móviles y de escritorio.	
Firebase	Es una plataforma para el desarrollo de aplicaciones web y móviles. Ofrece soporte para Android, iOS, Javascript, Python entre los más populares. Se encuentra alojada en la nube, usa un conjunto de herramientas para la creación y sincronización de proyectos ofreciendo funciones como estadísticas, base de datos, mensajería, hosting, entre otros. [26].	Aplicaciones web y móviles.	





4.2.3 Entornos de desarrollo para hardware y software









Tabla 53. Entornos de desarrollo para hardware y software

NOMBRE	DESCRIPCIÓN	UTILIZADO EN	LOGO
Android Studio	Es el entorno de desarrollo oficial para la creación de aplicaciones móviles bajo la plataforma Android. Cuenta con un gran número de herramientas y características que permiten obtener como resultado aplicaciones móviles de calidad [49].	Aplicaciones móviles	
PyCharm	Es un entorno de desarrollo integrado, mantenido por JetBrains para la creación de aplicaciones bajo el lenguaje de programación Python. Se caracteriza por su gran cantidad de herramientas y productividad al momento de escribir código [50].	Aplicaciones web, móviles y de escritorio.	
Proteus	Es un entorno de desarrollo para el diseño y simulación de circuitos electrónicos, con soporte para la elaboración de placas de circuito impreso. Cuenta con una biblioteca sumamente amplia y con características que lo convierten en el software más popular para el diseño de circuitos electrónicos [51].	Diseño de circuitos electrónicos	

4.2.4 Librerías





Tabla 54. Librerías para desarrollo

NOMBRE	DESCRIPCIÓN	UTILIZADO EN	LOGO
GPIO Zero	Librería desarrollada por Ben Nutall, miembro de la Fundación Raspberry Pi, en conjunto con otros colaboradores. Provee un acceso de manera sencilla a las interfaces GPIO de Raspberry Pi, para interactuar con elementos electrónicos [52].	Circuitos electrónicos con Raspberry Pi	
Pi Camera	Es un paquete que proporciona una interfaz puramente escrita en Python para acceder al módulo de la cámara de Raspberry Pi. Cuenta con soporte para Python 2.7 o superior [53].	Aplicaciones desarrolladas con Python y Raspberry Pi	
Twython	Es un paquete escrito puramente en Python que permite interactuar con la API de la red social Twitter [54].	Aplicaciones desarrolladas con Python.	
Smtplib	Es un módulo que permite la definición de objetos SMTP para el envío de correos electrónicos a cualquier computador o dispositivo que soporte dicho protocolo [55].	Aplicaciones desarrolladas con Python.	

Firestore admin	Es un paquete para los desarrolladores de Python que permite la integración de Firestore a servicios o aplicaciones [56].	Aplicaciones web y móviles	 sin imagen
Flask	Es un framework escrito en Python que permite la creación de aplicaciones web, así como también de servidores streaming [57].	Aplicaciones web	
Glide	Es una biblioteca de carga de imágenes rápida y eficiente desarrollada para la plataforma móvil Android [58].	Aplicaciones móviles	
Lottie	Es una biblioteca con soporte para Android, iOS y React Native, permite la adición de animaciones de alta calidad de manera sencilla y eficiente [59].	Aplicaciones web y móvil	
Bubble navigation	Es una biblioteca liviana, fácil de usar, que permite la creación de sorprendentes barras de navegación con una gran cantidad de opciones de personalización, para aplicaciones móviles Android [60].	Aplicaciones móviles	 sin imagen
Firestore UI	Es una biblioteca que permite simplificar la vinculación de los datos de Firestore Real Time Database o Firestore a la interfaz de aplicaciones móviles [61]	Aplicaciones móviles	 sin imagen
Material dialogs	Es una biblioteca que permite la creación de cuadros de diálogos basados en material design, para aplicaciones móviles Android escritas en Kotlin [62].	Aplicaciones móviles	 sin imagen
MP Android Chart	Es una biblioteca que permite la creación de gráficos para aplicaciones móviles Android, tales como líneas, barras, pasteles, burbujas y velas [63].	Aplicaciones móviles	

4.2.5 Recursos y diagramación

Tabla 55. Recursos y diagramación

NOMBRE	DESCRIPCIÓN	UTILIZADO EN	LOGO
LottieFiles	Es un repositorio para la descarga, creación y edición de animaciones para la biblioteca Lottie [64].	Aplicaciones web y móviles	
Flaticon	Es una página web que permite la descarga de iconos en formatos PNG, SVG, EPS, PSD y Base64 [65].	Aplicaciones web, móviles y de escritorio.	
Freepik	Es una página web que permite la descarga de recursos tales como iconos, fondos, vectores, entre otros [66].	Aplicaciones web, móviles y de escritorio.	
Lucidchart	Es una herramienta web que permite el trabajo colaborativo para la elaboración de diagramas, mapas, entre otros [67].	Aplicaciones web, móviles y de escritorio.	

4.3 CONSTRUCCIÓN DEL APLICATIVO MÓVIL

En este apartado se presenta la construcción del aplicativo móvil desarrollado para la plataforma Android, *mismo satisface al requerimiento con identificador R02*.

Para su implementación se usó el IDE Android Studio en conjunto con el lenguaje de programación Kotlin, Firebase y bibliotecas de terceros expuestas en la sección 4.3.4. Todos los elementos mencionados permitirán obtener como resultado interfaces, funcionalidades y características solicitadas en los requerimientos iniciales.

4.3.1 Compatibilidad del aplicativo móvil

El aplicativo móvil ha sido desarrollado para ejecutarse en dispositivos con un nivel mínimo de API 19, esto se traduce a que la aplicación podrá ejecutarse en móviles que cuenten con un sistema operativo KitKat 4.4 o superior. Alrededor de 95.3% dispositivos móviles con sistema operativo Android según [68] y expuesto en la Figura 65.

ANDROID PLATFORM VERSION	API LEVEL	CUMULATIVE DISTRIBUTION
4.0 Ice Cream Sandwich	15	
4.1 Jelly Bean	16	99,6%
4.2 Jelly Bean	17	98,1%
4.3 Jelly Bean	18	95,9%
4.4 KitKat	19	95,3%
5.0 Lollipop	21	85,0%
5.1 Lollipop	22	80,2%
6.0 Marshmallow	23	62,6%
7.0 Nougat	24	37,1%
7.1 Nougat	25	14,2%
8.0 Oreo	26	6,0%
8.1 Oreo	27	1,1%

Figura 65. Porcentaje de dispositivos móviles compatibles con la app
Fuente [68]

4.3.2 Creación de proyecto e integración del SDK de Firebase

Para la incorporación de los SDK de Firebase al aplicativo móvil, es necesario contar con una cuenta de Google ya que esta será solicitada al tratar de ingresar a la plataforma expuesta en la Figura 66 y disponible en [69].

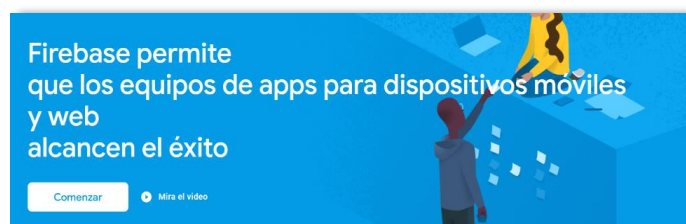


Figura 66. Sitio oficial de Firebase

Una vez que se proporcione y sean validadas las credenciales de acceso, se procederá con la creación de un nuevo proyecto. Para ello hay que dirigirse a la opción *Ir a la consola*, en ella se podrá crear y observar los diversos proyectos del usuario en cuestión, tal como se muestra en la Figura 67.

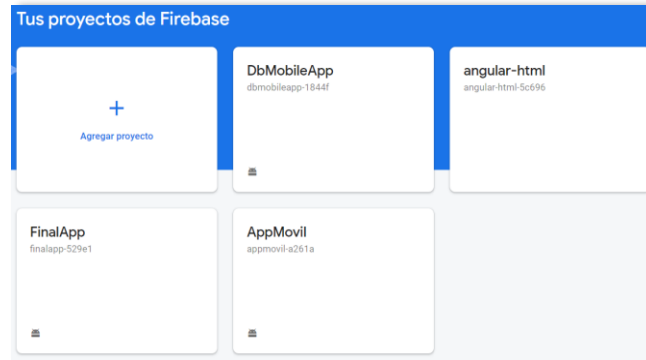


Figura 67. Creación de proyectos en Firebase

Para la creación de un nuevo proyecto, únicamente basta con seleccionar la opción *Agregar proyecto*. Se desplegará un modal como el que se muestra en la Figura 68, el mismo que solicitará información concerniente al proyecto a desarrollarse.

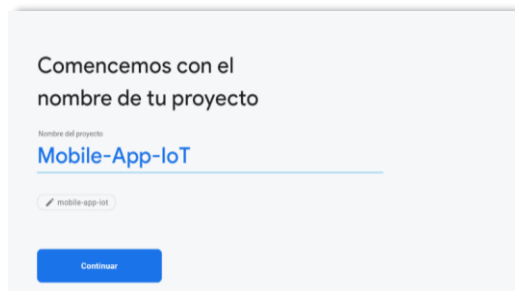


Figura 68. Asignación de nombre del proyecto

El segundo paso consiste en habilitar o deshabilitar la opción de Google Analytics desde la pantalla mostrada en la Figura 69 para nuestro proyecto, en este caso usaremos dicha opción ya que es lo recomendado por Google.

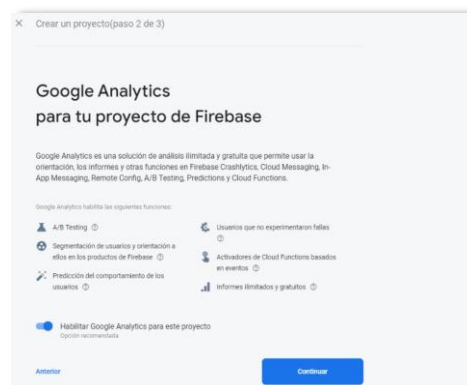


Figura 69. Adición de Google Analytics

Finalmente, el tercer paso consiste en seleccionar la zona en donde se está llevando a cabo el desarrollo del proyecto, así como también la aceptación de condiciones de Google Analytics, desde la página mostrada en la Figura 70.

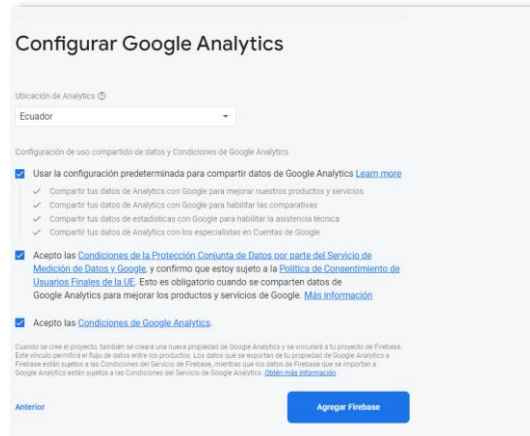


Figura 70. Selección de zona y condiciones del servicio

Posteriormente, Firebase empezará con la creación del proyecto que tomará unos pocos segundos, cuando esté haya finalizado, el proyecto estará listo para ser usado. Se nos redirigirá a la consola de Firebase, en ella podremos observar todo el abanico de funciones que nos ofrece la plataforma, tal como se muestra en la Figura 71.



Figura 71. Consola de Firebase

Una vez dentro de la consola se procederá con integración de Firebase a nuestro aplicativo. Para ello se procede con la selección de la plataforma móvil Android, como se muestra en la Figura 72.



Figura 72. Plataformas disponibles

Seleccionada la plataforma en cuestión, se desplegará una pantalla como la presentada en la Figura 73, misma que solicitará el ingreso del nombre de paquete del aplicativo generado con Android Studio, así como también el certificado de firma SH1, que puede ser generado con las herramientas de Google.



Figura 73. Ingreso de información del aplicativo móvil

Una vez que se haya ingresado toda la información se procederá a registrar la app. Con ello Firebase generará un archivo en formato JSON que deberá ser agregado al árbol del proyecto generado en Android Studio como se muestra en la Figuras 74 y 75.



Figura 74. Generación de archivo google-services.json

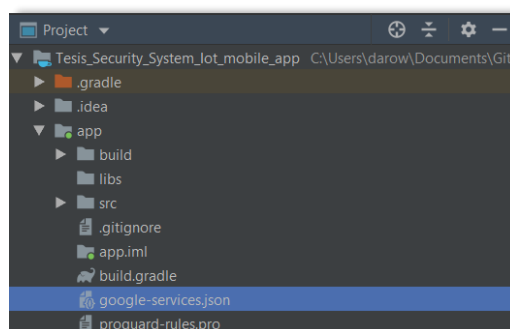


Figura 75. Adición del archivo google-services.json al árbol del proyecto

Cuando el archivo `google-services.json` haya sido añadido al árbol del proyecto, es necesario incluir las reglas de Google Services. Para ello hay que dirigirse al gradle que se encuentra al nivel de proyecto y agregar las dependencias como se muestra en la Figura 76.

```
buildscript {
    // ...
    dependencies {
        // ...
        // Add the following line:
        classpath 'com.google.gms:google-services:4.2.0' // Google Services plugin
    }
}

allprojects {
    // ...
    repositories {
        // Check that you have the following line (if not, add it):
        google() // Google's Maven repository
        // ...
    }
}
```

Figura 76. Dependencias a nivel de proyecto

Posteriormente es necesario agregar las dependencias de Firebase y Google Services al gradle que se encuentra a nivel de aplicación como se visualiza en la Figura 77.

```
dependencies {
    // ...
    implementation 'com.google.firebase:firebase-core:16.0.8'

    // Getting a "Could not find" error? Make sure that you've added
    // Google's Maven repository to your root-level build.gradle file
}

apply plugin: 'com.android.application'

android {
    // ...
}

// Add the following line to the bottom of the file:
apply plugin: 'com.google.gms:google-services' // Google Play services Gradle plugin
```

Figura 77. Dependencias a nivel de aplicación

Una vez agregadas todas las dependencias, se procede a sincronizar el gradle para que las acciones surtan efecto, con ello la integración de los SDK de Firebase al aplicativo se habrán añadido satisfactoriamente.

4.3.3 Métodos de acceso para autenticación de usuarios en Firebase

La consola de Firebase trae consigo un gran número de funciones que pueden ser añadidas a las diversas plataformas soportadas. Una de ellas es Firebase Authentication, función que brinda soporte para autenticación de usuarios haciendo uso de proveedores de acceso tales como Correo electrónico/contraseña, Google, Facebook, Twitter, entre otros.

Para hacer uso de esta función es necesario dirigirse a la consola de Firebase, opción *Authentication*, pestaña *Método de acceso* y habilitar los proveedores a ser usados, en este caso *Correo electrónico/contraseña* y *Google*, como se muestra en la Figura 78.

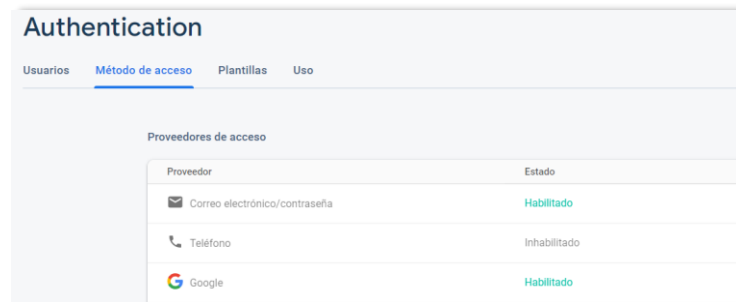


Figura 78. *Habilitación de proveedores de acceso*

Cuando las opciones hayan sido habilitadas en la consola, el aplicativo podrá acceder a todas las funcionalidades de Firebase Authentication.

4.3.4 Estructura del proyecto en Android Studio

En la Figura 79, se presenta la estructura del proyecto que brindará soporte al funcionamiento de la app. Cada uno de los componentes de la misma será descrita, de manera que permita entender cada una de las carpetas y archivos que forma parte del aplicativo móvil.

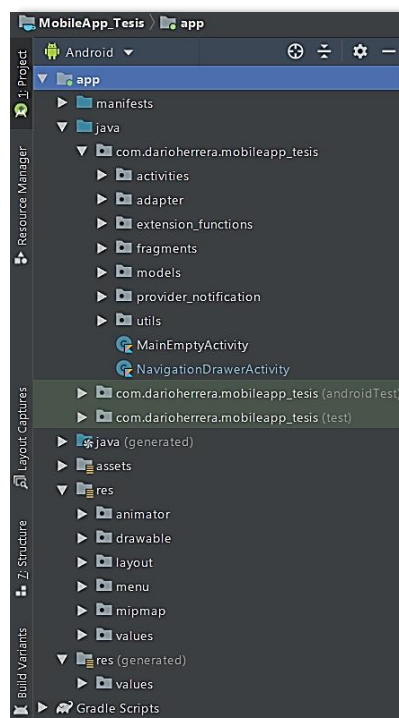


Figura 79. *Estructura del proyecto en Android Studio*

A continuación, se describe la estructura que brinda soporte a la ejecución del aplicativo móvil.

- **app:** Carpeta raíz creada por defecto por Android Studio, contiene todas las subcarpetas que forman parte del proyecto.

