

UNIVERSIDAD PERUANA UNIÓN

FACULTAD DE INGENIERÍA Y ARQUITECTURA

Escuela Profesional de Ingeniería de Sistemas



Una Institución Adventista

Desarrollo de un sistema domótico controlado mediante dispositivos móviles para la vigilancia remota de viviendas familiares

Por:

Angie Mireya Mamani Chambi

Asesor:

Ing. Angel Rosendo Condori Coaquira

Juliaca, mayo de 2019

DECLARACIÓN JURADA
DE AUTORIA DEL INFORME DE TESIS

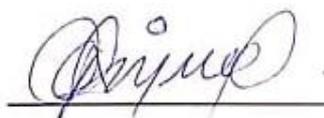
Ing. Angel Rosendo Condori Coaquira, de la Facultad de Ingeniería y Arquitectura,
Escuela Profesional de Ingeniería de Sistemas, de la Universidad Peruana Unión.

DECLARO:

Que el presente informe de investigación titulado: **“DESARROLLO DE UN SISTEMA DOMÓTICO CONTROLADO MEDIANTE DISPOSITIVOS MÓVILES PARA LA VIGILANCIA REMOTA DE VIVIENDAS FAMILIARES”** constituye la memoria que presenta la **Bachiller Angie Mireya Mamani Chambi** para aspirar al título Profesional de Ingeniero de Sistemas ha sido realizada en la Universidad Peruana Unión bajo mi dirección.

Las opiniones y declaraciones en este informe son de entera responsabilidad del autor, sin comprometer a la institución.

Y estando de acuerdo, firmo la presente declaración en Juliaca a los catorce días del mes de mayo del año dos mil diecinueve.



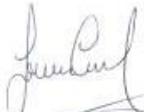
Ing. Angel Rosendo
Condori Coaquira

Desarrollo de un sistema domótico controlado mediante dispositivos
móviles para la vigilancia remota de viviendas familiares

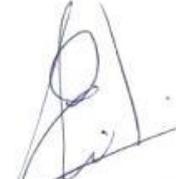
TESIS

Presentada para optar el título profesional de Ingeniero de Sistemas

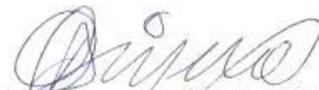
JURADO CALIFICADOR


Mg. Lennin Henry Centurión Julca
Presidente


Dr. Jorge Alejandro Sánchez Garcés
Secretario


Ing. Eder Gutiérrez Quispe
Vocal


Ing. Benazir Francis Herrera Yucra
Vocal


Ing. Angel Rosendo Condori Coaquira
Asesor

Juliaca, 13 de Mayo de 2019

DEDICATORIA

La presente tesis es dedicada a Dios, por brindarme la sabiduría necesaria y las fuerzas para no desistir en el proceso, a mi familia quienes han sido un pilar muy importante para realización de esta investigación, a mis padres Fermin y Josefina por el apoyo incondicional y económico, a mis hermanos por los ánimos que me dieron durante el proceso de realización de estas mismas.

AGRADECIMIENTOS

A Dios por la vida y las oportunidades que me ofreció como es el estudiar.

A mi padre Fermin por la confianza y el apoyo incondicional tanto emocional y económico.

A mi madre Josefina por su amor y paciencia.

A mis hermanos Guido, Josue, Melanie, Henry por darme los ánimos para continuar.

Al Dr. Jorge Sánchez por impulsarnos a iniciar este proceso de tesis.

A mi asesor el Ing. Angel Coaquira por apoyarme y guiarme en algunas dificultades que tuve al momento de desarrollar esta tesis.

A la promoción 2018-II de Ingeniería de Sistemas, una promoción de inspiración.

A mis amigos Cristian, Betsaida y Nilda por su compañía en este proceso y sus energías positivas.

ÍNDICE GENERAL

DEDICATORIA.....	iv
AGRADECIMIENTOS.....	v
ÍNDICE DE TABLAS.....	ix
ÍNDICE DE FIGURAS.....	xi
ÍNDICE DE ANEXOS.....	xiv
SÍMBOLOS USADOS.....	xv
RESUMEN.....	xvi
ABSTRACT.....	xvii
CAPÍTULO I. EL PROBLEMA.....	18
1.1.Descripción de la situación problemática.....	18
1.2.Objetivos de la investigación.....	19
1.2.1.Objetivo general.....	19
1.2.2.Objetivos específicos.....	19
1.3.Justificación.....	19
1.4.Presunción filosófica.....	20
CAPÍTULO II. MARCO TEÓRICO.....	21
2.1.Revisión de la literatura.....	21
2.2.Bases teóricas.....	23
2.2.1.Vigilancia remota.....	23
2.2.2.Seguridad ciudadana.....	24
2.2.3.Domótica.....	25
2.2.4.Microcontrolador.....	27
2.2.5.Sensores.....	32
2.2.6.Actuadores.....	34
2.2.7.Relé.....	35
2.2.8.No-IP.....	37

2.2.9.IP dinámica	38
2.2.10.IP estática.....	38
2.2.11.IP pública.....	38
2.2.12.Redes de área local (LAN)	38
2.2.13.Redes de área amplia (WAN).....	39
2.2.14.Protocolo HTTP.....	40
2.2.15.Android.....	40
2.2.16.Aplicación móvil	41
2.2.17.Desarrollo móvil nativo	41
2.2.18.Lenguaje de programación	43
2.2.19.Frameworks	43
2.2.20.Base de datos	46
2.2.21.Metodología ágil XP.....	47
CAPÍTULO III. MATERIALES Y MÉTODOS	50
3.1.Descripción del lugar de ejecución.....	50
3.2.Metodología de la investigación.....	50
3.2.1.Tipo de investigación	50
3.3.Materiales e insumos	50
3.3.1.Hardware	51
3.3.2.Software.....	51
3.3.3.Arquitectura de solución.....	52
3.3.4.Metodología XP.....	53
CAPÍTULO IV. DESARROLLO DEL SISTEMA DOMÓTICO	55
4.1.Fase de planificación	55
4.1.1.Historias de usuario	55
4.2.Fase de diseño.....	63
4.2.1.Diseño simple - arquitectura tecnológica	63
4.2.2.Prototipo del esquema físico de las conexiones en Arduino	64
4.2.3.Prototipo del servicio web Arduino.....	65
4.2.4.Prototipo de la aplicación móvil.....	65

4.3.Fase de codificación	67
4.3.1.Desarrollo servidor web - Arduino	67
4.3.2.Desarrollo móvil – NativeScritp-Vue.....	68
4.4.Fase de pruebas.....	70
4.4.1.Pruebas de configuración del hardware.....	70
4.4.2.Pruebas de aceptación del aplicativo móvil.....	71
CAPÍTULO V. RESULTADOS Y DISCUSIÓN	76
5.1.Resultado del objetivo específico 1	76
5.1.1.Discusión	76
5.2.Resultado del objetivo específico 2	76
5.2.1.Discusión.....	77
5.3.Resultado del objetivo específico 3	78
5.3.1.Discusión.....	78
CAPÍTULO VI. CONCLUSIONES Y RECOMENDACIONES	79
6.1.Conclusiones.....	79
6.2.Recomendaciones	80
REFERENCIAS	81
ANEXOS	87

ÍNDICE DE TABLAS

Tabla 1. Tabla de comparación de microcontroladores.....	28
Tabla 2. Tabla de comparación entre aplicación híbrida y aplicación nativa.....	41
Tabla 3. Tabla de comparación de lenguajes de programación.....	43
Tabla 4. Tabla de comparación de frameworks.....	44
Tabla 5. Tabla de comparación entre base de datos relacional y no relacional.....	46
Tabla 6. Comparación de las metodologías ágiles SCRUM y XP.	49
Tabla 7. Historia de usuario número 1	56
Tabla 8. Historia de usuario número 2	56
Tabla 9. Historia de usuario número 3	57
Tabla 10. Historia de usuario número 4	57
Tabla 11. Historia de usuario número 5	58
Tabla 12. Historia de usuario número 6	58
Tabla 13. Historia de usuario número 7	59
Tabla 14. Requerimientos funcionales	59
Tabla 15. Requerimientos no funcionales	60
Tabla 16. Tarea 1, HDU-01	60
Tabla 17. Tarea 2, HDU-02 y HDU-03	61
Tabla 18. Tarea 3, HDU-03, HDU-04 y HDU-07	61
Tabla 19. Tarea 4, HDU-05	62

Tabla 20. Tarea 5, HDU-06	62
Tabla 21. Tarea 6, HDU-01 hasta HDU-07	63
Tabla 22. Caso de prueba - Correcto funcionamiento de los dispositivos electrónicos integrados con la placa Arduino y Shield Ethernet	70
Tabla 23. Caso de prueba - Validar usuarios	72
Tabla 24. Caso de prueba - Visualizar video en tiempo real	74
Tabla 25. Caso de prueba - Encender y apagar los dispositivos integrados al sistema domótico	75