- **manifest:** Subcarpeta que contiene el archivo de manifiesto del proyecto.
 - **AndriodManifest.xml:** Archivo que describe la información esencial del proyecto y permite la aplicación de configuraciones y permisos.
- **java:** Subcarpeta contenedora de tres directorios para el desarrollo y pruebas de la aplicación.
 - **com.darioherrera.mobileapp_tesis:** Subcarpeta del directorio java generada automáticamente en base al nombre del paquete y aplicación. Contiene toda la lógica de la aplicación a desarrollarse.
 - **com.darioherrera.mobileapp_tesis (andriodTest):** Subcarpeta del directorio java generado automáticamente para la creación de pruebas instrumentadas.
 - **com.darioherrera.mobileapp_tesis (test):** Subcarpeta del directorio java generado automáticamente para la creación de pruebas unitarias.
- **java (generated):** Carpeta contenedora del archivo BuilCofing.java.
- **assets:** Carpeta creada para agrupar recursos JSON que serán usados por la librería Lottie.
- **res:** Carpeta generada automáticamente que contiene subcarpetas para el desarrollo del aplicativo.
 - **animator:** Contiene animaciones creadas en archivos XML.
 - **drawable:** Contiene recursos estáticos como imágenes y formas personalizadas creadas con archivos XML.
 - **layout:** En esta carpeta se encuentran todos los layouts o interfaces de usuario creadas para el desarrollo de la aplicación.
 - **menu:** Carpeta contenedora de archivos XML que definen las interfaces de los menús a ser usados en la app.
 - **mipmap:** Contiene el o los iconos de la aplicación en todas sus resoluciones.
 - **values:** Carpeta que contiene archivos que definen valores constantes a ser usados en el desarrollo del proyecto.
 - **colors.xml:** Archivo que contiene la definición de colores a ser usados en el aplicativo.
 - **dimens.xml:** Contiene la definición de la resolución que adoptará el aplicativo una vez compilado.
 - **strings.xml:** Archivo que contiene las cadenas de texto y traducciones (en caso de ser multilinguaje) usados en el desarrollo de la app.

- **styles.xml:** Fichero que contiene el estilo de la app.
- **Gradle:** Es un fichero generado automáticamente por Android Studio para la compilación de aplicaciones.

Una vez definida la estructura del proyecto en Android Studio, es necesario conocer la estructura de la lógica utilizada para el desarrollo del aplicativo móvil, mismo que se presenta en la Figura 80.

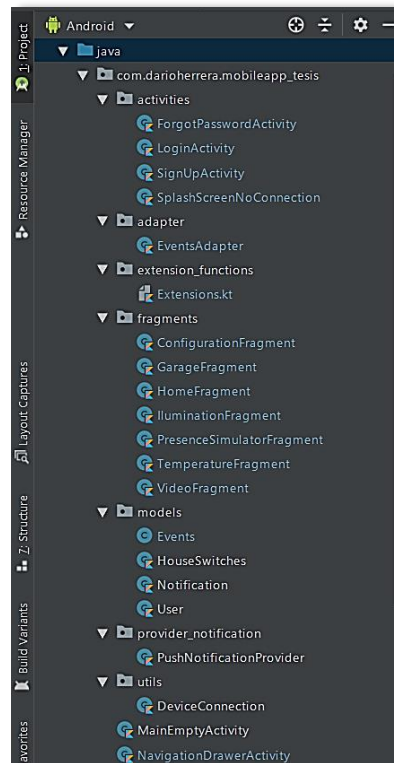


Figura 80. Estructura lógica de la aplicación móvil

A continuación, se describe y detalla cada uno de los directorios y ficheros creados para el desarrollo de la lógica del aplicativo móvil.

- **activities:** Directorio creado para contener ficheros (activities) que permitirán comunicarse y desarrollar la lógica sobre las interfaces.
 - **ForgotPasswordActivity.kt:** Proporciona al usuario la opción de restablecimiento de contraseña en caso de ser olvidada.
 - **LoginActivity.kt:** Provee al usuario los medios para autenticarse e ingresar a las funciones de la app.
 - **SignUpActivity.kt:** Permite al usuario la creación de una cuenta para poder acceder al aplicativo.

- **SplashScreenNoConnection.kt:** Despliega una pantalla de informe en caso de que el usuario intente acceder al aplicativo y este no tenga conexión a internet.
- **adapter:** Carpeta que contiene archivos de tipo adapter.
 - **EventsAdapter.kt:** Archivo que permite la comunicación entre los datos y la vista del componente RecyclerView (usado para mostrar el historial de eventos del hogar).
- **extension_functions:** Contenedor del archivo Extensions.kt.
 - **Extensions.kt:** Fichero que contiene cierto número de funciones implementadas tales como toasts, snackbars, etc., con la finalidad de que el desarrollador ahorre tiempo al momento de escribir código.
- **fragments:** Carpeta contenedora de archivos (fragments) que permitirán establecer comunicación con el componente Navigation Drawer (menú lateral), mismo que desplegará el fragmento correspondiente sobre un activity.
 - **ConfigurationFragment.kt:** Provee al usuario una lista de opciones para configurar parámetros del sistema de seguridad y aplicativo móvil.
 - **GarageFragment.kt:** Permite al usuario abrir o cerrar el garaje del hogar.
 - **HomeFragment.kt:** Proporciona al usuario un historial de todos los eventos o incidentes suscitados en el inmueble.
 - **IlluminationFragment.kt:** Despliega una interfaz que le permite al usuario controlar la iluminación del hogar.
 - **PresenceSimulatorFragment.kt:** Muestra al usuario una interfaz que le permite activar o desactivar un simulador de presencia en caso de que el hogar no se encuentre habitado.
 - **TemperatureFragment.kt:** Pantalla que monitoriza la temperatura del hogar y genera un gráfico de la misma.
 - **VideoFragment.kt:** Permite al usuario acceder a la cámara del sistema de seguridad para monitorear el hogar en formato de video.
- **models:** Directorio que alberga modelos de datos.
 - **Events.kt:** Contiene el modelo de datos que permite mostrar los eventos que se registren en el hogar.
 - **HouseSwitches.kt:** Contiene el modelo de datos de los interruptores del hogar.
 - **Notification.kt:** Contiene el modelo de datos para la recepción de notificaciones push.

- **User.kt:** Contiene el modelo de datos que permite guardar la información de los usuarios registrados.
- **provider_notification:** Contenedor del archivo PushNotificationProvider.kt
 - **PushNotificationProvider.kt:** Permite la generación de tokens y suscripción a un “Topic” en Firebase para que los dispositivos móviles puedan recibir notificaciones push.
- **utils:** Directorio que contiene utilitarios para el desarrollo.
 - **DeviceConnection.kt:** Contiene la funcionalidad para verificar si un dispositivo tiene conexión a internet.
- **MainEmptyActivity.kt:** Punto de partida para la ejecución de la aplicación. Despliega las interfaces según el estado de la sesión del usuario.
- **Navigation DrawerActivity.kt:** Contiene toda la lógica del menú lateral.

Con la estructura y descripción de los componentes, directorios y ficheros que forman parte del aplicativo móvil, a continuación, se presenta la construcción de las interfaces que permitirán al usuario la interacción, ejecución y manejo de acciones del sistema de seguridad.

4.3.5 Construcción de la interfaz para inicio de sesión

La interfaz que se muestra en la Figura 81, es el resultado final de la pantalla que permitirá al usuario el inicio de sesión y autenticación mediante un proveedor de acceso en el aplicativo; *misma que a su vez satisface los requerimientos con identificador R03 y R05, respectivamente.*

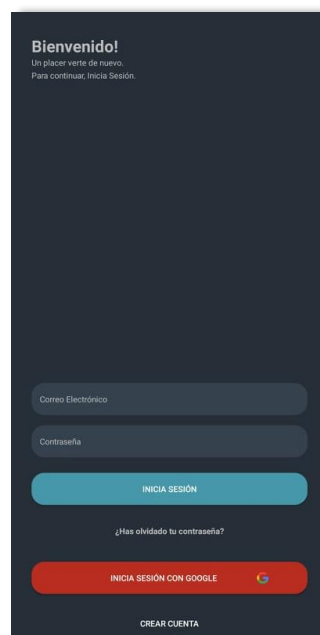


Figura 81. Interfaz para inicio de sesión

En la Tabla 56, se presenta la descripción y funcionalidad de los componentes que forman parte de la pantalla de inicio de sesión.

Tabla 56. Descripción de los componentes de la interfaz de inicio de sesión

ÍTEM	COMPONENTE	ID	DESCRIPCIÓN
1	EditText	txtEmailLogin	Campo de entrada de texto para el ingreso de correo electrónico.
2	EditText	txtPasswordLogin	Campo de entrada de texto para el ingreso de contraseña.
3	Button	btnLogin	Botón que permite el acceso a la aplicación, una vez que las credenciales hayan sido autenticadas.
4	TextView	btnForgotPassword	Botón que permite desplegar la pantalla para restablecer contraseña.
5	Button	btnLoginGoogle	Botón que permite autenticarse para el ingreso a la app, mediante una cuenta de Google.
6	Button	btnCreateAccount	Botón que permite navegar hacia la pantalla para crear una cuenta.
7	ProgressBar	progressBar	Componente que se mostrará para indicar al usuario el procesamiento de la información suministrada.

4.3.6 Construcción de la interfaz para creación de cuenta

La interfaz que se muestra en la Figura 82, es el resultado final de la pantalla que permitirá al usuario la creación de cuenta; *misma que a su vez satisface los requerimientos con identificador R04, R06, R08, R09 y R10.*

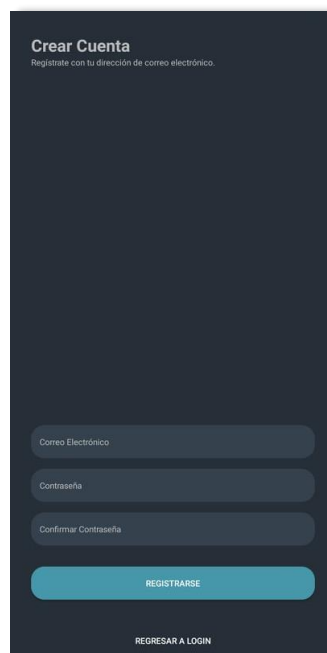


Figura 82. Interfaz para creación de cuenta

En la Tabla 57, se presenta la descripción y funcionalidad de los componentes que forman parte de la pantalla para creación de cuenta.

Tabla 57. Descripción de los componentes de la interfaz para crear cuenta

ÍTEM	COMPONENTE	ID	DESCRIPCIÓN
1	EditText	textEmailAddress	Campo de entrada de texto para el ingreso de correo electrónico.
2	EditText	txtPassword UserSignUp	Campo de entrada de texto para creación de contraseña.
3	EditText	txtConfirmPassword UserSignUp	Campo de entrada de texto para verificación de coincidencia de contraseñas.
4	Button	btnSignUp	Botón que permite la validación de la información proporcionada y él envío de email para confirmación de cuenta.
5	Button	btnGoBackLogin	Botón que permite navegar a la pantalla de inicio de sesión.
6	ProgressBar	progressBarCreateAccount	Componente que se mostrará para indicar al usuario el procesamiento de la información suministrada.

4.3.7 Construcción de la interfaz para restablecimiento de contraseña

La interfaz que se muestra en la Figura 83, es el resultado final de la pantalla que permitirá al usuario el restablecimiento de la contraseña; *misma que a su vez satisface el requerimiento con identificador R07.*

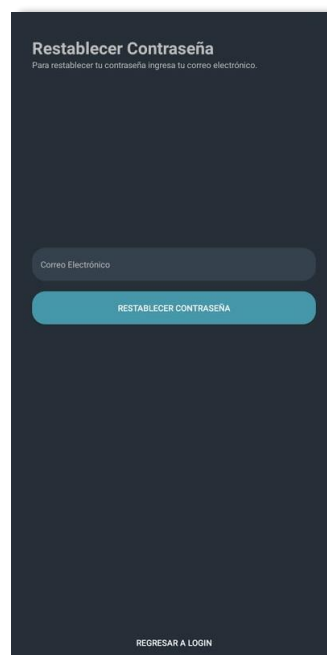


Figura 83. Interfaz para restablecimiento de contraseña

En la Tabla 58, se presenta la descripción y funcionalidad de los componentes que forman parte de la pantalla para restablecer la contraseña.

Tabla 58. Descripción de los componentes de la interfaz para restablecimiento de contraseña

ÍTEM	COMPONENTE	ID	DESCRIPCIÓN
1	EditText	txtEmailUser ResetPassword	Campo de entrada de texto para el ingreso de correo electrónico registrado en la app.
2	Button	btnResetPasswordUser	Botón que permite el envío de un email a la cuenta ingresada con la opción para restablecer la contraseña
3	Button	btnGoBackLogin	Botón que permite navegar a la pantalla de inicio de sesión.
6	ProgressBar	progressBarResetPassword	Componente que se mostrará para indicar al usuario el procesamiento de la información suministrada.

4.3.8 Construcción del Navigation Drawer

La interfaz que se muestra en la Figura 84, es el resultado final del Navigation Drawer o menú lateral que contendrá las diversas opciones del aplicativo móvil.



Figura 84. Interfaz del Navigation Drawer

En la Tabla 59, se presenta la descripción y funcionalidad de los componentes que forman parte del *Navigation Drawer*.

Tabla 59. Descripción de los componentes del Navigation Drawer

ÍTEM	COMPONENTE	ID	DESCRIPCIÓN
1	ImageView	imageViewUser	Componente que permite mostrar recursos de tipo Bitmap o Drawable (imágenes).
2	TextView	txtNameUser	Componente que permite desplegar el nombre del usuario que accedió al aplicativo.
3	TextView	txtUserEmail	Componente que permite desplegar el correo electrónico del usuario que accedió al aplicativo.
4	AppBar	toolbar	Es una barra de título y herramientas situada en la parte superior.
5	Menu-ítem	action_sign_out	Menú que forma parte del AppBar y despliega la opción para cerrar sesión.
6	Gropu-menu-ítem	nav_home	Ítem que forma parte del grupo de opciones del menú lateral. Permite navegar al fragmento que despliega la interfaz para mostrar el historial de eventos del hogar.
7	Gropu-menu-ítem	nav_video	Ítem que forma parte del grupo de opciones del menú lateral. Permite navegar al fragmento que despliega la interfaz para visualizar video en tiempo real.
8	Gropu-menu-ítem	nav_gate_house	Ítem que forma parte del grupo de opciones del menú lateral. Permite navegar al fragmento que despliega la interfaz para abrir o cerrar el garaje.
9	Gropu-menu-ítem	nav_lights	Ítem que forma parte del grupo de opciones del menú lateral. Permite navegar al fragmento que despliega la interfaz para controlar la iluminación del hogar.
10	Gropu-menu-ítem	nav_temperature	Ítem que forma parte del grupo de opciones del menú lateral. Permite navegar al fragmento que despliega la interfaz para monitorear y graficar la temperatura del hogar.
11	Gropu-menu-ítem	nav_presence	Ítem que forma parte del grupo de opciones del menú lateral. Permite navegar al fragmento que despliega la interfaz para activar o desactivar un simulador de presencia.
12	Gropu-menu-ítem	nav_configurations	Ítem que forma parte del grupo de opciones del menú lateral. Permite navegar al fragmento que despliega la interfaz para configuración de parámetros.
13	Gropu-menu-ítem	sign_out	Ítem que forma parte del grupo de opciones del menú lateral. Permite al usuario cerrar sesión.

4.3.9 Construcción de la interfaz para mostrar el historial de eventos

La interfaz que se muestra en la Figura 85, es el resultado final de la pantalla que permitirá mostrar y eliminar los eventos e incidentes suscitados en el hogar; *misma que en conjunto con la interfaz presenta en la Figura 86 satisfacen los requerimientos con identificador R28 y R29.*



Figura 85. Interfaz para mostrar el historial de eventos

En caso de no contar con ningún ítem para mostrar, se desplegará la pantalla mostrada en la Figura 86.

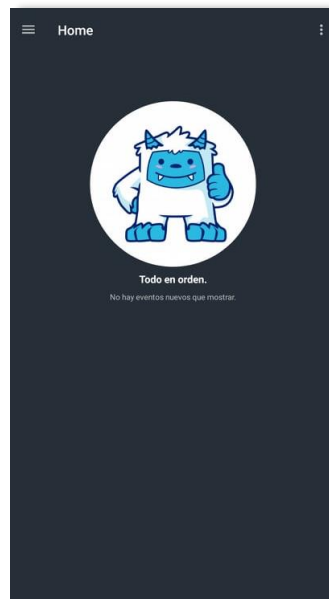


Figura 86. Interfaz que notifica la no existencia de eventos

En la Tabla 60, se presenta la descripción y funcionalidad de los componentes que forman parte de la pantalla para mostrar el historial de eventos.

Tabla 60. Descripción de los componentes de la interfaz para mostrar el historial de eventos

ÍTEM	COMPONENTE	ID	DESCRIPCIÓN
1	RecyclerView	recyclerViewNotifications	Componente que permiten renderizar un template dinámicamente, para la visualización de eventos.
2	CardView	cardView	Contenedor que envuelve el contenido de los eventos.
3	ImageView	imageViewAllItsOkInHouse	Permite mostrar una imagen en caso de no contar con eventos registrados.
4	TextView	txtTitleAllItsOk	Permite mostrar un título en caso de no contar con eventos registrados.
5	TextView	txtAllItsOkDetails	Permite mostrar un detalle en caso de no contar con eventos registrados.
6	TextView	txtTimestampEvent	Permite mostrar la marca de tiempo en la que se suscitó el incidente.
7	TextView	txtDetailEvent	Permite mostrar los detalles del evento o incidente suscitado en el hogar.
8	ImageButton	imageButtonDelete	Botón que permite eliminar un evento.

4.3.10 Construcción de la interfaz para control y visualización de video

La interfaz que se muestra en la Figura 87, es el resultado final de la pantalla que permitirá rotar la cámara que transmite video en tiempo real; *misma que en conjunto con las interfaces presentadas en las Figuras 88 y 89 satisfacen los requerimientos con identificador R16 y R15.*



Figura 87. Interfaz para rotar cámara y visualizar video en tiempo real

En la Figura 88, se presenta la pantalla que se desplegará en caso de que la opción de video se encuentre deshabilitada.



Figura 88. Interfaz que notifica que la opción de video está deshabilitada

Finalmente, en la Figura 89, se presenta la pantalla que se desplegará en caso de que por algún motivo no se pueda establecer conexión para la visualización de video.

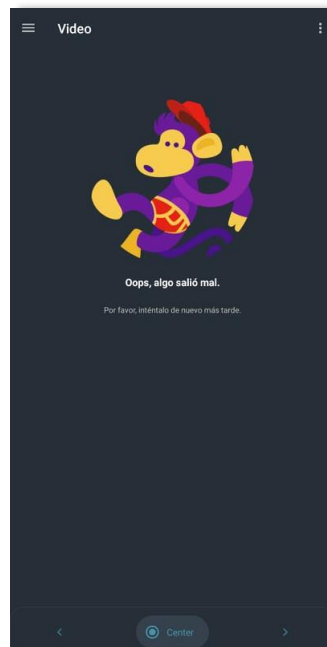


Figura 89. Interfaz que notifica que algo ha salido mal en el despliegue de video

En la Tabla 61, se presenta la descripción y funcionalidad de los componentes que forman parte de la pantalla para la rotación de cámara y visualización de video en tiempo real.

Tabla 61. Descripción de los componentes de la interfaz para rotación y visualización de video

ÍTEM	COMPONENTE	ID	DESCRIPCIÓN
1	WebView	webViewVideo	Componente que permite renderizar una página web. Será el encargado de mostrar el video proveniente de la cámara.
2	Bubbleavigation View	bottonNavigation Constraint	Componente que envuelve a los botones que permitirán rotar la cámara.
3	BubbleToogleView	cameraLeft	Botón que permite rotar la cámara hacia la izquierda.
4	BubbleToogleView	cameraCenter	Botón que permite centrar la cámara.
5	BubbleToogleView	cameraRight	Botón que permitirá rotar la cámara hacia la derecha.
6	ProgressBar	progressBarVideo	Componente que se mostrará para indicar al usuario la carga de los recursos de la página.
7	ImageView	imageViewSomething WentWrong	Componente que desplegará las imágenes según el estado de la solicitud.
8	TextView	txtTitleWentWrong	Permite mostrar el título del fallo en la carga de video.
9	TextView	textViewWent WrongDetails	Permite mostrar los detalles del fallo en la carga de video.

4.3.11 Construcción de la interfaz para control de garaje

La interfaz que se muestra en la Figura 90, es el resultado final de la pantalla que permitirá la apertura y cierre del garaje; *misma que a su vez satisface el requerimiento con identificador R17.*



Figura 90. Interfaz para la apertura y cierre de garaje

En la Tabla 62, se presenta la descripción y funcionalidad de los componentes que forman parte de la pantalla para la apertura y cierre de garaje.

Tabla 62. Descripción de los componentes de la interfaz para la apertura y cierre del garaje

ÍTEM	COMPONENTE	ID	DESCRIPCIÓN
1	LottieAnimationView	displayOpenOrCloseAnimation	Permite desplegar una animación cuando se abre o cierra el garaje.
2	LottieAnimationView	lottieAnimationView	Permite desplegar una animación cuando se accede a la interfaz.
3	ImageView	btnOpenAndCloseGarage	Botón que permite efectuar la apertura o cierre del garaje.
4	TextView	txtTitleStatusGarage	Permite mostrar una cadena de texto.
5	TextView	txtValueStatusGarage	Permite mostrar un texto con el estado actual del garaje.

4.3.12 Construcción de la interfaz para control de iluminación

La interfaz que se muestra en la Figura 91, es el resultado final de la pantalla que permitirá al usuario controlar la iluminación del hogar; *misma que a su vez satisface los requerimientos con identificador R11, R13 y R14.*

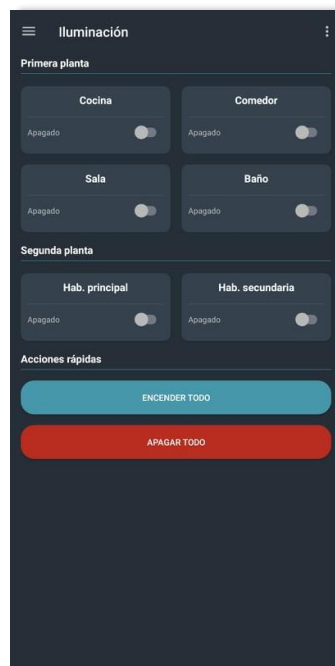


Figura 91. Interfaz para el control de iluminación

En la Tabla 63, se presenta la descripción y funcionalidad de los componentes que forman parte de la pantalla para el control de iluminación.

Tabla 63. Descripción de los componentes de la interfaz para el control de la iluminación

ÍTEM	COMPONENTE	ID	DESCRIPCIÓN
1	CardView	cardView	Contenedor con bordes redondeados que envuelve los componentes para controlar la iluminación.
2	TextView	txtStatusOfSwitch	Permite observar el estado de las lámparas de las diferentes estancias del hogar.
3	Switch	swEnableDisale	Permite activar o desactivar las lámparas de las diferentes estancias del hogar.
4	Button	btnTurnOnAllLamps	Permite activar todas las lámparas del hogar.
5	Button	btnTurnOffAllLamps	Permite desactivar todas las lámparas del hogar.
6	ProgressBar	progressBarIllumination	Componente que se mostrará para indicar al usuario el procesamiento de la tarea solicitada.

4.3.13 Construcción de la interfaz para control y monitoreo de temperatura

La interfaz que se muestra en la Figura 92, es el resultado final de la pantalla que permitirá al usuario obtener información de la temperatura del hogar; *misma que a su vez satisface a los requerimientos con identificador R19 y R20.*

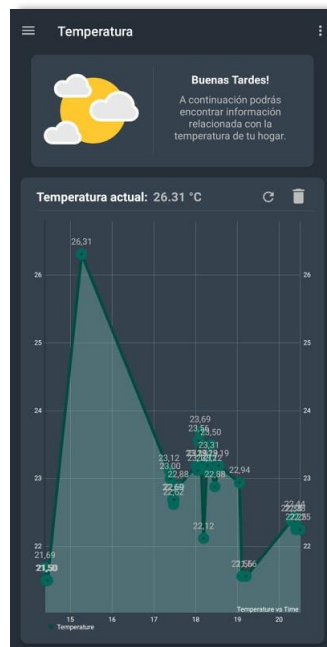


Figura 92. Interfaz para el monitoreo de temperatura

En la Tabla 64, se presenta la descripción y funcionalidad de los componentes que forman parte de la pantalla que proporcionara información sobre la temperatura del hogar.

Tabla 64. Descripción de los componentes de la interfaz para el monitoreo de temperatura

ÍTEM	COMPONENTE	ID	DESCRIPCIÓN
1	CardView	cardViewTimeOf DayUser	Contenedor con bordes redondeados que envuelve los componentes para mostrar el momento del día.
2	LottieAnimationView	lottieAnimationView	Permite desplegar una animación en base al momento del día que el usuario se encuentre.
3	TextView	txtTimeOfDay	Permite observar el momento del día.
4	CardView	cardView ShowTemperature	Contenedor con bordes redondeados que envuelve los componentes para mostrar el gráfico de temperatura.
5	ImageButton	btnRefreshChart Temperature	Botón que permite refrescar o actualizar el gráfico de temperatura.
6	ImageButton	btnDeleteChart Temperature	Botón que permite eliminar el gráfico de temperatura.
7	LineChart	chart	Componente que permite la construcción y despliegue del gráfico de temperatura.
8	TextView	txtShow Temperature	Permite observar la temperatura del hogar en tiempo real.

4.3.14 Construcción de la interfaz para control del simulador de presencia

La interfaz que se muestra en la Figura 93, es el resultado final de la pantalla que permitirá al usuario la activación y desactivación del simulador de presencia; *misma que a su vez satisface el requerimiento con identificador R18.*



Figura 93. Interfaz para el control del simulador de presencia

En la Tabla 65, se presenta la descripción y funcionalidad de los componentes que forman parte de la pantalla para el control del simulador de presencia.

Tabla 65. Descripción de los componentes de la interfaz para activar o desactivar el simulador de presencia

ÍTEM	COMPONENTE	ID	DESCRIPCIÓN
1	LottieAnimationView	displayAnimation SimulatorPresence	Permite desplegar una animación cuando se activa o desactiva el simulador de presencia.
2	ImageView	btnEnableDisable SimulatorPresence	Botón que permite activar o desactivar el simulador de presencia.
3	TextView	txtTitleStatusGarage	Permite mostrar una cadena de texto.
4	TextView	txtValueStatusGarage	Permite mostrar un texto con el estado actual del simulador de presencia.

4.3.15 Construcción de la interfaz para configuraciones

La interfaz que se muestra en la Figura 94, es el resultado final de la pantalla que permitirá al usuario la configuración de los parámetros expuestos en la sección 3 con identificador *R30*, concerniente al apartado de requerimientos iniciales.



Figura 94. Interfaz que permite la configuración de parámetros

En la Tabla 66, se presenta la descripción y funcionalidad de los componentes que forman parte de la pantalla para configuración.

Tabla 66. Descripción de los componentes de la interfaz de configuración

ÍTEM	COMPONENTE	ID	DESCRIPCIÓN
1	Switich	swShowCentigrade	Permite activar o desactivar la visualización de la temperatura en grados centígrados.
2	Switch	swShowFahrenheit	Permite activar o desactivar la visualización de la temperatura en grados fahrenheit.
3	TextView	btnAdjustFanLevel	Permite desplegar un cuadro de diálogo para que el usuario ingrese el valor de temperatura máxima para activar el ventilador del hogar.
4	TextView	btnAdjustHeatingLevel	Permite desplegar un cuadro de diálogo para que el usuario ingrese el valor de temperatura mínima para activar el calefactor del hogar.
5	TextView	btnAdjustLampsIntensity	Permite desplegar un cuadro de diálogo para que el usuario seleccione el nivel de intensidad de las lámparas del hogar.
6	Switch	swEnableDisable MagneticSensor	Permite activar o desactivar la detección de apertura de puertas.
7	Switch	swEnableRealTime Video	Permite activar o desactivar la visualización de video en tiempo real.
8	Switch	swMotionDetection	Permite activar o desactivar la detección de movimiento.
9	TextView	txtShowFan Temperature	Permite mostrar el valor de temperatura máxima para activar el ventilador.
10	TextView	txtShowHeater Temperature	Permite mostrar el valor de temperatura mínima para activar el calefactor.

4.3.16 CONSTRUCCIÓN DE LOS CUADROS DE DIÁLOGO

4.3.16.1 Cuadro de diálogo en la interfaz de historial de eventos

La interfaz que se muestra en la Figura 95, es el resultado final del cuadro de diálogo que se desplegará en la interfaz de historial de eventos; *mismo forma parte de la solución al requerimiento con identificador R29.*



Figura 95. Cuadro de diálogo confirmar eliminación de evento.

En la Tabla 67, se presenta la descripción y funcionalidad del componente del cuadro de diálogo.

Tabla 67. Descripción del componente del cuadro de diálogo para confirmación de eliminación de evento

ÍTEM	COMPONENTE	DESCRIPCIÓN
1	Material Dialog Core	Componente que se despliega en la interfaz de historial de eventos cuando el usuario presiona el botón de eliminar, preguntando al usuario la confirmación de eliminación.

4.3.16.2 Cuadro de diálogo en la interfaz de simulador de presencia

La interfaz que se muestra en la Figura 96, es el resultado final del cuadro de diálogo que se desplegará en la interfaz de simulador de presencia; *mismo forma parte de la solución al requerimiento con identificador R18.*

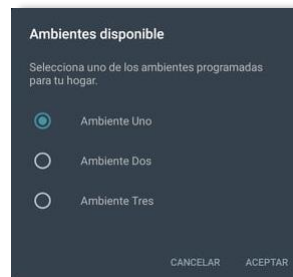


Figura 96. Cuadro de diálogo para selección de ambientes.

En la Tabla 68, se presenta la descripción y funcionalidad del componente del cuadro de diálogo.

Tabla 68. Descripción del componente del cuadro de diálogo para selección de ambientes

ÍTEM	COMPONENTE	DESCRIPCIÓN
1	Material Dialog Core	Componente que se despliega en la interfaz de simulador de presencia para que el usuario elija uno de los tres ambientes programados.

4.3.16.3 Cuadro de diálogo en la interfaz de temperatura

La interfaz que se muestra en la Figura 97, es el resultado final del cuadro de diálogo que se desplegará en la interfaz de temperatura; *mismo forma parte de la solución al requerimiento con identificador R19.*

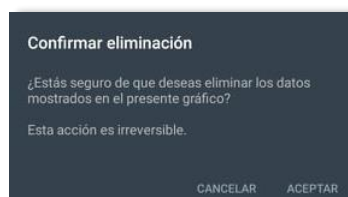


Figura 97. Cuadro de diálogo confirmación de eliminación de gráfico de temperatura.

En la Tabla 69, se presenta la descripción y funcionalidad del componente del cuadro de diálogo.

Tabla 69. Descripción del componente del cuadro de diálogo para confirmación de eliminación del gráfico de temperatura

ÍTEM	COMPONENTE	DESCRIPCIÓN
1	Material Dialog Core	Componente que se despliega en la interfaz de temperatura cuando el usuario presiona el botón de eliminar, preguntando al usuario la confirmación de eliminación.

4.3.16.4 Cuadros de diálogo en la interfaz de configuración

4.3.16.4.1 Cuadro de diálogo para establecer niveles de activación

Las interfaces que se muestran en las Figuras 98 y 99, son el resultado final de los cuadros de diálogo para establecer los niveles de activación tanto del ventilador como del calefactor; *mismos que forman parte de la solución al requerimiento con identificador R30.*



Figura 98. Cuadro de diálogo para establecer nivel de activación del ventilador



Figura 99. Cuadro de diálogo para establecer nivel de activación del calefactor

En la Tabla 70, se presenta la descripción y funcionalidad del componente del cuadro de diálogo.

Tabla 70. Descripción del componente del cuadro de diálogo para establecer niveles de activación

ÍTEM	COMPONENTE	DESCRIPCIÓN
1	Material Dialog Input	Componente que se despliega cuando el usuario presiona en las opciones para ajuste de nivel de temperatura del ventilador y calefactor, pidiendo al usuario el ingreso de valor límite o mínimo.

4.3.16.4.2 Cuadro de diálogo para establecer niveles de intensidad

La interfaz que se muestra en la Figura 100, es el resultado final del cuadro de diálogo para seleccionar el nivel de intensidad de las lámparas. *mismo forma parte de la solución al requerimiento con identificador R30.*

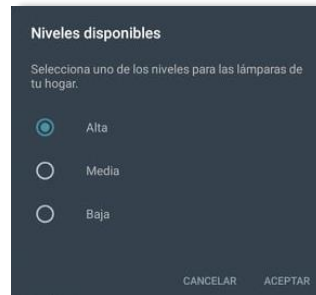


Figura 100. Cuadro de diálogo para establecer nivel de intensidad

En la Tabla 71, se presenta la descripción y funcionalidad del componente.

Tabla 71. Descripción del componente del cuadro de diálogo para ajustar nivel de intensidad de las lámparas

ÍTEM	COMPONENTE	DESCRIPCIÓN
1	Material Dialog Core	Componente que se despliega cuando el usuario presiona en la opción para ajuste de nivel de intensidad de las lámparas para que el usuario elija una de las tres opciones.

4.3.17 Construcción de la interfaz que notifica la no existencia de conexión

La interfaz que se muestra en la Figura 101, es el resultado final de la pantalla que permitirá notificar al usuario la no existencia de una conexión a internet; *característica extra a las solicitadas en los requerimientos iniciales para un mejor control del sistema.*

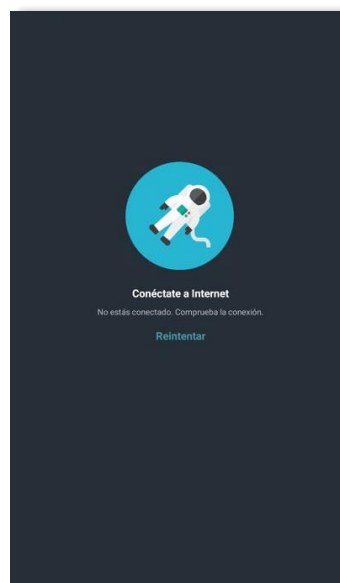


Figura 101. Interfaz que notifica la usencia de una red de internet

En la Tabla 72, se presenta la descripción y funcionalidad de los componentes que forman parte de la pantalla que proporcionará información sobre el evento expuesto.

Tabla 72. Descripción de los componentes para notificación de la no existencia de una conexión a internet

ÍTEM	COMPONENTE	ID	DESCRIPCIÓN
1	ImageView	imageViewNo InternetConnection	Componente que permite desplegar el recurso mostrado en la Figura 101.
2	TextView	btnTryAgain	Botón que permite verificar y navegar hacia las funcionalidades de la app si se establece conexión con una red.

4.4 CONSTRUCCIÓN DEL HARDWARE

En esta sección se presenta la construcción del hardware que dará soporte al sistema de seguridad y que a su vez se comunicará con el aplicativo móvil para el monitoreo y ejecución de tareas de manera remota.

Para el desarrollo y construcción de los componentes de hardware se hizo uso del software para diseño de circuitos electrónicos Proteus en conjunto con la librería gpiozero, misma que sirvió como intermediario para la comunicación entre los dispositivos electrónicos (sensores y actuadores) y las interfaces GPIO de Raspberry Pi.

4.4.1 Construcción del circuito para lectura de temperatura

El mercado ofrece una gran variedad de soluciones electrónicas capaces de entregar información concerniente a la temperatura, sin embargo, en este caso se ha optado por el uso del sensor DS18B20, dispositivo económico, fácil de usar y que es capaz de transformar la temperatura en voltaje con una precisión considerable.

Para la lectura de la información proveniente del sensor, Raspberry Pi deberá tener habilitada la interfaz del protocolo I2C, mediante la cual se podrá establecer comunicación con el dispositivo externo.