ÍNDICE DE FIGURAS

Figura 1. Viviendas a nivel nacional urbano, afectadas por robo o intento de robo, según ámbito de estudio, 2010 – 2017.....	18
Figura 2. Índice de paz global.	24
Figura 3. Arquitectura centralizada.	25
Figura 4. Arquitectura distribuida.	26
Figura 5. Arquitectura mixta.	26
Figura 6. Componentes en un hogar inteligente.	27
Figura 7. Arquitectura del microcontrolador, arquitectura Harvard.....	28
Figura 8. Arquitectura RISC, es la arquitectura de la placa Arduino.....	30
Figura 9. IDE Arduino.....	30
Figura 10. Arduino Uno R3.....	31
Figura 11. Arduino Ethernet Shield.....	32
Figura 12. Sensor PIR.....	33
Figura 13. Sensor buzzer pasivo Arduino.	34
Figura 14. Cámara IP.....	34
Figura 15. Focos LED.	35
Figura 16. Vista interna del relé y cómo se puede construir.	36
Figura 17. Relé de dos canales.	37
Figura 18. No-IP.....	37
Figura 19. Red LAN.....	39

Figura 20. Red WAN.....	39
Figura 21. Componentes principales de la plataforma Android.....	40
Figura 22. Aplicación nativa e híbrida.	42
Figura 23. NativeScript-Vue.	46
Figura 24. Funcionalidad de Firebase.....	47
Figura 26. Metodología XP.	48
Figura 27. Arquitectura de solución de la investigación.	52
Figura 28. Metodología XP reducida.	53
Figura 29. Arquitectura Tecnológica del Sistema Domótico.	64
Figura 30. Diseño de esquema físico de conexiones en Arduino.....	64
Figura 31. Vista del servicio web Arduino.....	65
Figura 32. Vista de Acceso a la aplicación.....	65
Figura 33. Vista para registro de usuario.....	66
Figura 34. Vista principal de la aplicación.	66
Figura 35. Integración de Arduino, Shield Ethernet, relé, sensor de alarma y bombilla. ...	71
Figura 36. Correcto funcionamiento de la integración.....	71
Figura 37. Habilitando los servicios de autenticación por correo electrónico/contraseña y google.	72
Figura 38. Métodos de acceso al aplicativo del sistema domótico DomoPit.	73
Figura 39. Registro de nuevo usuario, almacenamiento en tiempo real Firebase.	73
Figura 40. Funcionalidad de la vista del video en tiempo real.	74

Figura 41. Funcionalidad del control para la iluminación y alarma.....	75
Figura 42. Esquema general de la integración de las tecnologías de automatización de viviendas.....	77

ÍNDICE DE ANEXOS

Anexo A. Especificaciones técnicas de hardware usado	87
Anexo B. Código de Arduino	92
Anexo C. Código NativeScript-Vue	95
Anexo D. Formatos	103
Anexo E. Código en repositorio Github	104

SÍMBOLOS USADOS

DHCP: Abreviatura en inglés de Protocolo de Configuración Dinámica de Host

UDP: Abreviatura en inglés de Protocolo de Datagramas de Usuario

TCP/IP: Abreviatura en inglés de Protocolo de Control de Transmisión/Protocolo de Internet

MAC: Abreviatura en inglés de Control de Acceso de Medios

WAN: Abreviatura en inglés Red de Área Amplia

LAN: Abreviatura en inglés de Red de Área Local

iCSee: Acrónimo de Aplicación de Monitoreo de Seguridad.

XP: Abreviatura en inglés de Programación Extrema

HTML: Abreviatura en inglés de Lenguaje de Marcas de Hipertexto

URL: Abreviatura en inglés Localizador Uniforme de Recursos

HTTP: Abreviatura en inglés Protocolo de Transferencia de Hipertexto

JS: Abreviatura de JavaScript

NS: Abreviatura de NativeScript

HDU: Historia de usuario

Plugin: Complemento relacional para agregar una función nueva

RESUMEN

Las incidencias de robo en viviendas han tenido un crecimiento notable en los últimos años, hasta el año 2018, el 12,8% de los hogares fueron afectados por algún hecho delictivo a nivel nacional urbano tal como menciona el Instituto Nacional de Estadística e Informática (INEI), frente a este problema el objetivo de esta investigación es desarrollar un sistema domótico controlado mediante dispositivos móviles para la vigilancia remota de viviendas familiares. Para el desarrollo de esta investigación se usó la metodología eXtreme Programming (XP) que cuenta con 04 fases que son: 1) Planificación en la cual se hizo el levantamiento de los requerimientos, 2) Diseño en el cual se realizó los prototipos, 3) Codificación donde se desarrolló cada requerimiento y 4) Pruebas donde se realizó las pruebas de aceptación. Se usó las herramientas Open hardware y software como Arduino, sensor PIR, sensor Buzzer pasivo, Shield Ethernet, para el desarrollo de la aplicación se utilizó NativeScript-Vue y Firebase como base de datos. Como resultados se obtuvieron 1) La integración de dispositivos al sistema domótico usando una placa Arduino en el cual se levantó un servidor HTTP usando el Ethernet Shield que permitió configurar un Router para la salida de una IP pública, 2) Visualización del video en tiempo real mediante la aplicación de la cámara y 3) Control de luces y alarmas desde el dispositivo móvil. Contribuyendo así a la sociedad tecnologías de información para la vigilancia remota en tiempo real de las viviendas familiares.

Palabras clave: Domótica, Arduino Uno R3, Ethernet Shield, Vue, NativeScript, Firebase.

ABSTRACT

Many incidents of burglary have had remarkable growth in recent years, until 2018, 12.8% of households were affected by a criminal act to urban, national level as mentioned the National Institute of Statistics and Informatics (INEI), for this problem the objective of this research is to develop the home automation system controlled by mobile devices for remote monitoring of family homes. For the development of this research was used the Extreme Programming (XP) methodology that counts with 04 phases are used: 1) Planning, in which the lifting of requirements, 2) Designing, in which the prototypes was performed, 3) Encoding, where each requirement was made and 4) Testing, where the acceptance test was performed. The tools that were used, the Open hardware and software as well as Arduino, PIR sensor, Buzzer passive sensor, Ethernet Shield, for the application development was used NativeScript-Vue and Firebase as a database. As results were obtained 1) The integration of devices to the home automation system using an Arduino in which a using Shield Ethernet HTTP server which allowed configuring a router to output a public IP, 2) Display real time video rose by applying the camera and 3) Control lights and alarms from the mobile device. So contributing information technologies to society for remote real time monitoring of family homes.

Keywords: Home automation, Uno R3 Arduino, Shield Ethernet, Vue, NativeScript, Firebase.

CAPÍTULO I

EL PROBLEMA

1.1. Descripción de la situación problemática

La problemática de esta investigación es la vigilancia remota para la seguridad de las viviendas familiares por las recurrentes incidencias de robos, la cual es una problemática social y esta sustentado en las siguientes estadísticas. Según el INEI (2017) “a nivel nacional urbano, en el año 2017 el 10,5% de las viviendas fueron afectadas por robo o intento de robo; en los años 2010 y 2011, este porcentaje superó el 20%; mientras que en los años 2012, 2013, 2014 y 2015 fue de 17,0%, 15,9%, 12,3% y 11,4% respectivamente. Según ámbito de estudio, los centros poblados urbanos entre 2 mil y menos de 20 mil habitantes, presentaron las mayores cifras porcentuales (11,2%) de viviendas afectadas por robo o intento de robo en el año 2017.”

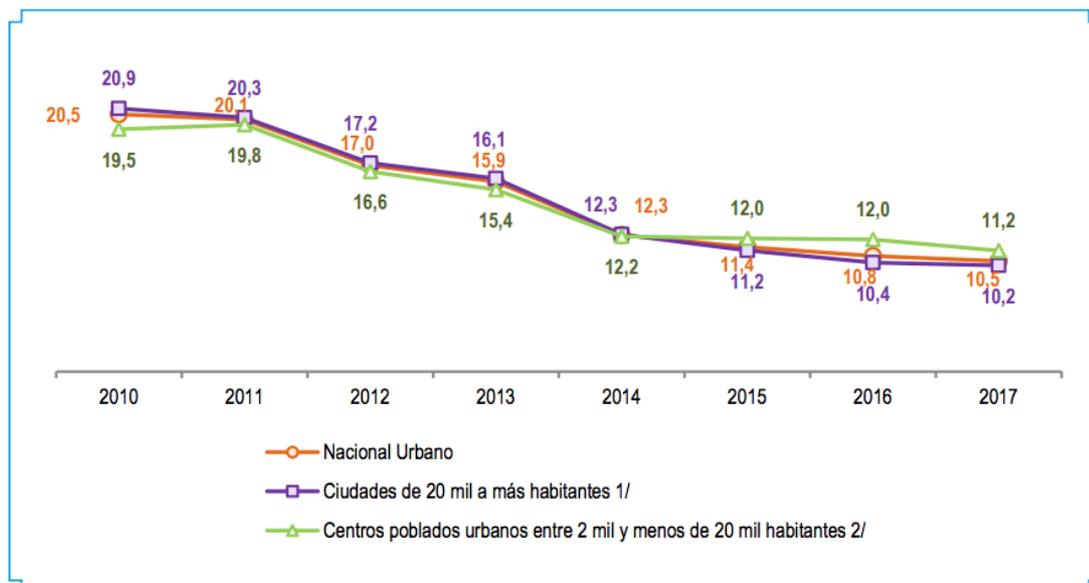


Figura 1. Viviendas a nivel nacional urbano, afectadas por robo o intento de robo, según ámbito de estudio, 2010 – 2017.