Las especificaciones del sensor DS18B20 [70], se muestran en la Tabla 73, mientras que el circuito electrónico que brinda soporte para la lectura de temperatura se expone en la Figura 102; mismo que forma parte de la solución al requerimiento con identificador **R19**.

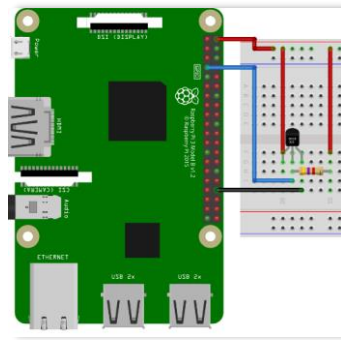


Figura 102. Circuito electrónico para la lectura de temperatura

Tabla 73. Especificaciones y características del sensor DS18B20

ESPECIFICACIONES	
Voltaje de operación	3V – 5V
Rango de lectura de temperatura	-55°C a +125°C
Precisión	± 0.5 °C
Características	
<ul style="list-style-type: none"> - La interfaz requiere únicamente de un solo cable para poner en funcionamiento al pin de lectura y comunicación. - Permite la conversión de 12 bits a dato digital en aproximadamente 750 ms. 	

4.4.2 Construcción del circuito para la detección de gases

El aire que circula dentro de la vivienda constituye un factor vital para los miembros de la misma, por esta razón es importante contar con un mecanismo que permita medir, monitorear o detectar sustancias que propicien la contaminación del aire que se respira. Frente a ello, el mercado ofrece una gran variedad de dispositivos, sin embargo, en este caso se ha optado por el uso del sensor de gases MQ-135, elemento económico, fácil de usar, con capacidad para la detección de sustancias como: NH₃, NO_x, alcohol, benceno, humo, CO₂.

Las especificaciones del sensor MQ-135 [71], se muestran en la Tabla 74, mientras que el circuito electrónico que brinda soporte para la detección de gases nocivos se expone en la Figura 103; mismo que forma parte de la solución al requerimiento con identificador **R25**.

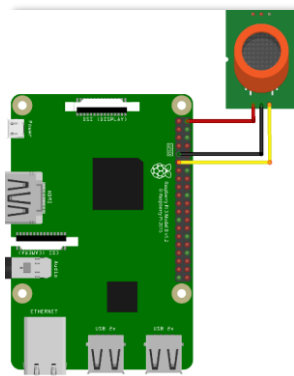


Figura 103. Circuito electrónico para la detección de gases

Tabla 74. Especificación y características del sensor MQ-135

ESPECIFICACIONES	
Voltaje de operación	5V
Resistencia de carga	Ajustable
Resistencia de calentador	33Ω±5%
Resistencia de detección	30KΩ - 200KΩ (100ppm NH3)
Características	
<ul style="list-style-type: none"> - Amplio alcance de detección. - Respuesta rápida y alta sensibilidad. - Dispositivo estable y de larga duración. - Circuito de accionamiento simple. - Lectura análoga y digital. - Capaz de detectar NH3, NOx, alcohol, benceno, humo, CO2, entre otros. 	

4.4.3 Construcción del circuito para la detección de movimiento

La detección de movimiento dentro de una vivienda puede ser usada como mecanismo para la protección tanto de los miembros del hogar como de sus bienes. Al contar con esta característica un hogar es menos propenso a robos, ya que provee la toma de acciones de manera inmediata y de forma automatizada.

El sensor usado para llevar a cabo dicha tarea, es el PIR HC-SR501, mismo que permite la detección de movimiento basado en la medición de luz infrarroja que irradian los objetos en su área de acción.

Las especificaciones del sensor PIR HC-SR501 [72], se muestran en la Tabla 75, mientras que el circuito electrónico que brinda soporte para la detección de gases nocivos se expone en la Figura 104; mismo que forma parte de la solución a los requerimientos con identificador **R21** y **R22**.

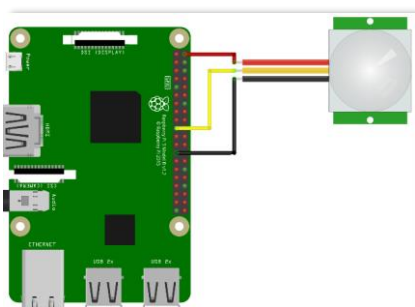


Figura 104. Circuito electrónico para la detección de movimiento

Tabla 75. Especificaciones y características del sensor PIR HC-SR501

ESPECIFICACIONES	
Voltaje de operación	5V – 12V
Consumo promedio	<1 mA
Rango de distancia	3m a 7m (ajustable)
Ángulo de detección	Cono de 110°

Tiempo de inicialización	Alrededor de 1 minuto desde el momento de su alimentación.
Temperatura de operación	-15° a +70° C
Características	
<ul style="list-style-type: none"> - Rango de detección y tiempo de alarma activa ajustable mediante un potenciómetro. - Jumper para configurar la salida en modo mono disparo o disparo repetitivo. - Salida de alarma de movimiento con ajuste de tiempo entre 3 segundos a 5 minutos. 	

4.4.4 Construcción del circuito para el control del ventilador

El ventilador actuará como mediador para la regulación de la temperatura del hogar en caso de que esta se encuentre elevada, así como también permitirá la extracción de aire contaminado en caso de detectar alguna sustancia nociva en el medio ambiente del inmueble.

En su interior el ventilador está conformado por un motor de corriente continua o también denominado motor DC, elemento electrónico que cuenta con dos terminales para su activación. Raspberry Pi cuenta con pines que pueden ser configurados como salida, sin embargo, el voltaje que proporcionan (entre 2V a 3.3V) no es suficiente para activar un motor DC. Por ello es necesario incorporar un circuito adicional que satisfaga los voltajes de operación.

Las especificaciones del ventilador se muestran en la Tabla 76, mientras que el circuito electrónico que soporta la activación del mismo se expone en la Figura 105; *mismo que forma parte de la solución al requerimiento con identificador R20.*

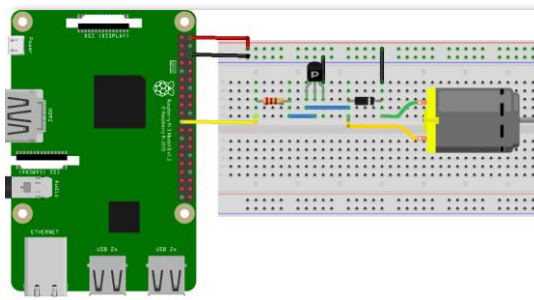


Figura 105. Circuito electrónico para el control del ventilador

Tabla 76. Especificaciones del ventilador

ESPECIFICACIONES	
Voltaje de operación	5V
Amperaje	0.25A
Revoluciones por minuto	8000
Dimensiones	60x60x25

4.4.5 Construcción del circuito para control del calefactor y buzzer

El calefactor dentro del hogar al igual que el ventilador actuará como mediador para regular la temperatura, de modo que, si los niveles son bajos, de forma automática se proceda con la activación del calefactor, que en este caso será representado por un diodo led, ya que no se cuenta con el electrodoméstico real.

Por otra parte, el buzzer será utilizado como una alarma sonora, que se activará cuando se detecte tanto la presencia de gases nocivos en el ambiente como la apertura de una puerta del hogar, permitiendo de esta forma alertar a los miembros del hogar.

Las especificaciones del buzzer [73], se muestran en la Tabla 77, mientras que el circuito electrónico que soporta la activación de los dispositivos se expone en la Figura 106; *mismo que forma parte de la solución a los requerimientos con identificador R20 y R27.*

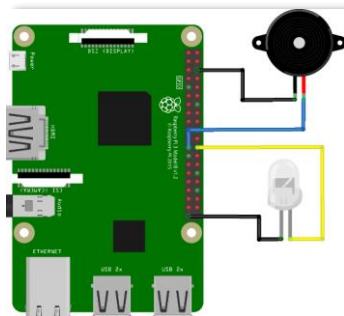


Figura 106. Circuito electrónico para la activación del calefactor y buzzer

Tabla 77. Especificaciones del buzzer

ESPECIFICACIONES	
Voltaje de operación	3.3V a 5V
Consumo de corriente	< 30mA
Salida de sonido	> 85dB
Frecuencia generada	2.3 KHz \pm 400 Hz
Temperatura de operación	-40° a 85° C

4.4.6 Construcción del circuito para detección de apertura de puertas

El mercado ofrece un sin número de dispositivos para la detección de apertura de puertas y ventanas, tanto a nivel de viviendas como a nivel industrial. En este sentido, para la incorporación de dicha característica dentro del hogar, se ha elegido y hecho uso del sensor magnético MC-38, mismo que está conformado por dos encapsulados que contienen un imán y un sensor, respectivamente.

El sensor es muy fácil de usar, ya que genera señales de salida en base al estado del circuito (abierto o cerrado), esto se logra mediante la posición que tomen los encapsulados, es decir, si estos se encuentran uno cerca del otro, generan un estado cerrado, mientras

que, si estos se separan, generan un estado abierto. De esta forma en base a dicho estado se puede detectar la apertura de una puerta o ventana.

Las especificaciones del sensor MC-38 [74], se muestran en la Tabla 78, mientras que el circuito electrónico que soporta la detección de apertura de puertas se expone en la Figura 107; *mismo que forma parte de la solución al requerimiento con identificador R24.*

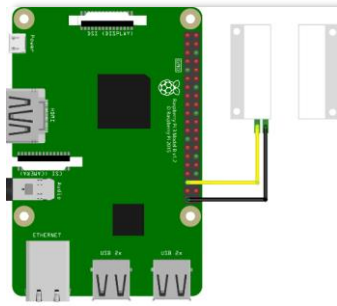


Figura 107. Circuito electrónico para la detección de apertura de puertas

Tabla 78. Especificaciones del sensor magnético MC-38

ESPECIFICACIONES	
Voltaje de operación	3.3V a 100V
Amperaje	0.5A
Distancia de activación	15 a 25mm
Material de construcción	Plástico
Dimensiones	34x41x6.5mm

4.4.7 Construcción del circuito para el control de servomotores

El sistema de seguridad incorporará dos micros servomotores SG90, el primero será usado para controlar la rotación de la cámara con la finalidad de abarcar un rango de visión más amplio, mientras que el segundo será usado para la apertura y cierre del garaje del hogar.

El funcionamiento de un servomotor es diferente a la de un motor DC tradicional. Su principio de operación está basado en la generación de movimiento mecánico mediante una señal PWM (Pulse With Modulation) o modulación por ancho de pulso, misma que permite tener un control preciso de la posición que adoptará un servomotor, característica principal que lo diferencia de un motor tradicional.

Raspberry Pi cuenta con soporte para la generación de señales PWM, sin embargo, dicha señal es generada con una temporización de tipo software, lo que produce pulsos no tan precisos, traduciéndose a movimientos continuos del servomotor incluso cuando la señal ha terminado su ciclo.

Para dar solución al problema expuesto, se ha incorporado el driver Mosfet IRF520, usado generalmente en circuitos con dispositivos que requieran manejar cargas superiores a 2A.

El principio de operación, se basa en la recepción de una señal en estado high (1 lógico), que activa el Gate del mosfet del circuito, que a su vez permite el flujo de corriente para la activación del dispositivo en cuestión [75].

Con el uso del mecanismo expuesto, se solventan los movimientos continuos en el servomotor, ya que el driver será el encargado de proporcionar la carga al mismo, de modo que cuando el ciclo de la señal PWM termine, una señal de tipo low (0 lógico) será enviada al driver, cortando el paso de corriente, lo cual desactivará al servomotor.

Las especificaciones del driver IRF520 [76], y del servomotor SG90 [77], se muestran en la Tabla 79, mientras que en las Figura 108 y 109 se exponen los circuitos que dan soporte para el control de los servomotores que permitirán tanto la rotación de la cámara como la apertura y cierre del garaje, respectivamente; *mismos que forma parte de la solución a los requerimientos con identificador R16 y R17.*

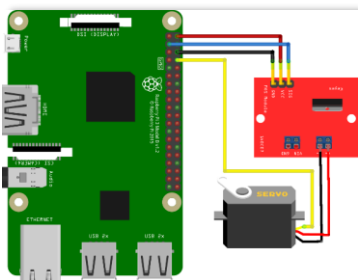


Figura 108. Circuito electrónico para control de servomotor para rotara cámara

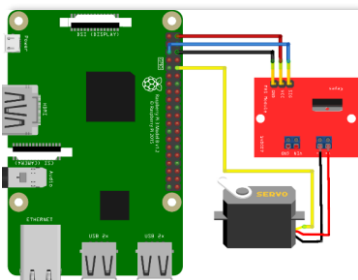


Figura 109. Circuito electrónico para control de servomotor de garaje

Tabla 79. Especificaciones del driver IRF520 y del micro servo SG90

ESPECIFICACIONES DRIVER MOSFET IRF520	
Voltaje de control	5V TTL
Voltaje de salida	0 a 24V
Corriente nominal	6A
Corriente máxima	9A
ESPECIFICACIONES MICRO SERVO SG90	
Voltaje de operación	3 – 7.2V
Velocidad	0.10 sec/60° @ 4.8V
Torque	1.8 Kg-cm @ 4.8V
Ángulo de rotación	180°
Ancho de pulso	500-2400 µs

4.4.8 Construcción del circuito para detección de luminosidad

Mediante el uso de una fotorresistencia o también denominada LDR, se puede sensar la cantidad de luz tanto artificial como natural. El sensor varía el estado de su resistencia en base a la cantidad de luz que este recibe, es decir, a mayor luz, menor resistencia y viceversa.

Con la incorporación de este mecanismo es posible gestionar el control de la iluminación del hogar, lo que supone un ahorro energético y reducción de costos para el usuario. En este caso el área a automatizar será el alumbrado externo del inmueble, con lo cual al llegar la noche se active de manera automática y al llegar el día lo contrario.

Las especificaciones de la LDR [78], se muestran en la Tabla 80, mientras que el circuito electrónico que brinda soporte para la detección de luminosidad, se expone en la Figura 110; *mismo que forma parte de la solución al requerimiento con identificador R25.*

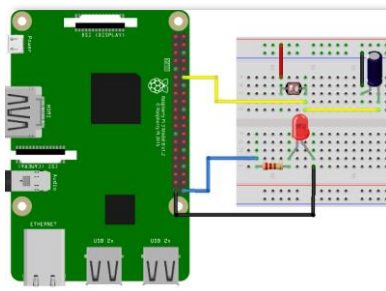


Figura 110. Circuito electrónico para la detección de luminosidad

Tabla 80. Especificaciones de la LDR

ESPECIFICACIONES	
Resistencia en luz (10 lux)	8K Ω - 20 K Ω
Resistencia en oscuridad	1M Ω
Voltaje máximo	150V
Material fotosensible	Sulfato de sodio

4.4.9 Construcción del circuito de potencia para iluminación

Como se expuso en el Capítulo III, sección 3.8.11.1, es necesario que el circuito que provee de potencia y voltaje a las lámparas del hogar, sea construido en una placa de circuito impreso, esto con la finalidad de asegurar que los componentes sean inmunes al movimiento, obteniendo así un circuito de calidad, duradero y menos propenso a errores de funcionamiento.

La placa de circuito impreso (PCB) que se presenta en la Figura 111, *forma parte de la solución al requerimiento con identificador R12*; la misma opera en base a la recepción de señales proveniente de Raspberry Pi, si una señal es recibida, esta es procesada por el

optoacoplador, mismo que cierra el circuito y emite una señal hacia el triac, que a su vez permite que el voltaje de corriente alterna (VCA) fluya en el circuito y de este modo se activen las lámparas o cualquier dispositivo que opere con voltajes de 110 VCA.



Figura 111. Placa de circuito impreso PCB para potencia de iluminación

En la Tabla 81, se exponen los elementos electrónicos utilizados para la construcción del PCB.

Tabla 81. Componentes utilizados para la construcción del PCB para potencia de iluminación

COMPONENTES	CANTIDAD
Baquelita	1
Optoacoplador MOC3021	8
Triac BT136	8
Bornera	9
Resistencia 220 Ω	16
Regleta hembra	1

4.4.10 Construcción del circuito para control de iluminación

El circuito para control de iluminación, al igual que el circuito de potencia para iluminación, necesita la construcción de una placa de circuito impreso, esto debido a que el usuario deberá poder acceder a los pulsadores del circuito, por tal razón es necesario que todos los componentes se encuentren inmunes al movimiento, obteniendo un así un circuito de calidad, duradero y menos propenso a errores de funcionamiento.

La placa de circuito impreso para control de iluminación se presenta en la Figura 112, la misma forma parte de la solución al requerimiento con identificador **R12** y su modo de operación está basado en el envío y recepción de señales. El envío se lo realiza desde los pulsadores hacia Raspberry Pi, misma que procesa la información con la ayuda de los controladores de hardware (expuestos en la sección 4.6), que a su vez generan una salida o respuesta que es recibida por el PCB para la activación o desactivación de las lámparas según el valor de la respuesta obtenido.



Figura 112. Placa de circuito impreso para control de lámparas

En la Tabla 82, se exponen los elementos electrónicos utilizados para la construcción del PCB.

Tabla 82. Componentes utilizados para la construcción del PCB

COMPONENTES	CANTIDAD
Baquelita	1
Pulsador	6
Condensador cerámico 0.1uf	6
Diodo led	6
Resistencia 220 Ω	6
Resistencia 10k Ω	6
Regleta macho	1

4.4.11 Instalación de cámara en Raspberry Pi

Para la transmisión de video y captura de fotografías se hará uso del módulo *Pi Camera Night Vision*, misma que forma parte de la solución a los requerimientos con identificador **R15** y **R22**. Entre las características del presente módulo se puede destacar:

- Visión nocturna
- Lente ajustable de 3.6mm
- Ángulo de visión de 75.7 grados
- Resolución de 5MP
- Grabación de video a 30fps en HD 1080p



Figura 113. Raspberry Pi cámara con visión nocturna

Para acceder a los recursos y funcionalidades de la cámara de Raspberry Pi, únicamente es necesario conectar el Flex al puerto CSI tanto de Raspberry Pi como del módulo de la cámara tal como se muestra en la Figura 114.



Figura 114. Instalación de la cámara en Raspberry Pi

4.5 CONSTRUCCIÓN DE LOS CONTROLADORES DE HARDWARE

En esta sección se presenta la construcción de los controladores que darán soporte al funcionamiento de los elementos de hardware que forman parte del sistema de seguridad.

Los controladores serán desarrollados con el lenguaje de programación Python, esto debido a que el lenguaje es muy versátil y compatible con todas las herramientas que dan soporte a la arquitectura del sistema.

Los controladores o scripts, permitirán procesar la información que recibe o envía Raspberry Pi hacia los sensores o actuadores, respectivamente. Estos se compilarán y ejecutarán sobre el sistema operativo Raspbian, mismo que contará con los SDK de Firebase, que a su vez permitirán que los scripts ejecuten acciones sobre la base de datos en tiempo real.

4.6 Integración del SDK de Firebase a Python

Como punto inicial, es necesario integrar los SDK de Firebase a la plataforma de hardware Raspberry Pi, misma que por defecto trae consigo Python.

Con Python instalado por defecto, el siguiente paso consiste en la instalación del paquete Firebase-admin mediante el uso del administrador de paquetes para Python (pip):

- *\$ sudo pip install requests*
- *\$ sudo pip install python-firebase*

Una vez que la instalación de los paquetes de la librería Firebase-admin hayan finalizado, es necesario generar una clave privada en Firebase, esto con la finalidad de poder acceder a todos los servicios que ofrece la plataforma desde Python.

Para la generación de la clave privada, basta con acceder a la configuración del proyecto, ítem *cuentas de servicio* y en ella seleccionar la plataforma de uso, que en este caso es Python, tal como se muestra en la Figura 115.

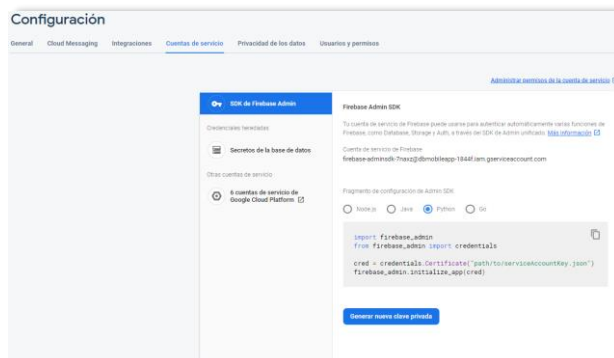


Figura 115. Generación de clave privada en Firebase para Python

Una vez que se haya seleccionado la plataforma de uso, se procede a presionar el botón *Generar nueva clave privada*, éste a su vez desplegará un modal con un mensaje de advertencia y con la opción para genera la clave privada que será guardada en un archivo con formato JSON, tal como se presenta en la Figura 116.

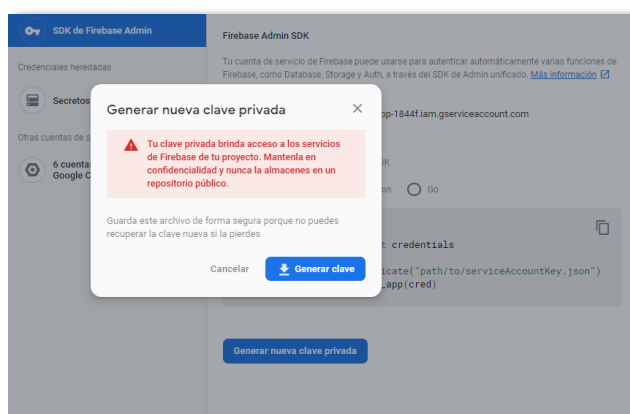


Figura 116. Generación del archivo que contendrá la clave privada

Al terminar la descarga, el fichero debe ser colocado en un lugar seguro, esto ya que mediante el, se puede acceder a todos los servicios que brinda Firebase.

Finalmente, con las instalaciones expuestas y el archivo en un lugar seguro, Firebase queda totalmente operable en Raspberry Pi.

4.6.1 Estructura del proyecto en Python

En esta sección se presenta la estructura del proyecto en Python que da soporte al funcionamiento de los controladores de hardware, misma que se expone en la Figura 117. Dichos controladores permitirán satisfacer en conjunto todos los requerimientos relacionados con el hardware, presentados en las secciones anteriores.

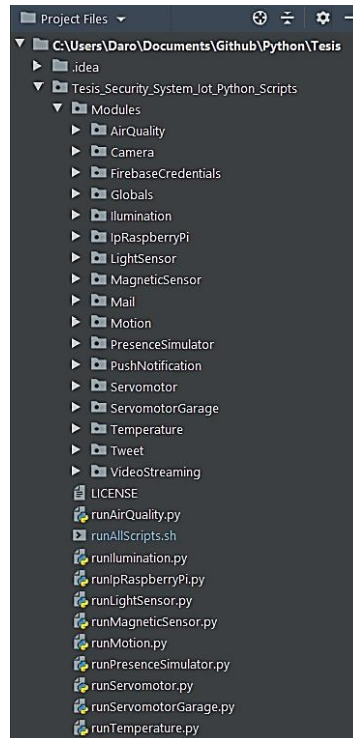


Figura 117. Estructura del proyecto para los controladores de hardware en Python

Con una visión general de la estructura del proyecto, a continuación, se describe y detalla la funcionalidad de cada uno de los directorios y ficheros creados para el control de los diversos sensores y actuadores presentes en el sistema de seguridad.

- **Modules:** Directorio raíz que agrupa lógicamente el código de los controladores de hardware.
 - **AirQuality:** Subdirectorio que alberga el archivo `air_quality.py`.
 - **air_quality.py:** Fichero que contiene la lógica para la detección de gases nocivos mediante el sensor MQ-135.
 - **Camera:** Subdirectorio contenedor del archivo `air_quality.py`.
 - **capture_picture.py:** Fichero que permite acceder a los recursos de la cámara para capturar una fotografía.
 - **FirebaseCredentials:** Subdirectorio que contiene ficheros para la conexión de Python con Firebase.
 - **credenciales_python.json:** Archivo generado por Firebase que contiene las credenciales para el acceso desde Python a los SDK de Firebase.
 - **firebase_credentials.py:** Fichero que contiene la lógica para establecer comunicación entre Firebase y Python.
 - **Globals:** Subdirectorio que alberga el archivo `globals_var.py`.

- **globals_var.py:** Fichero que contiene variables constantes usadas en los distintos scripts de Python.
- **Illumination:** Subdirectorio que contiene el archivo illumination.py.
 - **illumination.py:** Fichero que contiene toda la lógica para el encendido y apagado de las lámparas del hogar mediante los pulsadores de la placa electrónica.
- **IpRaspberryPi:** Subdirectorio contenedor del archivo get_and_post_ip_raspberry.py.
 - **get_and_post_ip_raspberry.py:** Fichero que permite la obtención y guardado de la dirección IP de Raspberry Pi en Firebase.
- **LightSensor:** Subdirectorio contenedor del archivo light_sensor.py.
 - **light_sensor.py:** Fichero que contiene toda la lógica para el control del sensor de luminosidad (LDR).
- **MagneticSensor:** Subdirectorio que alberga el archivo magnetic_sensor.py.
 - **magnetic_sensor.py:** Fichero que contiene toda la lógica para el control del sensor magnético.
- **Mail:** Subdirectorio que contiene el archivo send_email.py.
 - **send_email.py:** Fichero que permite el envío de emails.
- **Motion:** Subdirectorio contenedor de los archivos motion.py y motion_model.py.
 - **motion.py:** Fichero que contiene toda la lógica para el control del sensor de movimiento PIR.
 - **motion_model.py:** Fichero que contiene el modelo de datos para guardar información sobre incidentes o eventos suscitados en hogar.
- **PresenceSimulator:** Subdirectorio que contiene el archivo presence_simulator.py.
 - **presence_simulator.py:** Fichero que contiene toda la lógica para el control del simulador de presencia.
- **PushNotification:** Subdirectorio que alberga el archivo push_notification_firestore_cloud_messaging.py.
 - **push_notification_firestore_cloud_messaging.py:** Fichero que permite el envío de notificaciones push a los dispositivos móviles Android.
- **Servomotor:** Subdirectorio que contiene el archivo servomotor.py.

- **servomotor.py:** Fichero que permite el control del servomotor de la cámara.
- **ServomotorGarage:** Subdirectorio que contiene el archivo `servomotor_garage.py`.
 - **servomotor_garage.py:** Fichero que permite el control del servomotor para la apertura y cierre del garaje.
- **Temperature:** Subdirectorio contenedor de los archivos `temperatura.py` y `temperatura_chart_model.py`.
 - **temperatura.py:** Fichero que permite el control del sensor de temperatura.
 - **temperatura_chart_model.py:** Fichero que contiene el modelo de datos para guardar información de la temperatura del hogar.
- **Tweet:** Subdirectorio que alberga los archivos `auth.py` y `post_tweet.py`.
 - **auth.py:** Fichero que contiene las credenciales de la cuenta de Twitter Developers.
 - **post_tweet.py:** Fichero que permite interactuar con la API de Twitter para el posteo de incidentes en la red social.
- **VideoStreaming:** Subdirectorio que contiene los recursos para la transmisión de video.
 - **static:** Directorio que contiene el archivo `style.css`.
 - **style.css:** Fichero que permite dar estilo a la página web `index.html`.
 - **templates:** Directorio que contiene el archivo `index.html`.
 - **index.html:** Fichero usado para desplegar video en una página web.
 - **set_pi_camera.py:** Fichero en el cual se establecen los parámetros de la cámara de Raspberry Pi para la transmisión de video.
 - **video_streaming.py:** Fichero que contiene la lógica para el despliegue de video en tiempo real en una página web.
 - **runVideoStreaming.sh:** Script en bash que permite la ejecución del programa en Python para video streaming.
- **runAirQuality.py:** Script que permite la ejecución del programa para la detección de gases nocivos.
- **runIllumination.py:** Script que permite la ejecución del programa para el control de iluminación.

- **runIpRaspberryPi.py:** Script que permite la ejecución del programa para la obtención de la dirección IP de Raspberry Pi.
- **runLightSensor.py:** Script que permite la ejecución del programa para el control del sensor de luminosidad.
- **runMagneticSensor.py:** Script que permite la ejecución del programa para la detección de apertura de puertas.
- **runMotion.py:** Script que permite la ejecución del programa para la detección de movimiento.
- **runPresenceSimulator.py:** Script que permite la ejecución del programa para el control del simulador de presencia.
- **runServomotor.py:** Script que permite la ejecución del programa para el control del servomotor de la cámara.
- **runServomotorGarage.py:** Script que permite la ejecución del programa para la apertura y cierre del garaje.
- **runTemperature.py:** Script que permite la ejecución del programa para el control del sensor de temperatura.
- **runAllScripts.sh:** Script en bash que permite la ejecución de todos los programas creados en Python para el control del hardware.

4.7 IMPLEMENTACIÓN DEL SISTEMA DE SEGURIDAD

El sistema de seguridad será implementado en base al prototipo presentado en el Capítulo III, sección 3.2. Los materiales de construcción serán acrílico transparente para la vivienda y madera MDF para la base, tal como se muestra en la Figura 118; *mismo que satisface al requerimiento con identificador R32.*



Figura 118. Vivienda prototipo para la implementación del sistema de seguridad para control domótico

Con la vivienda prototipo lista, se procederá a la instalación de todos los componentes de hardware que forman parte del sistema de seguridad, mismos que se detallan en la Tabla 83.

Tabla 83. Componentes de hardware a instalarse en la vivienda prototipo

ÍTEM	COMPONENTES	CANTIDAD
1	Raspberry Pi 3 B+	1
2	Switch de encendido Raspberry Pi	1
3	Fuente de alimentación para Raspberry Pi	1
4	Placa de circuito impreso para control de iluminación	1
5	Placa de circuito impreso para potencia de iluminación	1
6	Boquilla	7
7	Lámpara 110VCA 60W	7
8	Diodo Led	2
9	Sensor de calidad de aire MQ-135	1
10	Sensor de temperatura DS18B20	1
11	Sensor de luminosidad LDR	1
12	Sensor magnético	1
13	Sensor de movimiento PIR HC-SR501	1
14	Micro servo SG90	2
15	Módulo mosfet IRF520	2
16	Ventilador 5V	1
17	Cámara para Raspberry Pi	1
18	Flex para cámara de Raspberry Pi	1
19	Buzzer 5V	1
20	Mini protoboard	1
21	Cable de alimentación para circuito de potencia de iluminación	1
22	Jumpers macho y hembra	100

En la Figura 119, se presenta el resultado final de la implementación de todos los componentes que forman parte del sistema de seguridad. Las instalaciones de los diversos elementos se encuentran ocultas en la base que reposa la vivienda prototipo.



Figura 119. Implementación del sistema de seguridad en la vivienda prototipo

CAPÍTULO V

5. PRUEBAS Y RESULTADOS

En esta sección se presentan las pruebas funcionales y resultados ejecutados sobre el sistema de seguridad prototipo para control domótico, mismos que se basan en los requerimientos establecidos en el Capítulo 3, sección 3.3.1.

Cada prueba será presentada en una Tabla, esto con el fin de detallar y mostrar de mejor manera los parámetros a ser evaluados y los resultados obtenidos, con los cuales se pueda generar un resultado final basado en una escala cualitativa (Excelente, Bueno, Regular, Malo) que obedece a los criterios establecidos en la Tabla 84.

Tabla 84. Criterios de la escala cualitativa para pruebas y resultados

Valor	Criterio
Malo	Interfaz de control y/o componentes del sistema no ejecutan <i>ninguna acción o la realizan parcialmente</i> con un retardo o <i>delay alto</i> (mayor o igual a 26 segundos).
Regular	Interfaz de control y/o componentes del sistema, se comportan de acuerdo a lo establecido y ejecutan acciones con un retardo o <i>delay medio alto</i> (entre 16 a 25 segundos).
Bueno	Interfaz de control y/o componentes del sistema, se comportan de acuerdo a lo establecido y ejecutan acciones con un retardo o <i>delay medio bajo</i> (entre 6 a 15 segundos).
Excelente	Interfaz de control y/o componentes del sistema, se comportan de acuerdo a lo establecido y ejecutan acciones con un retardo o <i>delay bajo</i> (entre 1 a 5 segundos).

5.1 Pruebas funcionales

5.1.1 Creación y autenticación de cuentas a través de correo electrónico

En la Tabla 85, se presenta la prueba efectuada en la interfaz del aplicativo móvil que permite la creación de cuentas de usuario a través del correo electrónico, para la posterior autenticación y acceso a la misma.

Tabla 85. P01_Prueba para la creación y autenticación de cuentas a través de correo electrónico

TIPO	DESCRIPCIÓN
Identificador	P01
Nombre de la prueba	Creación y autenticación de cuentas a través de correo electrónico.
Descripción de la prueba	Prueba que permitirá la creación de una cuenta a través de una dirección de correo electrónico.
Pre-condiciones	<ul style="list-style-type: none">• Aplicativo móvil instalado en un smartphone.• Smartphone conectado a una red Wi-Fi o datos móviles.• Cuenta de correo electrónico.

Resultado esperado	a) El aplicativo móvil enviará de manera automática un email de verificación a la cuenta proporcionada para otorgar el acceso a la misma. b) Si el usuario no verifica su cuenta de email no podrá acceder a las funciones del aplicativo. c) Una vez que se haya verificado la cuenta, el usuario podrá acceder a las funcionalidades del aplicativo.
USUARIO 1	
Resultado obtenido	a) SI / NO / PARCIALMENTE b) SI / NO / PARCIALMENTE c) SI / NO / PARCIALMENTE
Resultado final	Excelente
Estado	Finalizada
USUARIO 2	
Resultado obtenido	a) SI / NO / PARCIALMENTE b) SI / NO / PARCIALMENTE c) SI / NO / PARCIALMENTE
Resultado final	Excelente
Estado	Finalizada
USUARIO 3	
Resultado obtenido	a) SI / NO / PARCIALMENTE b) SI / NO / PARCIALMENTE c) SI / NO / PARCIALMENTE
Resultado final	Excelente
Estado usuario	Finalizada

5.1.2 Creación y autenticación de cuentas a través de un proveedor de acceso

En la Tabla 86, se presenta la prueba efectuada para la creación y autenticación de cuentas a través de Google como proveedor de acceso.

Tabla 86. P02_Prueba para la creación y autenticación de cuentas a través de un proveedor de acceso

TIPO	DESCRIPCIÓN
Identificador	P02
Nombre de la prueba	Creación y autenticación de cuentas a través de un proveedor de acceso.
Descripción de la prueba	Prueba que permitirá la creación y autenticación de cuentas a través de Google como proveedor de acceso.
Pre-condiciones	<ul style="list-style-type: none"> • Aplicativo móvil instalado en un smartphone. • Smartphone conectado a una red Wi-Fi o datos móviles. • Cuenta de Google.
Resultado esperado	a) El aplicativo móvil desplegará un modal que permitirá al usuario seleccionar una de las cuentas de Google registradas en el smartphone, con la cual podrá acceder a las funciones de la app.
USUARIO 1	
Resultado obtenido	a) SI / NO / PARCIALMENTE
Resultado final	Excelente

Estado	Finalizada
USUARIO 2	
Resultado obtenido	a) SI / NO / PARCIALMENTE
Resultado final	Excelente
Estado	Finalizada
USUARIO 3	
Resultado obtenido	a) SI / NO / PARCIALMENTE
Resultado final	Excelente
Estado usuario	Finalizada

5.1.3 Validación de campos de los formularios

Las Tabla 87, 88 y 89, presentan las pruebas de validación efectuadas a los formularios de las interfaces para inicio de sesión, creación de cuenta y restablecimiento de contraseña.