Fuente: INEI, 2017.

Y en el año 2018 según el INEI & MININTER (2018) el 12,8% de los hogares fueron afectados por algún hecho delictivo a nivel nacional urbano. Los departamentos que

presentaron los mayores porcentajes fueron Madre de Dios (22,8%) y Amazonas (21,3%). De igual modo, el 4,3% de los hogares fueron afectados por robo en la vivienda. Al ver el aumento del índice de robos en las viviendas, los dueños de las viviendas familiares, necesitan contar con un sistema de vigilancia remota, que les permita monitorear sus viviendas desde el lugar de donde se encuentren.

En Juliaca, una ciudad comercial, existe la informalidad, el desorden y un gran incremento de actos delictivos, entre ellos el robo en viviendas, en feriados largos eventualmente las viviendas quedan sin la presencia de personas, es en este momento que los delincuentes aprovechan para robar. Haciendo uso de tecnologías como la Fibra Óptica y la Red 4G LTE, se puede apoyar a la reducción de este problema en la sociedad.

1.2. Objetivos de la investigación

1.2.1. Objetivo general

Desarrollar un sistema domótico controlado mediante dispositivos móviles para la vigilancia remota de viviendas familiares.

1.2.2. Objetivos específicos

- Analizar y diseñar los requerimientos levantados para el sistema domótico.
- Integrar las tecnologías de automatización de viviendas.
- Codificar y realizar pruebas del sistema domótico DomoPit.

1.3. Justificación

Los habitantes de las viviendas familiares, están alertados por las incidencias de robo que se incrementan día a día y generan pérdidas económicas, no hay confianza en dejar sus viviendas sin la presencia de alguno de ellos, con los avances en la tecnología y en la infraestructura de internet se pueden proponer soluciones para contrarrestar esta causa, por ello se propone esta solución tecnológica para la vigilancia remota de las viviendas, haciendo uso de dispositivos móviles con acceso a internet, sensores que detecten presencia de intrusos, dispositivos electrónicos como lámparas, alarmas, televisores para simular la presencia de personas en las viviendas y para visualizar los hechos ocurridos en ellas el uso

de cámaras. Así las personas que realicen viajes familiares, las que se ausenten por mucho tiempo, podrían dejar sus casas sin ningún habitante con mayor confianza, teniendo en cuenta que mediante sus celulares podrían ver los hechos ocurridos y reaccionar ante ellos, controlando la iluminación, activando un sistema de alarmas, entre otras. Así se podría reducir las incidencias de robo y reducir costos con el uso de software y hardware libre.

Martínez (2015) en su investigación titulada Sistema de domótica para control y supervisión de una habitación de manera remota, implementó un sistema domótico haciendo uso de ordenadores de placa reducida como lo es Raspberry Pi, con sensores (sensor de movimiento) y actuadores, para el desarrollo de la aplicación móvil usa el lenguaje de programación JavaScript, y como base de datos MySQL, con estas herramientas soluciona el objetivo planteado.

En la investigación anterior citada, se demostró que la domótica, tiene una aplicación muy importante tratándose de controlar e integrar dispositivos electrónicos y controlarlos remotamente con la ayuda de una aplicación móvil, para la interacción del usuario con la interfaz gráfica. Para esta investigación se usará herramientas opensource y openhardware, como son arduino, sensor PIR, sensor Buzzer pasivo, Shield Ethernet. NativeScript-Vue para el desarrollo de la aplicación, y firebase como base de datos, estas herramientas nos ayudaran en desarrollar un sistema domótico de bajo costo. Además, con este trabajo, sería posible sentar las bases para obtener el siguiente beneficio: Visibilidad remota de la vivienda desde cualquier lugar donde se encuentren los dueños.

1.4. Presunción filosófica

Con este proyecto aportaría a la mejora de la calidad de vida de las personas que hagan uso de este sistema domótico, para prevenir las incidencias de robo, de esta manera se cumple con los principios de la iglesia, la cual sigue el ejemplo de Cristo, el cual siempre vela por nuestra seguridad e integridad física, también agradecer a Él por la sabiduría que nos brinda para poder aprovechar los recursos tecnológicos para ayudar a la sociedad.

CAPÍTULO II

MARCO TEÓRICO

2.1. Revisión de la literatura

Investigaciones acerca de la vigilancia remota

En la Pontificia Universidad Javeriana Bogotá(Colombia), Martínez (2015), realizó una investigación titulada “Sistema de domótica para control y supervisión de una habitación de manera remota” con el objetivo de controlar de manera remota una habitación a través de internet, para lo cual usó la metodología tradicional en cascada: análisis, diseño e implementación, pruebas. Se alcanzó el objetivo obteniendo conclusiones que van más allá de la posibilidad de controlar completamente una casa o un edificio, como una visión de lo que puede llegar a ser el futuro y de cuáles son las herramientas necesarias para participar en el.

En la Pontificia Universidad Católica del Perú, Guerra (2014), realizó una investigación titulada “Diseño de un sistema de control domótico y video vigilancia supervisado por un teléfono móvil” con el objetivo de diseñar un sistema de control domótico y de seguridad que permita supervisar la actividad en el hogar mediante sensores de movimiento, video cámaras, que permita tener puertas controladas remotamente; y que además brinde la facilidad de controlarlo a través de un dispositivo móvil, para lo cual usó la metodología tradicional en cascada: análisis, diseño e implementación, pruebas. Se llegó al resultado que al tratarse de un sistema que incluye servicios de red como la descarga de capturas desde el microcontrolador y la central de llamadas, es necesario analizar su consumo de ancho de banda para determinar su factibilidad en una red local. Para ello, se usó el software iptraf, el cual permite monitorear el envío y recepción de paquetes por una determinada interfaz.

En la Universidad de Las Américas Quito(Ecuador), Núñez (2017), realizó una investigación titulada “Implementación de un prototipo de un sistema domótico basado en

Arduino para control, monitorio y asistencia en hogares para personas de la tercera edad” con el objetivo de implementar un prototipo de un sistema domótico Basado en Arduino para controlar, monitorear y asistir hogares de personas de la tercera edad, para lo cual usó la metodología tradicional en cascada: análisis, diseño e implementación, pruebas. Los resultados en cuanto al uso del sensor PIR son los siguientes: se detecta si hubo movimiento del usuario en dicha habitación. Se verifica que el sensor encienda correctamente las luces de la habitación por un minuto. Este minuto se reinicia cada que el sensor detecta movimiento.

En la Universidad Nacional de Ingeniería(Perú), Paiz (2015), realizó una investigación titulada “Diseño e implementación de una conexión remota bidireccional usando teléfono móvil con sistema operativo Android para terminales de potencia aplicados en domótica” con el objetivo de controlar a distancia vía teléfono móvil con sistema operativo Android para terminales domóticos, para lo cual usó la metodología tradicional en cascada: análisis, diseño e implementación, pruebas. Los resultados fueron el control de una cámara de vigilancia de forma remota, que permite ver desde zonas alejadas con acceso a internet las imágenes que se muestran de la vivienda, implica el uso de un teléfono móvil con sistema operativo Android.

Investigación acerca de desarrollo móvil nativo

En la Universidad Nacional de Loja(Ecuador), Atiencia (2017), realizó una investigación titulada “Aplicación móvil multiplataforma que brinda información de las unidades de salud públicas y privadas de la ciudad de Loja” con el objetivo de desarrollar una aplicación móvil multiplataforma, que cuente con información actualizada online y detallada de los servicios que disponen las unidades de salud públicas y privadas de la ciudad de Loja, para lo cual usó la metodología XP: planificación, diseño, codificación, pruebas. Llegó a las siguientes conclusiones: la herramienta Advance Rest Client fue necesaria para realizar las pruebas de funcionalidad de los servicios web que son utilizados para el consumo de la aplicación móvil, las pruebas de carga se determinó la velocidad, el rendimiento, funcionalidad con la que el sistema realiza una determinada tarea, bajo condiciones particulares de trabajo, permitiendo validar y verificar la calidad del software. Obteniendo como resultado una aplicación de calidad.