Tabla 87. P03_ Prueba para la validación de los campos del formulario para inicio de sesión

TIPO	DESCRIPCIÓN
Identificador	P03
Nombre de la prueba	Validación de campos del formulario de inicio de sesión.
Descripción de la prueba	Prueba que permitirá la validación de los campos del formulario de la interfaz para inicio de sesión.
Pre-condiciones	<ul style="list-style-type: none"> • Aplicativo móvil instalado en un smartphone. • Smartphone conectado a una red Wi-Fi o datos móviles.
Resultado esperado	a) Al ingresar un email no válido, la app desplegará un mensaje en tiempo real, informando al usuario lo ocurrido. b) Al ingresar una contraseña que no cumpla con el formato establecido, la app desplegará un mensaje en tiempo real información al usuario lo ocurrido. c) Al ingresar datos erróneos no se podrá acceder a las funciones de la app.
USUARIO 1	
Resultado obtenido	a) SI / NO / PARCIALMENTE b) SI / NO / PARCIALMENTE c) SI / NO / PARCIALMENTE
Resultado final	Excelente
Estado	Finalizada
USUARIO 2	
Resultado obtenido	a) SI / NO / PARCIALMENTE b) SI / NO / PARCIALMENTE c) SI / NO / PARCIALMENTE
Resultado final	Excelente
Estado	Finalizada
USUARIO 3	
Resultado obtenido	a) SI / NO / PARCIALMENTE b) SI / NO / PARCIALMENTE c) SI / NO / PARCIALMENTE
Resultado final	Excelente
Estado usuario	Finalizada

Tabla 88. P04_Prueba para la validación de los campos del formulario para creación de cuenta

TIPO	DESCRIPCIÓN
Identificador	P04
Nombre de la prueba	Validación de campos del formulario para creación de cuentas.
Descripción de la prueba	Prueba que permitirá la validación de los campos del formulario de la interfaz para creación de cuenta.
Pre-condiciones	<ul style="list-style-type: none"> • Aplicativo móvil instalado en un smartphone. • Smartphone conectado a una red Wi-Fi o datos móviles.
Resultado esperado	a) Al ingresar un email no válido, la app desplegará un mensaje en tiempo real, informando al usuario lo ocurrido. b) Al ingresar una contraseña que no cumpla con el formato establecido, la app desplegará un mensaje en tiempo real informando al usuario lo ocurrido. c) Al ingresar una contraseña que no coincida con la proporcionada en primera instancia, la app desplegará un mensaje en tiempo real informando lo ocurrido. d) La app no permitirá la creación de cuenta si se proporcionan datos erróneos.
USUARIO 1	
Resultado obtenido	a) SI / NO / PARCIALMENTE b) SI / NO / PARCIALMENTE c) SI / NO / PARCIALMENTE d) SI / NO / PARCIALMENTE
Resultado final	Excelente
Estado	Finalizada
USUARIO 2	
Resultado obtenido	a) SI / NO / PARCIALMENTE b) SI / NO / PARCIALMENTE c) SI / NO / PARCIALMENTE d) SI / NO / PARCIALMENTE
Resultado final	Excelente
Estado	Finalizada
USUARIO 3	
Resultado obtenido	a) SI / NO / PARCIALMENTE b) SI / NO / PARCIALMENTE c) SI / NO / PARCIALMENTE d) SI / NO / PARCIALMENTE
Resultado final	Excelente
Estado usuario	Finalizada

Tabla 89. P05_Prueba para la validación de los campos del formulario para restablecimiento de contraseña

TIPO	DESCRIPCIÓN
Identificador	P05
Nombre de la prueba	Validación de campos del formulario para restablecimiento de contraseña.
Descripción de la prueba	Prueba que permitirá la validación de los campos del formulario de la interfaz para restablecimiento de contraseña.
Pre-condiciones	<ul style="list-style-type: none"> • Aplicativo móvil instalado en un smartphone. • Smartphone conectado a una red Wi-Fi o datos móviles.

Resultado esperado	a) Al ingresar un email no válido, la app desplegará un mensaje en tiempo real, informando al usuario lo ocurrido.
USUARIO 1	
Resultado obtenido	a) SI / NO / PARCIALMENTE
Resultado final	Excelente
Estado	Finalizada
USUARIO 2	
Resultado obtenido	a) SI / NO / PARCIALMENTE
Resultado final	Excelente
Estado	Finalizada
USUARIO 3	
Resultado obtenido	a) SI / NO / PARCIALMENTE
Resultado final	Excelente
Estado usuario	Finalizada

5.1.4 Restablecimiento de contraseña

En la Tabla 90, se presenta la prueba efectuada sobre la interfaz del aplicativo móvil que permite el restablecimiento de contraseña de la cuenta creada para el acceso a la app en caso de ser olvidada.

Tabla 90. P06_Prueba para restablecimiento de contraseña

TIPO	DESCRIPCIÓN
Identificador	P06
Nombre de la prueba	Restablecimiento de contraseña.
Descripción de la prueba	Prueba que permitirá el restablecimiento de contraseña de la cuenta creada en caso de ser olvidada.
Pre-condiciones	<ul style="list-style-type: none"> • Aplicativo móvil instalado en un smartphone. • Smartphone conectado a una red Wi-Fi o datos móviles. • Cuenta válida para el acceso a las funciones de la app.
Resultado esperado	a) El aplicativo móvil solicitará la dirección de correo electrónico con la que se creó la cuenta, misma a la que será enviado un email para el restablecimiento de la contraseña. b) Con las nuevas credenciales, el usuario podrá acceder a las funciones de la app.
USUARIO 1	
Resultado obtenido	a) SI / NO / PARCIALMENTE b) SI / NO / PARCIALMENTE
Resultado final	Bueno
Estado	Finalizada
USUARIO 2	
Resultado obtenido	a) SI / NO / PARCIALMENTE b) SI / NO / PARCIALMENTE
Resultado final	Excelente
Estado	Finalizada

USUARIO 3	
Resultado obtenido	a) SI / NO / PARCIALMENTE b) SI / NO / PARCIALMENTE
Resultado final	Bueno
Estado usuario	Finalizada

5.1.5 Control de iluminación

En la Tabla 91, se muestran las pruebas efectuadas para el control de iluminación desde el aplicativo móvil y la placa de circuito impreso.

Tabla 91. P07_Prueba para el control de iluminación

TIPO	DESCRIPCIÓN
Identificador	P07
Nombre de la prueba	Control de iluminación
Descripción de la prueba	Prueba para el control de iluminación del hogar desde el aplicativo móvil y la placa de circuito impreso.
Pre-condiciones	<ul style="list-style-type: none"> • Aplicativo móvil instalado en un smartphone. • Smartphone conectado a una red de internet vía Wi-Fi o datos móviles. • Ingreso a las funciones de la app opción Iluminación. • Raspberry Pi conectada a la fuente de alimentación. • Raspberry Pi conectada a una red de internet vía Wi-Fi o ethernet. • Cable de poder de las lámparas conectadas a una fuente de alimentación. • Script de Python para control de iluminación ejecutándose.
Resultado esperado	<ol style="list-style-type: none"> a) En la interfaz para control de iluminación del aplicativo móvil existirán interruptores digitales de cada estancia del hogar que permitirán activar o desactivar las lámparas. b) En la interfaz para control de iluminación del aplicativo móvil existirá un botón para activar todas las lámparas del hogar. c) En la interfaz para control de iluminación del aplicativo móvil existirá un botón para desactivar todas las lámparas del hogar. d) Al presionar los interruptores digitales, las lámparas del hogar deberán prenderse o apagarse según su estado. e) Todas las lámparas del hogar deberán activarse al presionar el botón con dicha funcionalidad. f) Todas las lámparas del hogar deberán desactivarse al presionar el botón con dicha funcionalidad. g) La interfaz para control de iluminación adoptará el valor actual de las lámparas en cuestión. h) La placa de circuito impreso contará con pulsadores físicos para cada estancia del hogar, permitiendo la activación y desactivación de las lámparas. i) Al presionar los pulsadores físicos, las lámparas deberán prenderse o apagarse según su estado. j) Los interruptores digitales del aplicativo móvil, se sincronizarán según el estado de las lámparas.

USUARIO 1	
Resultado obtenido	a) SI / NO / PARCIALMENTE b) SI / NO / PARCIALMENTE c) SI / NO / PARCIALMENTE d) SI / NO / PARCIALMENTE e) SI / NO / PARCIALMENTE f) SI / NO / PARCIALMENTE g) SI / NO / PARCIALMENTE h) SI / NO / PARCIALMENTE i) SI / NO / PARCIALMENTE j) SI / NO / PARCIALMENTE
Resultado final	Excelente
Estado	Finalizada
USUARIO 2	
Resultado obtenido	a) SI / NO / PARCIALMENTE b) SI / NO / PARCIALMENTE c) SI / NO / PARCIALMENTE d) SI / NO / PARCIALMENTE e) SI / NO / PARCIALMENTE f) SI / NO / PARCIALMENTE g) SI / NO / PARCIALMENTE h) SI / NO / PARCIALMENTE i) SI / NO / PARCIALMENTE j) SI / NO / PARCIALMENTE
Resultado final	Bueno
Estado	Finalizada
USUARIO 3	
Resultado obtenido	a) SI / NO / PARCIALMENTE b) SI / NO / PARCIALMENTE c) SI / NO / PARCIALMENTE d) SI / NO / PARCIALMENTE e) SI / NO / PARCIALMENTE f) SI / NO / PARCIALMENTE g) SI / NO / PARCIALMENTE h) SI / NO / PARCIALMENTE i) SI / NO / PARCIALMENTE j) SI / NO / PARCIALMENTE
Resultado final	Excelente
Estado usuario	Finalizada

5.1.6 Visualización de video en tiempo real

En la Tabla 92, se presenta la prueba efectuada para la visualización de video en tiempo real desde el aplicativo móvil.

Tabla 92. P08_Prueba para la visualización de video en tiempo real

TIPO	DESCRIPCIÓN
Identificador	P08
Nombre de la prueba	Visualización de video en tiempo real.
Descripción de la prueba	Prueba que permitirá la visualización de video en tiempo real desde el aplicativo móvil.
Pre-condiciones	<ul style="list-style-type: none"> • Aplicativo móvil instalado en un smartphone. • Smartphone conectado a una red de internet vía Wi-Fi o datos móviles. • Habilitar la función de video en tiempo real en Configuración > Video. • Ingreso a las funciones de la app opción Video. • Raspberry Pi conectada a la fuente de alimentación. • Raspberry Pi conectada a una red de internet vía Wi-Fi o ethernet. • Script de Python para transmisión de video ejecutándose.
Resultado esperado	a) El aplicativo móvil contará con una interfaz que permitirá observar la transmisión de video del hogar en tiempo real. b) Al acceder a la opción Video se desplegará la transmisión en tiempo real.
USUARIO 1	
Resultado obtenido	a) SI / NO / PARCIALMENTE b) SI / NO / PARCIALMENTE
Resultado final	Bueno
Estado	Finalizada
USUARIO 2	
Resultado obtenido	a) SI / NO / PARCIALMENTE b) SI / NO / PARCIALMENTE
Resultado final	Regular
Estado	Finalizada
USUARIO 3	
Resultado obtenido	a) SI / NO / PARCIALMENTE b) SI / NO / PARCIALMENTE
Resultado final	Malo
Estado	Finalizada

5.1.7 Control de la rotación de cámara

En la Tabla 93, se presenta la prueba efectuada para el control de la rotación de la cámara de video del hogar.

Tabla 93. P09_Prueba para control de la rotación de cámara

TIPO	DESCRIPCIÓN
Identificador	P09
Nombre de la prueba	Control de la rotación de la cámara.
Descripción de la prueba	Prueba que permitirá la rotación de la cámara de video con un grado de libertad.
Pre-condiciones	<ul style="list-style-type: none"> • Aplicativo móvil instalado en un smartphone. • Smartphone conectado a una red de internet vía Wi-Fi o datos móviles. • Ingreso a las funciones de la app opción Video. • Raspberry Pi conectada a la fuente de alimentación. • Raspberry Pi conectada a una red de internet vía Wi-Fi o ethernet. • Script de Python para control de servomotor ejecutándose.
Resultado esperado	a) El aplicativo móvil contará con una interfaz que permitirá rotar la cámara en las direcciones: izquierda, centro y derecha. b) Al presionar el botón opción izquierda, se controlará el servomotor para rotar la cámara hacia dicha dirección. c) Al presionar el botón opción centro, se controlará el servomotor para rotar la cámara hacia dicha dirección. d) Al presionar el botón opción derecha, se controlará el servomotor para rotar la cámara hacia dicha dirección.
USUARIO 1	
Resultado obtenido	a) SI / NO / PARCIALMENTE b) SI / NO / PARCIALMENTE c) SI / NO / PARCIALMENTE d) SI / NO / PARCIALMENTE
Resultado final	Excelente
Estado	Finalizada
USUARIO 2	
Resultado obtenido	a) SI / NO / PARCIALMENTE b) SI / NO / PARCIALMENTE c) SI / NO / PARCIALMENTE d) SI / NO / PARCIALMENTE
Resultado final	Excelente
Estado	Finalizada
USUARIO 3	
Resultado obtenido	a) SI / NO / PARCIALMENTE b) SI / NO / PARCIALMENTE c) SI / NO / PARCIALMENTE d) SI / NO / PARCIALMENTE
Resultado final	Excelente
Estado	Finalizada

5.1.8 Control de apertura y cierre del garaje

En la Tabla 94, se presenta la prueba efectuada para el control de la apertura y cierre del garaje de la vivienda.

Tabla 94. P10_Prueba para el control de apertura y cierre del garaje

TIPO	DESCRIPCIÓN
Identificador	P10
Nombre de la prueba	Control del garaje.
Descripción de la prueba	Prueba que permitirá controlar la apertura y cierre del garaje de la vivienda prototipo.
Pre-condiciones	<ul style="list-style-type: none"> • Aplicativo móvil instalado en un smartphone. • Smartphone conectado a una red de internet vía Wi-Fi o datos móviles. • Ingreso a las funciones de la app opción Garaje. • Raspberry Pi conectada a la fuente de alimentación. • Raspberry Pi conectada a una red de internet vía Wi-Fi o ethernet. • Script de Python para control de servomotor ejecutándose.
Resultado esperado	<p>a) El aplicativo móvil contará con una interfaz que permitirá la apertura y cierre del garaje del hogar.</p> <p>b) Al presionar el botón central de la interfaz se controlará el servomotor para la apertura o cierre del garaje.</p> <p>c) Al presionar el botón central la interfaz desplegará una animación y un texto con el estado actual del garaje (Abierto o Cerrado).</p>
USUARIO 1	
Resultado obtenido	<p>a) SI / NO / PARCIALMENTE</p> <p>b) SI / NO / PARCIALMENTE</p> <p>c) SI / NO / PARCIALMENTE</p>
Resultado final	Excelente
Estado	Finalizada
USUARIO 2	
Resultado obtenido	<p>a) SI / NO / PARCIALMENTE</p> <p>b) SI / NO / PARCIALMENTE</p> <p>c) SI / NO / PARCIALMENTE</p>
Resultado final	Excelente
Estado	Finalizada
USUARIO 3	
Resultado obtenido	<p>a) SI / NO / PARCIALMENTE</p> <p>b) SI / NO / PARCIALMENTE</p> <p>c) SI / NO / PARCIALMENTE</p>
Resultado final	Excelente
Estado	Finalizada

5.1.9 Control del simulador de presencia

En la Tabla 95, se presenta la prueba efectuada para el control del simulador de presencia del hogar.

Tabla 95. P11_Prueba para el control del simulador de presencia

TIPO	DESCRIPCIÓN
Identificador	P11
Nombre de la prueba	Control del simulador de presencia.
Descripción de la prueba	Prueba que permitirá controlar el simulador de presencia.
Pre-condiciones	<ul style="list-style-type: none"> • Aplicativo móvil instalado en un smartphone. • Smartphone conectado a una red de internet vía Wi-Fi o datos móviles. • Ingreso a las funciones de la app opción Simulador de presencia. • Raspberry Pi conectada a la fuente de alimentación. • Raspberry Pi conectada a una red de internet vía Wi-Fi o ethernet. • Cable de poder de las lámparas conectadas a una fuente de alimentación. • Script de Python para control del simulador de presencia ejecutándose.
Resultado esperado	<ol style="list-style-type: none"> a) El aplicativo móvil contará con una interfaz que permitirá la activación y desactivación del simulador de presencia. b) Al presionar el botón central se desplegará una cuadro de diálogo que permitirá seleccionar uno de los tres ambientes programados. c) Al seleccionar uno de los tres ambientes programados, se activarán las lámparas del hogar con una secuencia determinada. d) Al presionar el botón central se desactivar el simulador de presencia. e) Al presionar el botón central se apagarán las lámparas del hogar. f) Al presionar el botón central de desplegar una animación y un texto con el estado actual del simulador de presencia (Activado o Desactivado).
USUARIO 1	
Resultado obtenido	<ol style="list-style-type: none"> a) SI / NO / PARCIALMENTE b) SI / NO / PARCIALMENTE c) SI / NO / PARCIALMENTE d) SI / NO / PARCIALMENTE e) SI / NO / PARCIALMENTE f) SI / NO / PARCIALMENTE
Resultado final	Regular
Estado	Finalizada
USUARIO 2	
Resultado obtenido	<ol style="list-style-type: none"> a) SI / NO / PARCIALMENTE b) SI / NO / PARCIALMENTE c) SI / NO / PARCIALMENTE d) SI / NO / PARCIALMENTE e) SI / NO / PARCIALMENTE f) SI / NO / PARCIALMENTE
Resultado final	Bueno
Estado	Finalizada

USUARIO 3	
Resultado obtenido	a) SI / NO / PARCIALMENTE b) SI / NO / PARCIALMENTE c) SI / NO / PARCIALMENTE d) SI / NO / PARCIALMENTE e) SI / NO / PARCIALMENTE f) SI / NO / PARCIALMENTE
Resultado final	Bueno
Estado	Finalizada

5.1.10 Monitoreo y control de temperatura

En la Tabla 96, se presenta la prueba efectuada para el monitoreo y control de la temperatura interna del hogar.

Tabla 96. P12_Prueba para el monitoreo y control de temperatura

TIPO	DESCRIPCIÓN
Identificador	P12
Nombre de la prueba	Monitoreo y control de temperatura.
Descripción de la prueba	Prueba que permitirá controlar y monitorear la temperatura interna del hogar.
Pre-condiciones	<ul style="list-style-type: none"> • Aplicativo móvil instalado en un smartphone. • Smartphone conectado a una red de internet vía Wi-Fi o datos móviles. • Elección de la unidad de medida de temperatura a ser mostrada (Grados Centígrados o Fahrenheit) en Configuración > Temperatura. • Establecimiento de niveles máximo y mínimo de temperatura para la activación del ventilador y calefactor en Configuración > Temperatura. • Ingreso a las funciones de la app opción Temperatura. • Raspberry Pi conectada a la fuente de alimentación. • Raspberry Pi conectada a una red de internet vía Wi-Fi o ethernet. • Script de Python para control y monitoreo de temperatura ejecutándose.
Resultado esperado	<ol style="list-style-type: none"> a) El aplicativo móvil contará con una interfaz que desplegará una animación y texto en base a las partes del día. b) El aplicativo móvil desplegará en la interfaz la temperatura actual del hogar en Grados Centígrados o Fahrenheit según la configuración. c) El aplicativo móvil construirá y desplegará en la interfaz una gráfica de la variación de la temperatura con el tiempo. d) La interfaz contará con dos botones para refrescar o borrar la gráfica de temperatura. e) Según los niveles máximos y mínimos de temperatura establecidos se activarán o desactivarán de forma automática el ventilador y calefactor.

	f) Una notificación push será enviada al aplicativo móvil indicando la activación o desactivación del ventilador y calefactor.
USUARIO 1	
Resultado obtenido	a) SI / NO / PARCIALMENTE b) SI / NO / PARCIALMENTE c) SI / NO / PARCIALMENTE d) SI / NO / PARCIALMENTE e) SI / NO / PARCIALMENTE f) SI / NO / PARCIALMENTE
Resultado final	Bueno
Estado	Finalizada
USUARIO 2	
Resultado obtenido	a) SI / NO / PARCIALMENTE b) SI / NO / PARCIALMENTE c) SI / NO / PARCIALMENTE d) SI / NO / PARCIALMENTE e) SI / NO / PARCIALMENTE f) SI / NO / PARCIALMENTE
Resultado final	Bueno
Estado	Finalizada
USUARIO 3	
Resultado obtenido	a) SI / NO / PARCIALMENTE b) SI / NO / PARCIALMENTE c) SI / NO / PARCIALMENTE d) SI / NO / PARCIALMENTE e) SI / NO / PARCIALMENTE f) SI / NO / PARCIALMENTE
Resultado final	Excelente
Estado	Finalizada

5.1.11 Detección de movimiento

En la Tabla 97, se presenta la prueba efectuada para la detección de movimiento dentro del hogar.

Tabla 97. P13_ Prueba para la detección de movimiento

TIPO	DESCRIPCIÓN
Identificador	P13
Nombre de la prueba	Detección de movimiento.
Descripción de la prueba	Prueba que permitirá detectar movimiento dentro del hogar.
Pre-condiciones	<ul style="list-style-type: none"> • Aplicativo móvil instalado en un smartphone. • Smartphone conectado a una red de internet vía Wi-Fi o datos móviles. • Habilitar la función de detección de movimiento en Configuración > Detección de movimiento. • Raspberry Pi conectada a la fuente de alimentación. • Raspberry Pi conectada a una red de internet vía Wi-Fi o ethernet.

	<ul style="list-style-type: none"> Script de Python para detección de movimiento ejecutándose.
Resultado esperado	a) Al detectar movimiento el sistema de seguridad tomará una fotografía, misma que será enviada vía email y posteada en la red social Twitter. b) El aplicativo móvil recibirá una notificación push del incidente.
USUARIO 1	
Resultado obtenido	a) SI / NO / PARCIALMENTE b) SI / NO / PARCIALMENTE
Resultado final	Excelente
Estado	Finalizada
USUARIO 2	
Resultado obtenido	a) SI / NO / PARCIALMENTE b) SI / NO / PARCIALMENTE
Resultado final	Bueno
Estado	Finalizada
USUARIO 3	
Resultado obtenido	a) SI / NO / PARCIALMENTE b) SI / NO / PARCIALMENTE
Resultado final	Bueno
Estado	Finalizada

5.1.12 Detección de apertura de puertas

En la Tabla 98, se presenta la prueba efectuada para la detección de apertura de puertas de la vivienda.

Tabla 98. P14_ Prueba para la detección de apertura de puertas

TIPO	DESCRIPCIÓN
Identificador	P14
Nombre de la prueba	Detección de apertura de puertas.
Descripción de la prueba	Prueba que permitirá la detección de apertura de puertas de la vivienda.
Pre-condiciones	<ul style="list-style-type: none"> Aplicativo móvil instalado en un smartphone. Smartphone conectado a una red de internet vía Wi-Fi o datos móviles. Habilitar la función de Apertura de puertas en Configuración > Detección de movimiento. Raspberry Pi conectada a la fuente de alimentación. Raspberry Pi conectada a una red de internet vía Wi-Fi o ethernet. Cable de poder de las lámparas conectadas a una fuente de alimentación. Script de Python para detección de apertura de puertas ejecutándose.
Resultado esperado	a) Al detectar la apertura de la puerta del hogar, el sistema de seguridad activará una alarma sonora informando el incidente.

	b) El aplicativo móvil recibirá una notificación push del incidente.
USUARIO 1	
Resultado obtenido	a) SI / NO / PARCIALMENTE b) SI / NO / PARCIALMENTE
Resultado final	Excelente
Estado	Finalizada
USUARIO 2	
Resultado obtenido	a) SI / NO / PARCIALMENTE b) SI / NO / PARCIALMENTE
Resultado final	Excelente
Estado	Finalizada
USUARIO 3	
Resultado obtenido	a) SI / NO / PARCIALMENTE b) SI / NO / PARCIALMENTE
Resultado final	Excelente
Estado	Finalizada

5.1.13 Control automático del alumbrado externo

En la Tabla 99, se presenta la prueba efectuará para el control automático del alumbrado externo del hogar.

Tabla 99. P15_Prueba para el control automático del alumbrado externo

TIPO	DESCRIPCIÓN
Identificador	P15
Nombre de la prueba	Control automático del alumbrado externo.
Descripción de la prueba	Prueba que permitirá el control automatizado del alumbrado externo del hogar.
Pre-condiciones	<ul style="list-style-type: none"> • Aplicativo móvil instalado en un smartpone. • Smartphone conectado a una red de internet vía Wi-Fi o datos móviles. • Raspberry Pi conectada a la fuente de alimentación. • Raspberry Pi conectada a una red de internet vía Wi-Fi o ethernet. • Script de Python para detección de luminosidad ejecutándose.
Resultado esperado	a) Al detectar la presencia o ausencia de luz natural o artificial, el sistema de seguridad activará o desactivará respectivamente, el alumbrado exterior de la vivienda. b) El aplicativo móvil recibirá una notificación push del evento.
USUARIO 1	
Resultado obtenido	a) SI / NO / PARCIALMENTE b) SI / NO / PARCIALMENTE
Resultado final	Malo
Estado	Finalizada
USUARIO 2	
Resultado obtenido	a) SI / NO / PARCIALMENTE b) SI / NO / PARCIALMENTE

Resultado final	Excelente
Estado	Finalizada
USUARIO 3	
Resultado obtenido	a) SI / NO / PARCIALMENTE b) SI / NO / PARCIALMENTE
Resultado final	Excelente
Estado	Finalizada

5.1.14 Detección de gases nocivos

En la Tabla 100, se presenta la prueba efectuará para la detección de gases nocivos dentro de la vivienda.

Tabla 100. P16_Prueba para la detección de gases nocivos

TIPO	DESCRIPCIÓN
Identificador	P16
Nombre de la prueba	Detección de gases nocivos.
Descripción de la prueba	Prueba que permitirá la detección de gases nocivos dentro del inmueble.
Pre-condiciones	<ul style="list-style-type: none"> • Aplicativo móvil instalado en un smartphone. • Smartphone conectado a una red de internet vía Wi-Fi o datos móviles. • Raspberry Pi conectada a la fuente de alimentación. • Raspberry Pi conectada a una red de internet vía Wi-Fi o ethernet. • Script de Python para detección de gases nocivos ejecutándose.
Resultado esperado	<ul style="list-style-type: none"> • Al detectar la contaminación del aire del inmueble con gas o humo, el sistema de seguridad activará el extractor de aire (ventilador) y una alarma sonora. • El aplicativo móvil recibirá una notificación push del incidente.
Resultado obtenido	a) El sistema de seguridad al detectar la contaminación del aire de inmueble con gas o humo, activó de manera automática tanto el extractor de aire como la alarma sonora. b) La app recibió una notificación push del incidente.
USUARIO 1	
Resultado obtenido	a) SI / NO / PARCIALMENTE b) SI / NO / PARCIALMENTE
Resultado final	Excelente
Estado	Finalizada
USUARIO 2	
Resultado obtenido	a) SI / NO / PARCIALMENTE b) SI / NO / PARCIALMENTE
Resultado final	Bueno
Estado	Finalizada
USUARIO 3	
Resultado obtenido	a) SI / NO / PARCIALMENTE b) SI / NO / PARCIALMENTE
Resultado final	Bueno
Estado	Finalizada

5.1.15 Visualización del historial de eventos

En la Tabla 101, se presenta la prueba efectuada sobre la interfaz que despliega el historial de eventos e incidentes suscitados en el hogar.

Tabla 101. P17_Prueba para la visualización del historial de eventos e incidentes del hogar

TIPO	DESCRIPCIÓN
Identificador	P17
Nombre de la prueba	Visualización del historial de eventos.
Descripción de la prueba	Prueba que permitirá la visualización y borrado de todos los eventos e incidentes registrado en el hogar.
Pre-condiciones	<ul style="list-style-type: none"> • Aplicativo móvil instalado en un smartphone. • Smartphone conectado a una red de internet vía Wi-Fi o datos móviles. • Ingreso a las funciones de la app opción Home. • Raspberry Pi conectada a la fuente de alimentación. • Raspberry Pi conectada a una red de internet vía Wi-Fi o ethernet. • Ejecución de uno o varios Scripts de Python (todos en su mayoría generan un historial de eventos o incidentes).
Resultado esperado	a) El aplicativo móvil contará con una interfaz que desplegará un historial de todos los eventos e incidentes captados por el sistema de seguridad. b) La interfaz contará con un botón que permitirá eliminar el evento o incidente en cuestión.
USUARIO 1	
Resultado obtenido	a) SI / NO / PARCIALMENTE b) SI / NO / PARCIALMENTE
Resultado final	Bueno
Estado	Finalizada
USUARIO 2	
Resultado obtenido	a) SI / NO / PARCIALMENTE b) SI / NO / PARCIALMENTE
Resultado final	Excelente
Estado	Finalizada
USUARIO 3	
Resultado obtenido	a) SI / NO / PARCIALMENTE b) SI / NO / PARCIALMENTE
Resultado final	Excelente
Estado	Finalizada

5.2 Resultados y Discusión

A continuación, en la Figura 120, se presentan tabulados los *resultados finales* de las pruebas funcionales efectuadas por *tres usuarios* al prototipo de seguridad para control domótico, en la *escala cualitativa* presentada en la sección 5.

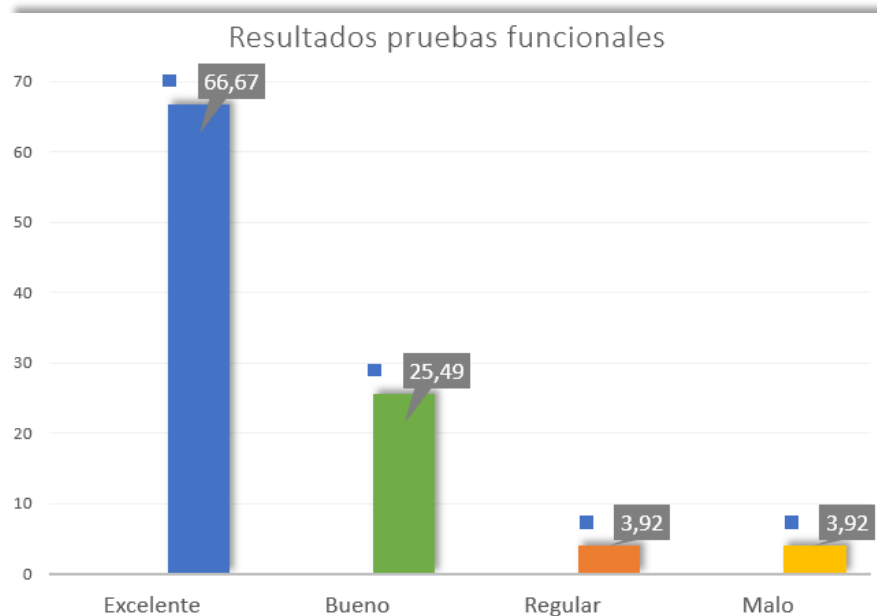


Figura 120. Resultados pruebas funcionales

Como se observa en el gráfico presentado en la Figura 120, el 66.67% corresponden a un total de 32 pruebas con resultado Excelente, un 25.49% a un total de 13 pruebas con resultado Bueno, mientras que un 3.92% de un total de 2 pruebas con resultado Regular y Malo, respectivamente.

Frente a lo expuesto y observado se puede deducir que: el 92.16% de las pruebas corresponden a un valor Excelente y Bueno; y que solo el 7.84% de las pruebas corresponden a los valores Regular y Malo.

Dichos indicadores nos permiten asegurar que el prototipo de seguridad para control domótico opera de manera adecuada tanto a nivel de hardware como software, generando una interacción en tiempo real de manera efectiva, con retardos de respuesta y ejecución relativamente bajos; entre 1s y 5s correspondientes al 66.67%, entre 6s y 15s correspondientes al 25.49% y tan solo un 7.84% que representa a valores de 16s o superiores.

Valores superiores a 16s, por lo general se los obtiene únicamente al tratar de ejecutar tareas relacionadas con la solicitud y carga de video en tiempo real, esto debido a que internamente Raspberry Pi debe esperar la solicitud generada por el aplicativo móvil, la

cual posteriormente es procesada por un servidor de video, el cual finalmente permite la creación de sockets para el establecimiento de comunicación entre la cámara y el aplicativo móvil, tarea que supone una mayor carga de procesamiento y un mayor retardo de respuesta hacia el usuario final.

Con dichas premisas, como punto final, se puede concluir y asegurar que el prototipo de seguridad para control domótico que aquí se presenta, cumple satisfactoriamente los requerimientos establecidos, basados en la generación de confort, mejoramiento de calidad de vida y el incremento de seguridad física hacia los miembros del hogar

CAPÍTULO VI

6. CONCLUSIONES Y RECOMENDACIONES

A continuación, se presentan las conclusiones y recomendaciones obtenidas a lo largo del desarrollo del presente proyecto integrador.

6.1 Conclusiones

- El constante desarrollo y evolución de las TIC permitió emplear tecnologías vanguardistas para el diseño y construcción del prototipo de seguridad para control domótico, generando como resultado un sistema de altas prestaciones, innovador y sobre todo económico frente a los sistemas ofertados en el mercado ecuatoriano.
- Las plataformas de software y hardware empleadas Firebase y Raspberry Pi respectivamente; permitieron generar un sistema de seguridad para control domótico basado en IoT.
- El sistema de seguridad para control domótico permitió un mejoramiento de la calidad de vida y sobre todo el aporte al elevamiento de seguridad física tanto de los miembros del hogar como de sus bienes. Esto gracias a características en tiempo real como: automatización, control, monitoreo y centralización de la información en un solo dispositivo.
- El sistema de seguridad para control domótico fue implementado en un prototipo que permitió simular las características de un hogar; con lo cual se pudo corroborar las bondades, características y el alto impacto en el elevamiento de seguridad física que consigo aportaría en viviendas reales del Distrito Metropolitano de Quito.
- El diseño y construcción del prototipo de seguridad para control domótico supone un aporte al sector de la seguridad enfocada en la vivienda de los diferentes tipos de niveles socioeconómicos de la ciudad de Quito.
- La implementación del sistema de seguridad para control domótico permitió reforzar mecanismos de seguridad tradicionales, mediante la incorporación de una central de procesamiento, sensores y actuadores a las diferentes áreas de la vivienda, generando así una inmersión del hogar a la nueva era tecnológica.

- El uso de Scrum como marco de trabajo permitió la elaboración tanto de definiciones como objetivos de los diversos Sprints a llevarse a cabo durante todo el ciclo de vida del proyecto, posibilitando de esta forma el cumplimiento de las tareas en base a fechas y tiempos establecidos; factores que trajeron consigo la posibilidad de una entrega funcional y continúa en cada iteración.
- La elección de las tecnologías y herramientas estuvieron basadas en la definición inicial de requerimientos, en las cuales se evaluaron compatibilidad, modo de comunicación y coexistencia tanto a nivel de hardware como de software.
- Android Studio como plataforma para el desarrollo provee un ambiente sencillo, funcional y completo para la creación de aplicaciones móviles; características que permitieron obtener como resultado final una aplicación robusta, intuitiva y agradable a la vista.
- Firebase como agente de escucha provee los mecanismos para el almacenamiento de información, autenticación de usuarios y mensajería. Funciones que son llevadas a cabo en tiempo real y reflejadas en todas las plataformas que cuenten con la incorporación de los SDK.
- Raspberry Pi como unidad central de procesamiento, permite el control y comunicación de diversos sensores y actuadores. Esto gracias a su alto poder computacional y a su cabezal de 40 pines que le permiten interactuar con elementos del mundo exterior.
- El desarrollo de un sistema de seguridad propio, basado en requerimientos iniciales supone un sistema más completo, económico y en mucho de los casos con un mejor rendimiento que los que ofrece el mercado.
- La domótica en conjunto con IoT constituyen tecnologías que se complementan y trabajan en perfecta armonía para la generación de soluciones en el ámbito de la vivienda.