Investigación acerca de domótica y seguridad

Núñez (2017) menciona que “la seguridad es uno de los campos más populares cuando se desea implementar un sistema domótico en una residencia, servicios que se encargan de proteger bienes y la integridad física de los usuarios. Un sistema de alarmas a estímulos externos, alertas médicas, sensores de puertas, detección de intrusos y simulación de presencia destacan entre los servicios de seguridad.”

González & Salcedo (2017) en su investigación titulada “Sistema de seguridad para locales comerciales mediante Raspberry Pi, cámara y sensor PIR” mencionan que “La tecnología permanece en constante desarrollo y afronta problemas como el de la seguridad; así entonces, para el desarrollo del sistema que se presenta en este artículo se tomaron una placa Raspberry Pi, una cámara para ese dispositivo y un sensor PIR, elementos de costo reducidos y con altos beneficios al integrarse. Como se evidencia una dependencia frente a los tiempos de robo, se considera como estrategia generar notificaciones mediante el envío al correo electrónico del dueño del local comercial: con esto se dispondrá de una imagen de lo ocurrido en caso de que el sensor haya detectado movimiento en un periodo de 10 segundos; este se reinicia exista o no la detección, con el fin de que el dueño tome decisiones efectivas y el sistema continúe sus labores de monitoreo”.

2.2. Bases teóricas

2.2.1. Vigilancia remota

Según Albusac (2008) el control remoto de las cámaras, incluye inclinación o el nivel de zoom, visión nocturna y detección de movimiento, que permite al sistema de alarmas cambiar a un estado de alerta ante posibles intrusiones. El sistema de vigilancia digital, más conocido es el sistema de vigilancia-IP, que transmite información en una red donde existen dispositivos conectados entre si(ordenadores que procesan las imágenes) y almacenan las imágenes con grabadores de vídeo digital (DVR). Las ventajas que presentan son múltiples, entre ellas, mayor facilidad de uso y mantenimiento, capacidades avanzadas de búsqueda, procesamiento de imágenes, accesibilidad remota, fácil integración con otros sistemas,

posibilidad de comprimir las imágenes, etc. Asimismo Aviles & Cobeña (2015) menciona que supervisa cada evento mientras se está lejos de casa o de la oficina. Se realiza la configuración de monitoreo y se ajusta los parámetros, tal como el nivel de 52 sensibilidad, enmascaramiento, registro de video y varias funciones de alertas tales como ftp o e-mail. También se puede controlar la calidad de los eventos guardados (videoclips) ajustando el formato de video o en la configuración de video avanzado.

2.2.2. Seguridad ciudadana

La seguridad ciudadana es la situación de tranquilidad pública y de libre ejercicio de los derechos individuales, cuya protección efectiva se encomienda a las fuerzas de orden público. En países como Islandia, Nueva Zelanda, Austria, Portugal y Dinamarca quienes lideran el ranking del índice de paz global, las políticas son de carácter igualitario, sus políticas de seguridad social son eficaces, es así cómo en estos países se reduce la tasa de inseguridad y de criminalidad. La educación igualitaria es la principal causa detrás de la casi ausencia de incidencias de robo, los hijos de los magnates van al mismo colegio que el resto de los niños, así la educación promueve la igualdad (BBC Mundo, 2013-2018) . En la Figura 2 podemos observar el ranking de los países con menos índice de inseguridad y de criminalidad.

RANK	COUNTRY	SCORE	CHANGE	RANK	COUNTRY	SCORE	CHANGE	RANK	COUNTRY	SCORE	CHANGE
1	Iceland	1.096	↔	29	Botswana	1.659	↓ 4	57	United Kingdom	1.876	↓ 6
2	New Zealand	1.192	↔	30	Spain	1.678	↓ 10	58	Montenegro	1.893	↑ 5
3	Austria	1.274	↑ 1	31	Latvia	1.689	↑ 1	59	Timor-Leste	1.895	↓ 5
4	Portugal	1.318	↓ 1	32	Poland	1.727	↑ 1	60	Vietnam	1.905	↔
5	Denmark	1.353	↔	33	Estonia	1.732	↑ 3	61	France	1.909	↓ 5
6	Canada	1.372	↔	34	Taiwan	1.736	↑ 3	62	Cyprus	1.913	↑ 3
7	Czech Republic	1.381	↔	35	Sierra Leone	1.74	↑ 5	63	Liberia	1.931	↑ 27
8	Singapore	1.382	↑ 3	36	Lithuania	1.749	↑ 2	64	Moldova	1.939	↔
9	Japan	1.391	↓ 1	37	Uruguay	1.761	↓ 2	65	Equatorial Guinea	1.946	↓ 7
10	Ireland	1.393	↑ 2	38	Italy	1.766	↑ 1	66	Argentina	1.947	↑ 8
11	Slovenia	1.396	↓ 1	38	Madagascar	1.766	↑ 4	67	Sri Lanka	1.954	↑ 5
12	Switzerland	1.407	↓ 3	40	Costa Rica	1.767	↓ 6	68	Nicaragua	1.96	↑ 7
13	Australia	1.435	↔	41	Ghana	1.772	↑ 6	69	Benin	1.973	↑ 12
14	Sweden	1.502	↔	42	Kuwait	1.799	↑ 5	70	Kazakhstan	1.974	↓ 2
15	Finland	1.506	↑ 3	43	Namibia	1.806	↑ 7	71	Morocco	1.979	↑ 4
16	Norway	1.519	↔	44	Malawi	1.811	↑ 8	72	Swaziland	1.98	↓ 2
17	Germany	1.531	↔	45	UAE	1.82	↑ 12	73	Oman	1.984	↓ 11
17	Hungary	1.531	↓ 2	46	Laos	1.821	↓ 2	74	Peru	1.986	↓ 1
19	Bhutan	1.545	↑ 5	46	Mongolia	1.821	↓ 1	75	Ecuador	1.987	↓ 8
20	Mauritius	1.548	↓ 1	48	Zambia	1.822	↓ 7	76	The Gambia	1.989	↑ 35
21	Belgium	1.56	↔	49	South Korea	1.823	↓ 6	77	Paraguay	1.997	↓ 8
22	Slovakia	1.568	↑ 3	50	Panama	1.826	↓ 4	78	Tunisia	1.998	↓ 7
23	Netherlands	1.574	↓ 1	51	Tanzania	1.837	↓ 2	79	Greece	2.02	↔
24	Romania	1.596	↑ 3	52	Albania	1.849	↑ 7	80	Burkina Faso	2.029	↑ 14
25	Malaysia	1.619	↑ 4	52	Senegal	1.849	↑ 9	81	Cuba	2.037	↑ 8
26	Bulgaria	1.635	↑ 2	54	Serbia	1.851	↑ 1	82	Guyana	2.043	↔
27	Croatia	1.639	↑ 4	55	Indonesia	1.853	↓ 2	83	Angola	2.048	↑ 9
28	Chile	1.649	↓ 5	56	Qatar	1.869	↓ 26	84	Nepal	2.053	↑ 4

Figura 2. Índice de paz global.

2.2.3. Domótica

Orientado por Spira (1961) se define a la domótica como una combinación de la tecnología de la información con la electrónica, estas dos especialidades hacen que un hogar sea inteligente, con la interacción de varios dispositivos tales como miniordenadores, sensores, actuadores y un teléfono inteligente. Según Paiz (2015) menciona que “el fin de la domótica es mejorar la seguridad, el confort, la flexibilidad, las comunicaciones, el ahorro energético, facilitar el control de los sistemas para los usuarios”. La domótica tiene 3 arquitecturas que son las siguientes:

2.2.3.1. *Arquitectura centralizada*

KJURADOE (2013) nos dice que “La arquitectura domótica centralizada consiste en contar con un controlador central, el cual recibe datos de los diferentes sensores (iluminación, movimiento, humo, etc.) para después enviar órdenes a los actuadores (cámaras, focos led, etc) e interfaces”.

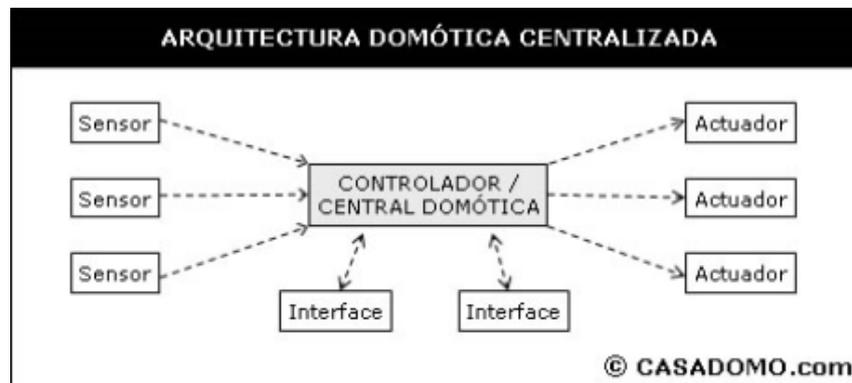


Figura 3. Arquitectura centralizada.

Fuente: KJURADOE, 2013.

2.2.3.2. *Arquitectura distribuida*

KJURADOE (2013) menciona que “La arquitectura domótica distribuida consiste en que los sensores y actuadores están separados por módulos, esto se presenta en lo que son las redes inalámbricas”.

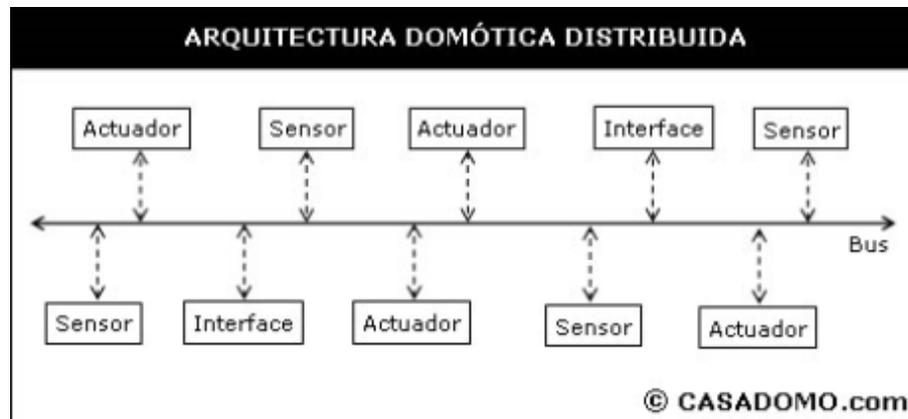


Figura 4. Arquitectura distribuida.

Fuente: KJURADOE, 2013.

2.2.3.3. *Arquitectura mixta*

KJURADOE (2013) menciona que “La arquitectura domótica mixta es una combinación de la centralizada y distribuida”.

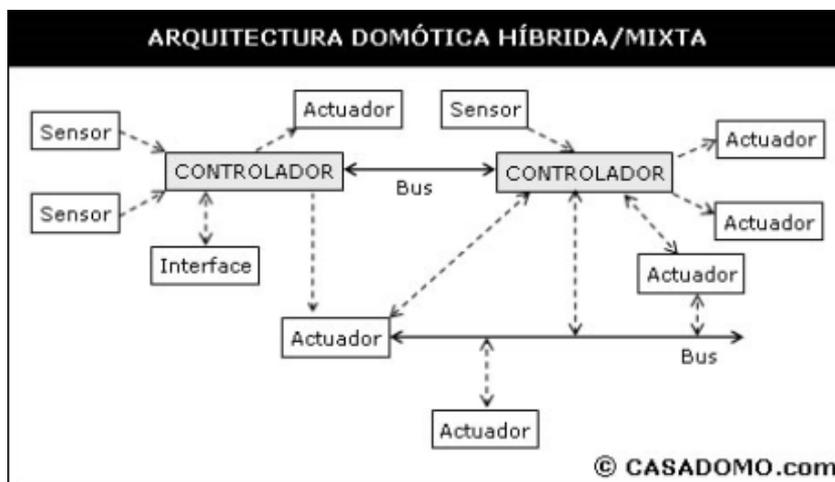


Figura 5. Arquitectura mixta.

Fuente: KJURADOE, 2013.

En la *Figura 6* podemos ver, la arquitectura mostrada en componentes que forman un hogar inteligente, la cual consiste en contar con un controlador central, el cual recibe

datos de los diferentes sensores (iluminación, movimiento, humo, etc.) para después enviar órdenes a los actuadores (cámaras, focos, etc) e interfaces.

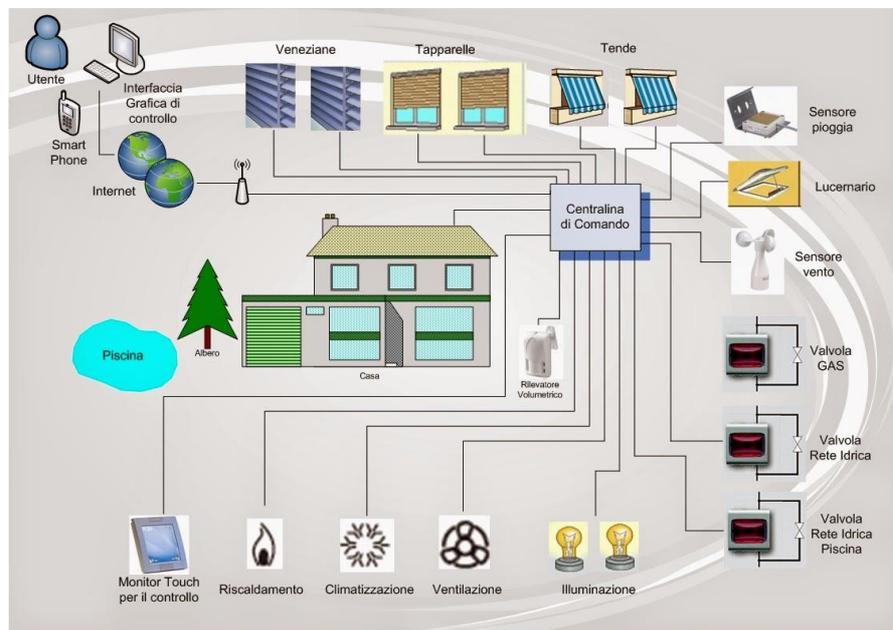


Figura 6. Componentes en un hogar inteligente.

Fuente: tecnologia, 2014.

2.2.3.4. Vigilancia remota en domótica

La domótica tiene aplicación en varios campos, uno de ellos es la seguridad, dentro de ello podemos hablar de la vigilancia remota. Los dispositivos a usarse para implementar un sistema de vigilancia remota, son las cámaras IP, que tienen un sensor de imagen, también un microcontrolador (Arduino) para el envío de órdenes a los dispositivos electrónicos.

2.2.4. Microcontrolador

Un microcontrolador es un circuito integrado que está diseñado para distintas funciones, los microcontroladores están formados por al menos tres elementos: procesador, memoria y dispositivos de entrada y salida, son también considerados como ordenadores de menor instrucción, son económicos y sirven para implementar proyectos pequeños o grandes, cada microcontrolador se diseña de acuerdo a los usos que se le va a dar por lo cual están en un punto entre un simple dispositivo y un potente ordenador. Anteriormente los microcontroladores podían ser usados como un dispositivo simple, ideal para usar en grandes

y pequeñas empresas para experimentar con pequeños proyectos, al pasar los años estos han ido evolucionando, ahora en el mercado se puede encontrar todo tipo de microcontroladores en las placas pequeñas, todos estos diseñados de distinta forma y con distintas funcionalidades, dentro de estas placas tenemos Arduino, Raspberry Pi, etc.



Figura 7. Arquitectura del microcontrolador, arquitectura Harvard.

Fuente: SHERLIN, 2018.

En la Figura 7 observamos que la arquitectura Harvard tiene tres elementos principales, memoria de programa, CPU y memoria de datos. En la actualidad la mayoría de los microcontroladores están bajo esta arquitectura porque tiene la ventaja de trabajar con dos memorias simultáneamente, esto mejora la velocidad de ejecución de los programas que se cargan en los microcontroladores.

Tabla 1.
Tabla de comparación de microcontroladores

HERRAMIENTAS	CARACTERÍSTICAS
Arduino Uno	Permite la captura de datos en tiempo real, voltaje de entrada 7-12v, 14 pines digitales, 32k de memoria flash.
Raspberry Pi 3 Modelo B	Socket ethernet, salida de video, 27 pines digitales, LAN inalámbrica, pantalla de visualización.

Fuente: Elaboración propia. Según Arduino & Raspberry Pi.

Para este proyecto se hará uso de la placa Arduino, por la captura de datos en tiempo real, es más económico, es suficiente para abarcar el proyecto.

2.2.4.1. *Arduino*

Arduino (2018) “Arduino es una plataforma electrónica de código abierto basada en hardware y software fácil de usar. Las placas Arduino pueden leer entradas (luz en un sensor, un dedo en un botón o un mensaje de Twitter) y convertirlo en una salida, activar un motor, encender un LED y publicar algo en línea. Puede decirle a su tablero qué hacer enviando un conjunto de instrucciones al microcontrolador en el tablero. Para hacerlo, utiliza el lenguaje de programación Arduino(basado en el cableado) y el software Arduino (IDE) , basado en el procesamiento.”

Núñez (2017) en su investigación menciona que siendo Arduino de hardware y software libre da paso a la personalización de la plataforma de acuerdo a las necesidades del usuario. La placa Arduino es de licencia Creative Commons (Arduino.CC, s.f.), por ello el usuario puede hacer uso de la plataforma sin ningún problema legal. Por su arquitectura Arduino puede expandirse en el uso de sus microcontroladores.