6.2 Recomendaciones

- Se ha demostrado que el marco de trabajo Scrum para el desarrollo de software, permite un mejor control de las tareas a efectuarse y sobre todo garantiza la entrega continua y funcional en cortos tiempos durante todo el ciclo de vida del proyecto, por lo cual su uso es recomendable.
- Para el desarrollo de software es recomendable el uso de repositorios en conjunto con herramientas para control de versiones, esto debido a que los mismos proveen una gran cantidad de beneficios a lo largo del desarrollo del aplicativo, tales como: trabajo colaborativo, histórico de acciones, copia del proyecto en la nube, etc.
- Se recomienda *no* iniciar con el desarrollo de los componentes de hardware y software, si no se cuenta con los requerimientos iniciales, ya que los mismos constituyen la base sobre la cual se trabajará durante todo el ciclo de vida del proyecto.
- La unidad central de procesamiento es recomendable mantenerla conectada cerca del punto de acceso que provee internet o mediante ethernet, esto con la finalidad de garantizar el correcto funcionamiento de los sensores y actuadores que forman parte del sistema de seguridad.
- Es recomendable conectar la unidad central de procesamiento a un UPS, con el fin de garantizar la operación continua del sistema de seguridad frente a cortes de energía eléctrica.
- Para acceder a la transmisión de video fuera de la red del hogar, es recomendable generar una IP pública, con el fin de poder acceder a la misma desde cualquier parte que el usuario se encuentre.
- Se recomienda el uso de la plataforma Firebase como solución a implementación de sistemas domóticos basados en IoT, esto debido a su escalabilidad, gran número de funciones, soporte para diversas plataformas y modo de operación en tiempo real.

- Al contar con una gran cantidad de diagramas electrónicos, es recomendable antes de su incorporación al sistema, la construcción y testeo de los circuitos en un protoboard, esto con la finalidad de observar el correcto o mal funcionamiento de los mismos, permitiendo así generar correcciones de ser necesario.
- Es recomendable realizar actualizaciones periódicas a la plataforma de hardware Raspberry Pi, ya que, con ello, cada librería y programa alojado en la misma, recibe las últimas características y funciones, permitiendo así mantener todo el sistema actualizado.
- Periódicamente es necesario revisar y actualizar las dependencias del gradle del proyecto móvil, esto debido a que las librerías que forman parte de la app generalmente liberan nuevas funciones, características o traen consigo cambios en la sintaxis y modo de uso de las mismas.

REFERENCIAS BIBLIOGRÁFICAS

- [1] A. Recuero, «Estado actual y perspectiva de la domótica,» *Informes de la construcción*, vol. 50, nº 459, p. 1, 1998.
- [2] Centro de Investigación CLAVE, «Bienes Raíces Clave,» 08 Abril 2018. [En línea]. Available: <https://www.clave.com.ec/2018/06/08/estadisticas-de-la-feria-mi-casa-clave/>. [Último acceso: 25 Noviembre 2019].
- [3] Bellavista S.C, «Bellavista S.C,» Septiembre 2019. [En línea]. Available: <https://www.bellavistasc.com/proyectos-en-ejecucion/jardines-del-dean/>. [Último acceso: 25 Noviembre 2019].
- [4] C. L. Jimeno, «La Domótica Como Solución del Futuro,» Consejería de Economía e Innovación Tecnológica, Madrid, 2007.
- [5] A. Flórez, «Hacia una definición de la domótica,» *ResearchGate*, p. 13, 2004.
- [6] F. J. J. Montero, «Sites Google,» 2008. [En línea]. Available: <https://sites.google.com/site/ejdlcdomotica/Home/tema-1-introduccion-a-la-domotica#TOC-DOM-TICA>. [Último acceso: 25 Noviembre 2019].
- [7] J. M. Huidobro, «La domótica entra en nuestras casas,» ACTA, Madrid, 2004.
- [8] TECNOSeguro, «TECNOSeguro,» Magazín Digital - Online Media, [En línea]. Available: <https://www.tecnoseguro.com/faqs/alarma/que-es-un-detector-de-movimiento-pasivo-o-pir>. [Último acceso: 27 Noviembre 2019].
- [9] Wikipedia , «Wikipedia,» Wikipedia , 20 Noviembre 2019. [En línea]. Available: https://es.wikipedia.org/wiki/Motor_de_corriente_continua. [Último acceso: 27 Noviembre 2019].
- [10] F. Mecafenix, «Ingeniería Mecafenix,» Mecafenix, 18 Abril 2017. [En línea]. Available: <https://www.ingmecafenix.com/electricidad-industrial/servomotor/>. [Último acceso: 27 Noviembre 2019].
- [11] Wikipedia, «Wikipedia,» 20 Noviembre 2019. [En línea]. Available: <https://es.wikipedia.org/wiki/Rel%C3%A9>. [Último acceso: 27 Noviembre 2019].
- [12] A. Barrio, «INTERNET DE LAS COSAS,» de *INTERNET DE LAS COSAS*, Madrid, Reus, 2018, p. 18.
- [13] A. Barrio, «INTERNET DE LAS COSAS,» de *INTERNET DE LAS COSAS*, Madrid, Reus, 2018, pp. 18,19.

- [14] M. López, «Internet de las cosas La transformación digital de la sociedad,» de *Internet de las cosas La transformación digital de la sociedad*, Madrid, Ra-Ma, 2019, pp. 44-50.
- [15] RS Components, «RS Components,» Amidata S.A.U, [En línea]. Available: <https://es.rs-online.com/web/generalDisplay.html?id=i/iot-internet-of-things>. [Último acceso: 28 Noviembre 2019].
- [16] J. Salazar y S. Silvestre, «Internet de las cosas,» TechPedia, Chequia.
- [17] «Arduino Cl,» [En línea]. Available: <https://arduino.cl/que-es-arduino/>. [Último acceso: 28 Noviembre 2019].
- [18] «RaspberryPi,» [En línea]. Available: <https://www.raspberrypi.org/help/what-%20is-a-raspberry-pi/>. [Último acceso: 29 Noviembre 2019].
- [19] J. M. López, «hipertextual,» 17 Enero 2020. [En línea]. Available: <https://hipertextual.com/2020/01/estacion-meteorologica-raspberry-pi>. [Último acceso: 4 Abril 2020].
- [20] J. M. López, «hipertextual,» 31 Julio 2018. [En línea]. Available: <https://hipertextual.com/2018/07/consola-videojuegos-raspberry-instrucciones>. [Último acceso: 4 Abril 2020].
- [21] J. M. López, «hipertextual,» 29 Diciembre 2018. [En línea]. Available: <https://hipertextual.com/2018/12/raspberry-centro-multimedia>. [Último acceso: 4 Abril 2020].
- [22] Rasperry pi foundation, «Raspberry Pi,» [En línea]. Available: <https://projects.raspberrypi.org/es-ES/projects/chatbot>. [Último acceso: 4 Abril 2020].
- [23] «Raspberrypi,» [En línea]. Available: <https://www.raspberrypi.org/products/>. [Último acceso: 29 Noviembre 2019].
- [24] «RaspberyPi,» [En línea]. Available: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/>. [Último acceso: 29 Noviembre 2019].
- [25] «RaspberryPi,» [En línea]. Available: <https://www.raspberrypi.org/documentation/usage/gpio/>. [Último acceso: 1 Diciembre 2019].
- [26] Google Developers, «Firebase,» Google, [En línea]. Available: <https://firebase.google.com/>. [Último acceso: 2 Diciembre 2019].
- [27] Google Developers, «Firebase,» Google, [En línea]. Available: <https://firebase.google.com/docs/firestore/>. [Último acceso: 2 Diciembre 2019].

- [28] Google Developers, «Firebase,» Google, [En línea]. Available: <https://firebase.google.com/docs/ml-kit/>. [Último acceso: 2 Diciembre 2019].
- [29] Google Developers, «Firebase,» Google, [En línea]. Available: <https://firebase.google.com/docs/functions/>. [Último acceso: 2 Diciembre 2019].
- [30] Google Developers, «Firebase,» Google, [En línea]. Available: <https://firebase.google.com/docs/auth/>. [Último acceso: 2 Diciembre 2019].
- [31] Google Developers, «Firebase,» Google, [En línea]. Available: <https://firebase.google.com/docs/storage/>. [Último acceso: 2 Diciembre 2019].
- [32] Google Developers, «Firebase,» Google, [En línea]. Available: <https://firebase.google.com/docs/cloud-messaging/>. [Último acceso: 2 Diciembre 2019].
- [33] K. Polanco y J. L. Beuperthuy, «“ANDROID” EL SISTEMA OPERATIVO DE GOOGLE PARA DISPOSITIVOS MÓVILES,» Revista Negotium, Venezuela , 2011.
- [34] «Android developers,» Google, [En línea]. Available: <https://source.android.com/setup/start/build-numbers>. [Último acceso: 25 Diciembre 2019].
- [35] Ó. Ávila, «Android,» UAM-I. Departamento de Ingeniería Eléctrica, Iztapalapa, 2011.
- [36] A. Menzinsky, L. Gertrudis y P. Juan, «Scrum Manager,» 2016.
- [37] A. Itzcoalt, «Desarrollo Ágil con SCRUM,» Bogotá, 2017.
- [38] S. Sara, «Saraclip,» Wordpress, 31 Octubre 2017. [En línea]. Available: <https://www.saraclip.com/eventos-en-scrum/>. [Último acceso: 23 Diciembre 2019].
- [39] S. Sara, «Saraclip,» Wordpress, 3 Noviembre 2017. [En línea]. Available: <https://www.saraclip.com/eventos-en-scrum-ii/>. [Último acceso: 23 Diciembre 2019].
- [40] S. Sara, «Saraclip,» Wordpress, 24 Octubre 2017. [En línea]. Available: <https://www.saraclip.com/roles-en-scrum/>. [Último acceso: 23 Diciembre 2019].
- [41] S. Casanova, «Samuel Casanova Efectividad en equipos de desarrollo,» Wordpress, 2016. [En línea]. Available: <https://samuelcasanova.com/2016/01/estimacion-agil-con-la-tecnica-planning-poker/>. [Último acceso: 27 Diciembre 2019].
- [42] «RaspberryPi,» [En línea]. Available: <https://www.raspberrypi.org/downloads/>. [Último acceso: 2 Enero 2020].
- [43] «balenaEtcher,» Balena, [En línea]. Available: <https://www.balena.io/etcher>. [Último acceso: 2 Enero 2020].
- [44] «Git,» [En línea]. Available: <https://git-scm.com/>. [Último acceso: 2 Enero 2020].

- [45] «Github,» Github, Inc., [En línea]. Available: <https://github.com/features>. [Último acceso: 2 Enero 2020].
- [46] «Github,» Github, Inc., [En línea]. Available: <https://desktop.github.com/>. [Último acceso: 2 Enero 2020].
- [47] «Android developers,» Google, [En línea]. Available: <https://developer.android.com/kotlin/overview>. [Último acceso: 2 Enero 2020].
- [48] «python,» [En línea]. Available: <https://www.python.org/about/>. [Último acceso: 2 Enero 2020].
- [49] «Android developers,» Google, [En línea]. Available: <https://developer.android.com/studio>. [Último acceso: 2 Enero 2020].
- [50] «PyCharm,» JetBrains s.r.o, [En línea]. Available: <https://www.jetbrains.com/es-es/pycharm/>. [Último acceso: 2 Enero 2020].
- [51] «Proteus,» Labcenter, [En línea]. Available: <https://www.labcenter.com/>. [Último acceso: 2 Enero 2020].
- [52] B. Nuttall, «gpiozero,» [En línea]. Available: <https://gpiozero.readthedocs.io/en/stable/>. [Último acceso: 3 Enero 2020].
- [53] D. Jones, «picamera,» [En línea]. Available: <https://picamera.readthedocs.io/en/release-1.13/>. [Último acceso: 3 Enero 2020].
- [54] R. McGrath, «twython,» [En línea]. Available: <https://twython.readthedocs.io/en/latest/index.html>. [Último acceso: 3 Enero 2020].
- [55] «Python docs,» [En línea]. Available: <https://docs.python.org/3/library/smtplib.html>. [Último acceso: 3 Enero 2020].
- [56] Python, «Python Package Index,» [En línea]. Available: <https://pypi.org/project/firebase-admin/>. [Último acceso: 3 Enero 2020].
- [57] «The pallets projects,» [En línea]. Available: <https://www.palletsprojects.com/p/flask/>. [Último acceso: 3 Enero 2020].
- [58] «Glide,» [En línea]. Available: <https://bumptech.github.io/glide/>. [Último acceso: 3 Enero 2020].
- [59] Airbnb, Inc., «Lottie,» Airbnb, Inc., [En línea]. Available: <https://airbnb.design/lottie/>. [Último acceso: 3 Enero 2020].
- [60] G. Kumar, «Github,» [En línea]. Available: <https://github.com/gauravk95/bubble-navigation>. [Último acceso: 3 Enero 2020].

- [61] Firebase, «Github,» [En línea]. Available: <https://github.com/firebase/FirebaseUI-Android/tree/master/database>. [Último acceso: 3 Enero 2020].
- [62] A. Follestad, «Github,» [En línea]. Available: <https://github.com/afollestad/material-dialogs>. [Último acceso: 3 Enero 2020].
- [63] P. Jahoda, «Github,» [En línea]. Available: <https://github.com/PhilJay/MPAndroidChart>. [Último acceso: 3 Enero 2020].
- [64] Barn Inc, «LottieFiles,» [En línea]. Available: <https://lottiefiles.com/>. [Último acceso: 3 Enero 2020].
- [65] Freepik Company, «Flaticon,» [En línea]. Available: <https://www.flaticon.com/>. [Último acceso: 3 Enero 2020].
- [66] Freepik Company, «Freepik,» [En línea]. Available: <https://www.freepik.com/>. [Último acceso: 3 Enero 2020].
- [67] ucid Software Inc., «Lucidchart,» [En línea]. Available: <https://www.lucidchart.com/pages/es>. [Último acceso: 3 Enero 2020].
- [68] «Andriod developers,» Google, [En línea]. Available: <https://developer.android.com/about/versions/android-4.4>. [Último acceso: 3 Enero 2020].
- [69] Google Developers, «Firebase,» Google, [En línea]. Available: <https://firebase.google.com/>. [Último acceso: 4 Enero 2020].
- [70] «Alldatasheet,» Alldatasheet.com, [En línea]. Available: <https://pdf1.alldatasheet.es/datasheet-pdf/view/58557/DALLAS/DS18B20.html>. [Último acceso: 7 Enero 2020].
- [71] «Olimex,» [En línea]. Available: <https://www.olimex.com/Products/Components/Sensors/Gas/SNS-MQ135/resources/SNS-MQ135.pdf>. [Último acceso: 7 Enero 2020].
- [72] «Punto flotante,» [En línea]. Available: <https://puntoflotante.net/MANUAL-DEL-USUARIO-SENSOR-DE-MOVIMIENTO-PIR-HC-SR501.pdf>. [Último acceso: 7 Enero 2020].
- [73] «Geekfactory,» [En línea]. Available: <https://www.geekfactory.mx/tienda/otros-accesorios/mini-buzzer-zumbador-acustico-5-v/>. [Último acceso: 8 Enero 2020].
- [74] «Naylamp Mechatronics,» [En línea]. Available: <https://naylampmechatronics.com/sensores-proximidad/215-sensor-magnetico-de-puerta-mc-38.html>. [Último acceso: 8 Enero 2020].

- [75] «Naylamp Mechatronics,» [En línea]. Available: <https://naylampmechatronics.com/drivers/239-driver-mosfet-irf520.html>. [Último acceso: 8 Enero 2020].
- [76] «Naylamp Mechatronics,» [En línea]. Available: <https://naylampmechatronics.com/drivers/239-driver-mosfet-irf520.html>. [Último acceso: 8 Enero 2020].
- [77] «Electronilab,» [En línea]. Available: <https://electronilab.co/tienda/micro-servo-9g-towerpro/>. [Último acceso: 8 Enero 2020].
- [78] «Naylamp Mechatronics,» [En línea]. Available: <https://naylampmechatronics.com/sensores-luz-y-sonido/241-sensor-ldr-5528.html>. [Último acceso: 8 Enero 2020].
- [79] Firebase, «Github,» [En línea]. Available: <https://github.com/firebase/FirebaseUI-Android/tree/master/database>. [Último acceso: 3 Enero 2020].
- [80] «Android developers,» Google, [En línea]. Available: <https://developer.android.com/about/versions/android-4.4>. [Último acceso: 3 Enero 2020].

ANEXOS

ANEXO 1. Código fuente del aplicativo móvil

ANEXO 2. APK del aplicativo móvil

ANEXO 3. Código fuente de los controladores de hardware del sistema de seguridad

ANEXO 4. Planificación de Sprints efectuados

ANEXO 5. Estimación de costos del sistema de seguridad prototipo

ANEXO 6. Diagramas y PCB de los circuitos electrónicos

ANEXO 7. Diagrama de la estructura de la base de datos

ANEXO 8. Mockups del aplicativo móvil

ANEXO 9. Pantallas del aplicativo móvil

ANEXO 10. Manual de usuario