Mejía (2016) menciona que cuando se busca implementar algo de automatización no se incrementa el desgaste físico a diferencia de las tareas sin automatización, en este caso el sistema debe ser controlado desde cualquier dispositivo, “ya sea con una interfaz dedicada, vía web por medio de un navegador o incluso a través de la línea telefónica” disminuyendo la necesidad de conexión a Internet, así se incrementa la posibilidad de usar “cualquier dispositivo móvil y no sólo aparatos electrónicos conectados o teléfonos inteligentes y tabletas”.

1. Arquitectura RISC de arduino

Orientado por Aviles & Cobeña (2015) se define que “La arquitectura RISC reduce el tiempo de ejecución simplificando las instrucciones que se dan a una computadora, su CPU tiene instrucciones sencillas, su operación es sencilla, y son procesados a una alta velocidad”.

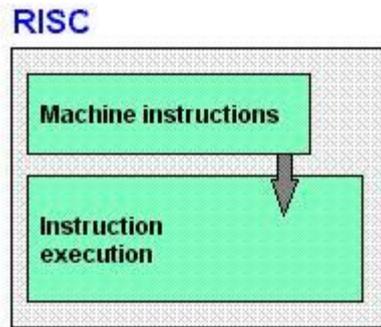


Figura 8. Arquitectura RISC, es la arquitectura de la placa Arduino.

Fuente: 4BP, n.d.

2. IDE Arduino

“Es el software que ofrece la comunidad de arduino, el cual usa un lenguaje de programación que está basado en el cableado. El entorno está escrito en java y a la vez está basado en Processing otro IDE de desarrollo de código abierto”. (DCervantes, 2014)

The screenshot shows the Arduino IDE interface with a code editor window titled 'Blink'. The code is as follows:

```

/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  This example code is in the public domain.
  */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);             // wait for a second
}

```

The IDE status bar at the bottom indicates '1 Arduino Uno on COM1'.

Figura 9. IDE Arduino.

Fuente: DCervantes, 2014.

Arduino y su impacto en la domótica, la placa fue inventada por un estudiante del instituto IVRAE, llamado Massimo Banzi. Banzi, inventó arduino por el costo elevado que tenían los microcontroladores en el mercado, fue creado con fines de aprendizaje, no llegó a

imaginar que años más tarde esto se convertiría en una de las tecnologías líder en DIY, tanto que ahora es un componente importante dentro de la automatización de cualquier objeto electrónico, es así como esta placa es integrada en muchos proyectos domóticos, sean grandes o pequeños (Arduino, 2018).

En los proyectos domóticos, el uso de los microcontroladores es habitual, en esta investigación se usará la placa Arduino Uno R3, como mencionan los anteriores investigadores la comunidad Arduino es de open hardware y open software esto nos brinda la seguridad de que no atravesemos por una persecución legal.



Figura 10. Arduino Uno R3.

Fuente: Propia, 2018.

2.2.4.2. *Arduino Ethernet Shield*

Lledó (2012) define este módulo “como una placa que se acopla encima del Arduino y permite establecer conexiones a internet mediante el estándar Ethernet que utiliza el protocolo TCP/IP”, la cual se puede conectar a un router con un cable RJ45 y se le asigna una dirección IP, con esto ya tendríamos un servidor para enviar flujos de datos de la placa.

Paz, Rodarte & Pérez (n.d.) mencionan que “El Arduino Ethernet Shield, permite a una placa Arduino conectarse a internet. Está basado en el chip Ethernet WIZnet W5100, que maneja protocolos IP y TCP como UDP. Soporta hasta cuatro conexiones de socket

simultáneamente. Utiliza librería Ethernet para escribir programas, los cuales se conectan a internet utilizándolo”

En las anteriores investigaciones mencionadas nos damos cuenta que el Shield Ethernet montado en el Arduino nos permitirá el acceso a internet, mediante la conexión a un router, a través de un cable RJ45 o cable de red, en el cual podremos levantar nuestro servidor utilizando una dirección IP asignada por la red.

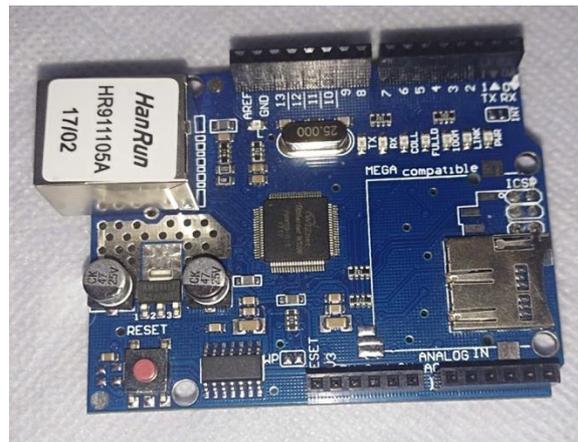


Figura 11. Arduino Ethernet Shield.

Fuente: Propia, 2018.

2.2.5. Sensores

2.2.5.1. *Sensor de movimiento PIR*

Es un sensor de infrarrojo pasivo y piro eléctrico, detecta movimiento mediante cambios de calor en algún cuerpo, como respuesta a ese cambio, el nivel lógico de su pin cambia de 0 a 1, las ventajas de este sensor es que son de bajo costo y la desventaja es que solo detecta cuerpos que emiten calor (personas y animales), es usado mayormente en proyectos de esta magnitud (Flotante, 2018).

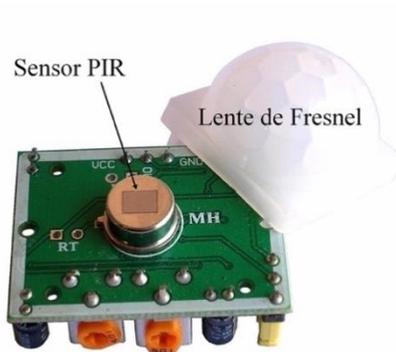


Figura 12. Sensor PIR.

Fuente: Flotante, 2018.

Guerra (2014) afirma que es un sensor pasivo infrarrojo y su funcionamiento se basa en la detección de la radiación infrarroja que emiten los cuerpos, esta radiación infrarroja varía por su movimiento, “un sensor PIR detecta la variación en los patrones de radiación en el ambiente, y envía un pulso a través de una salida que puede conectarse a un microcontrolador.” Asimismo Núñez (2017) en su investigación menciona que “este sensor es usado en proyectos auto-sensibles y con batería como fuente de energía”. En su proyecto fue usado en cada habitación de la casa prototipo para detectar presencia, para encender luces y alarmas en horas determinadas.

2.2.5.2. *Sensor buzzer pasivo Arduino*

Llamas (2016) da a conocer que es este un sensor que genera sonido a una frecuencia determinada y fija, esto se da al conectar a una tensión. “Un buzzer activo incorpora un oscilador simple por lo que únicamente es necesario suministrar corriente al dispositivo para que emita sonido”. Estos sensores necesitan recibir ondas de frecuencia, al incorporar de forma interna “la electrónica necesaria para hacer vibrar el altavoz un buzzer activo resulta muy sencillo de conectar y controlar”.

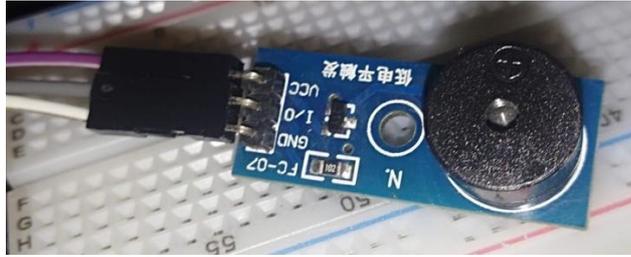


Figura 13. Sensor buzzer pasivo Arduino.

Fuente: Propia, 2018.

2.2.6. Actuadores

2.2.6.1. Cámara IP

Paiz (2015) menciona que las cámaras IP facilitan la visualización de los hechos de un lugar en tiempo real, aunque no se esté cerca al dispositivo. “Son cámaras de vídeo de gran calidad que tienen incluido un ordenador a través del que se conectan directamente a Internet”. Se conectan a través de Internet, asignando una dirección IP o esta ya viene con la cámara, la configuración es simple para su funcionamiento.

Guerra (2014) menciona que las cámaras IP son similares, y todas cuentan con tecnología IP, asimismo compresión JPEG, conexión inalámbrica mediante Wifi, pero una es mas económica que las otras. Entonces se puede escoger el modelo de acuerdo a las necesidades de cada investigador.



Figura 14. Cámara IP.

Fuente: MercadoLibre, 2018.

2.2.6.2. Focos LED

Para conseguir un ahorro energético adecuado, es necesario apagar las luces automáticamente al poco tiempo de detectar falta de presencia, con la programación adecuada del sistema domótico, que espera cierto tiempo a la última detección de presencia producida antes de dar la orden de apagado al sistema (Domoprac, 2018). En esta investigación usaremos focos led y no los focos normales, como el propósito es desarrollar un sistema de bajo costo, los focos led reducen el consumo energético.



Figura 15. Focos LED.

Fuente: LOWES, 2016.

2.2.7. Relé

Es considerado como un interruptor, este se encarga de abrir y cerrar los circuitos de forma electromecánica, la operación principal del relé es hacer o interrumpir el contacto con ayuda de una señal enviada de un microcontrolador, así se puede encender o apagar un artefacto electrónico sin la intervención de un humano (Thonti, 2017).

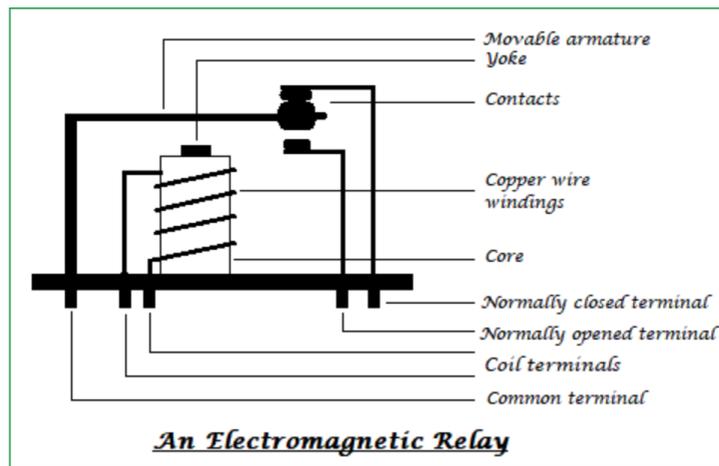


Figura 16. Vista interna del relé y cómo se puede construir.

Fuente: CircuitDigest, 2018.

En una carcasa, se coloca un núcleo con bobinas de cobre (forma una bobina) sin viento. Una armadura móvil consiste en un soporte de resorte a un soporte conectada a un extremo, y un contacto de metal conectado a otro lado, todas estas disposiciones se colocan sobre el núcleo de manera que, cuando se activa la bobina, atrae la armadura. La armadura móvil generalmente se considera como un terminal común que se va a conectar a la circuitería externa (Thonti, 2017).

Guerra (2014) menciona que un relé es un dispositivo que funciona igual que un interruptor, permite apagar y encender dispositivos electrónicos asociados a ella, esto se hace mediante una señal eléctrica proveniente de un microcontrolador. En su investigación, el relé es usado como interruptor para abrir y cerrar la iluminación y la red eléctrica de la casa. Se integra con el microcontrolador Raspberry Pi, esta integración con la finalidad de que el microcontrolador pueda dar órdenes al relé, en qué momento abrir y cerrar circuitos. En nuestra investigación se usará un relé, este se encargará de encender y apagar una el foco led y la alarma. Mediante las ordenes que enviará el microcontrolador Arduino.

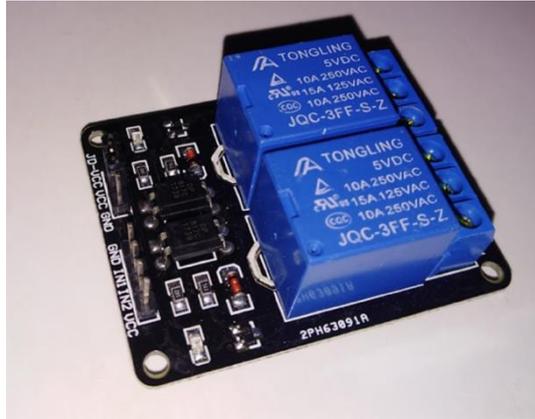


Figura 17. Relé de dos canales.

Fuente: Propia, 2018.

2.2.8. No-IP

“Es un servicio de DNS dinámico que permite identificar tu PC con un nombre de dominio fácil de recordar en lugar de una dirección IP, además de ello podemos montar un servidor sin complicaciones independientemente de si tenemos o no una IP estática” (MundoGeek, 2019). Los proveedores del servicio de internet como Movistar, no nos brindan IP estáticas, sean locales o públicas, lo que nos brindan son IP dinámicas, así que en caso que intentemos levantar un servidor, tendríamos que usar No-IP para obtener un dominio en el cual no importa si la IP pública de nuestro router cambia, podríamos acceder al servicio desde el dominio creado en No-IP.



Figura 18. No-IP.

Fuente: No-ip, 2019.

2.2.9. IP dinámica

Hernandez & Parrales (2011) mencionan que la IP dinámica “Es una dirección IP asignada mediante un servidor DHCP al usuario. La IP que se obtiene tiene una duración máxima determinada. El servidor DHCP provee parámetros de configuración específicos para cada cliente que desee conectarse a la red”. La mayoría de los proveedores de internet ofrecen este tipo de IP, estas IP varían cada vez que el usuario se reconecta a la red.

2.2.10. IP estática

Esta IP es asignada manualmente o “por el servidor de la red(ISP en el caso de internet, router o switch en caso de LAN) con base en la dirección MAC del cliente” (Adaptix, 2019).

2.2.11. IP pública

La dirección IP pública es aquella que ofrece un proveedor que brinda el acceso a internet y se le da a cualquier equipo o dispositivo conectado de forma directa a internet. “Las direcciones IP públicas son siempre únicas, no se pueden repetir. Dos equipos con IP de este tipo pueden conectarse directamente entre sí”(Adaptix, 2019).

2.2.12. Redes de área local (LAN)

Hernandez & Parrales (2011) mencionan que son redes privadas y la conexión de las computadoras se debe hacer en un solo área, por ejemplo, computadoras personales y lugares como una oficina, donde solo comparten datos entre ellos, “la conexión en esta red es mediante cable coaxial, fibra óptica o cable UTP”, para la conexión inalámbrica se puede usar la tecnología infrarroja, “estas redes emplean protocolos o reglas para intercambiar información, evitando el tráfico de datos”.

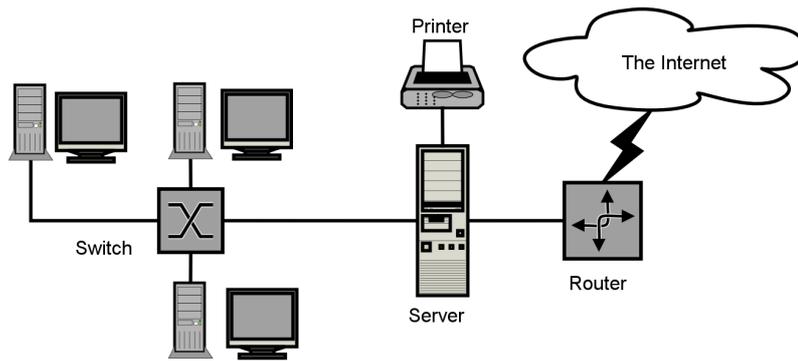


Figura 19. Red LAN.

Fuente: IONOS, 2019.

2.2.13. Redes de área amplia (WAN)

Hernandez & Parrales (2011) dan a conocer que esta red es extensa geográficamente que se desenvuelve en un país o continente, utiliza computadoras “hosts conectadas por una subred de comunicaciones para conducir mensajes de una host a otra”, en redes grandes de subred tiene dos componentes “las líneas de transmisión y los elementos de conmutación que son computadoras especializadas que conectan dos o mas líneas de transmisión, las WAN contienen numerosos cables y usan enrutadores”.

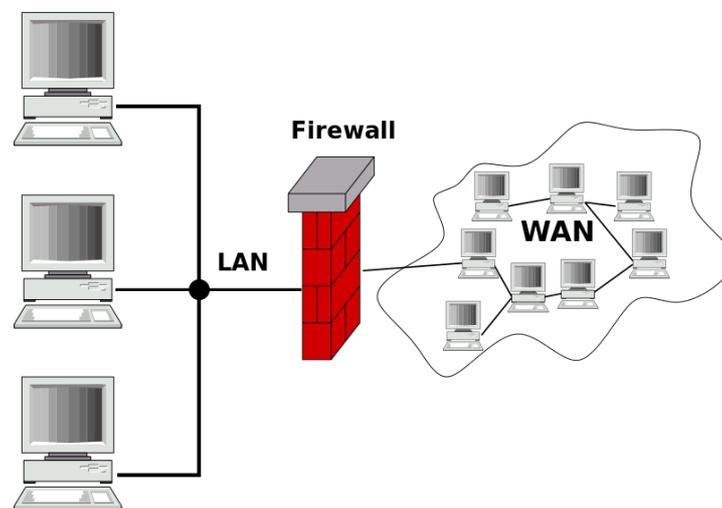


Figura 20. Red WAN.

Fuente: IONOS, 2019.

2.2.14. Protocolo HTTP

Prieto (2007) define “El protocolo HTTP es un sencillo protocolo cliente-servidor que articula los intercambios de información entre los clientes web y los servidores HTTP. Propuesta por Tim Berners-Lee, atendiendo a las necesidades de un sistema global de distribución de información como World Wide Web”.

2.2.15. Android

Android es un sistema operativo móvil perteneciente a Google, este sistema operativo móvil es open source y gratuito, está basado en Linux, fue una revolución total en el mercado de los teléfonos inteligentes, en la actualidad la mayoría de los teléfonos inteligentes funcionan bajo esta plataforma, es adaptable a cualquier marca de teléfonos inteligentes y sumamos a esto las tabletas (Developers, 2018).



Figura 21. Componentes principales de la plataforma Android.

Fuente: Developers, 2018.

La ventaja del sistema operativo Android es que es opensource, por lo cual cualquiera puede desarrollar una aplicación para Android, otra ventaja de este sistema operativo es que en la actualidad existen 650.000 aplicaciones disponibles para esta plataforma, la libertad del código permite que este sistema operativo se adapte no solo en Smartphones sino también en Tablets, GPS, relojes y último para PC (Scoello, 2012).

2.2.16. Aplicación móvil

Es un software informático que tiene funcionamiento en teléfonos móviles y tabletas. Existen distintos tipos de aplicaciones, dirigidos a la educación, al ocio, acceso a servicios, etc. Estas aplicaciones les dan la facilidad de acceso rápido al usuario desde donde se encuentre, por ejemplo, a sus cuentas de email. Algunas de estas aplicaciones pueden ser encontradas en la tienda de Google Play Store en los teléfonos inteligentes con sistema operativo Android. (Techopedia, 2019)

Tabla 2.
Tabla de comparación entre aplicación híbrida y aplicación nativa

HERRAMIENTAS	CARACTERÍSTICAS
Aplicación híbrida	Menor inversión en el desarrollo, mantenimiento en una sola fuente de código.
Aplicación nativa	Experiencia de usuario, más velocidad de ejecución y respuesta, función offline, sincronización asíncrona, seguridad.

Fuente: Elaboración propia. Según Cuatroochenta.

La diferencia entre estos dos tipos de aplicación móvil no es como o con que lenguaje se programan, si no como se ejecuta para el usuario final. En esta investigación se desarrollará una aplicación nativa porque se ejecutan directamente en el sistema operativo.

2.2.17. Desarrollo móvil nativo

Una de las confusiones que existen al momento de desarrollar un aplicativo móvil, es decir que si no se usa Swift o Java es desarrollo móvil híbrido, pero con las nuevas tecnologías no es así. La diferencia entre una aplicación móvil nativa y una híbrida no es como va ser programada, si no como quedan para el usuario final. Cuando una aplicación

móvil requiere algo adicional para su ejecución como contenedores o navegadores, se considera híbrida. Como ejemplos tenemos a Apache Cordova, Ionic y Unity. Cuando una aplicación móvil es nativa, se ejecuta directamente en el sistema operativo, como por ejemplo ReactNative, NativeScript y Xamarin (Uriel, 2017).

Vesely (2017) menciona que “el método mayor es la creación de una única aplicación múltiples veces para cada plataforma de destino por separado”. Es por ello que, no es un desarrollo para varias plataformas, cada plataforma tiene numerosas ventajas que se pueden utilizar en su total o solo lo que se necesita en características y rendimiento. Por lo tanto, “estas aplicaciones también serán los más rápidos y los más suaves”. Por otra parte, “a fin de orientar para las plataformas, tenemos que crear nuestras aplicaciones. Esa es la razón por la cual los marcos cruzados para plataformas móviles entraron en la escena”.

Atencia (2017) menciona que “las aplicaciones nativas tienen la capacidad de aprovechar hardware y software específico del dispositivo, interactúan con toda la API nativa del dispositivo (cámara, GPS, acelerómetro, agenda, entre otras) por ello tiene una ejecución rápida”.



Figura 22. Aplicación nativa e híbrida.

Fuente: Uriel, 2017.

2.2.18. Lenguaje de programación

Es el idioma artificial de las computadoras, se usa para la creación de softwares y estos pueden controlar la parte física y lógica de una computadora o una máquina, para dar instrucciones de pasos que debe seguir, para poder comprenderse con el ser humano. Está formado por un conjunto de expresiones, símbolos y reglas sintácticas, estas dan sentido y significado a su estructura. Partiendo de todo esto se crean los softwares.

Tabla 3.

Tabla de comparación de lenguajes de programación

HERRAMIENTAS	CARACTERÍSTICAS
JavaScript	Maneja paradigmas de programación, como programación orientada a objetos, funcional e imperativa, menos programación procedimental.
Python	Maneja varios paradigmas de programación, como programación orientada a objetos, procedimientos, funcional e imperativa.

Fuente: Elaboración propia. Según JavaScript & Python.

Una de las ventajas de JavaScript es que es multiplataforma, es soportado por todos los navegadores más populares, puede ser ejecutado de manera híbrida en cualquier sistema operativo móvil, JavaScript es un lenguaje sencillo de aprender.

2.2.19. Frameworks

En la actualidad existen varios frameworks, la principal función de estos es agilizar el desarrollo de las aplicaciones. En la Tabla 5 haremos una comparación de frameworks para el desarrollo de aplicaciones móviles.

Tabla 4.
Tabla de comparación de frameworks

HERRAMIENTAS	CARACTERÍSTICAS
NativeScript	Permite el acceso a toda la API nativa, soporta múltiples frameworks y bibliotecas de JavaScript, es de fuente abierta.
React Native	Cuenta con componentes de iOS nativos, basado en solo React.
Ionic	Reutiliza el código de Angular frontend, prototipado rápido.

Fuente: Elaboración propia. Según Stackshare.

Cada uno de estos framework tiene sus bondades, y son usados de acuerdo a la aplicación que se desea realizar, en esta investigación se usará NativeScript, porque “soporta los múltiples framework y bibliotecas con las que cuenta JavaScript, por ejemplo, el framework Vuejs. NativeScript es apreciado por sus usuarios ya que les permite acceder a toda API nativa, puede ser a la cámara, contactos, etc” (NativeScript, 2018a).

2.2.19.1. Framework NativeScript

Rendimiento nativo, Interfaz de usuario hermosa, accesible y nativa de la plataforma, sin visitas web. Defina una vez y deje que NativeScript se adapte para ejecutarse en cualquier lugar, o adapte la interfaz de usuario a dispositivos y pantallas específicos. Esta es una de las razones por las que SAP eligió NativeScript. Extensible, Reutilice fácilmente los complementos existentes de npm, CocoaPods (iOS) y Gradle (Android) directamente en proyectos de NativeScript, además de cientos de complementos de NativeScript (NativeScript, 2018a).

Fácil de aprender, Utilice las habilidades web que ya tiene para desarrollar aplicaciones verdaderamente nativas con JavaScript, CSS y el marcado UI nativo. Multiplataforma, escriba e implemente aplicaciones móviles nativas para iOS y Android desde un solo código base. Use Angular o Vue para compartir código existente basado en web. Fuerte respaldo, respaldado por Progress (NASDAQ: PRGS) - Décadas de experiencia

en la creación y soporte de software para desarrolladores. El apoyo empresarial de misión crítica está disponible. Libre y de código abierto, NativeScript es 100% gratuito y de código abierto , con licencia Apache 2 (NativeScript, 2018a).

Vesely (2017) menciona que “cuando se construye una aplicación con NativeScript se puede elegir si escribir código en JavaScript puro o utilizar el mecanografiado”. A pesar de que el mecanografiado ofrece comprobación de errores útiles y finalización de código avanzado algunos desarrolladores aún lo usan.

1. NativeScript-Vue

Estructura de la aplicación, Vue.js proporciona toda la plomería necesaria para crear aplicaciones de alta calidad y altamente mantenibles, y puede reutilizar una cantidad considerable de su conocimiento existente de Vue.js e incluso código basado en la web para construir su aplicación NativeScript-Vue. Rendimiento móvil nativo, con NativeScript, puede crear aplicaciones móviles sin compromiso con Vue.js que cuentan con una interfaz de usuario y un rendimiento verdaderamente nativos. Disfruta de gloriosos 60 fps sin ser un asistente de CSS o JavaScript, incluso en Android. Dile adiós a jank móvil. Vue y NativeScript-Vue Community, aproveche la comunidad de desarrolladores de Vue.js en expansión para obtener ayuda adicional, capacitación y código reutilizable que se puede aplicar directamente a los proyectos móviles de NativeScript-Vue. Únase a la comunidad en Slack para ayudar a que este proyecto de código abierto, creado por la comunidad, crezca y madure (NativeScript, 2018b).

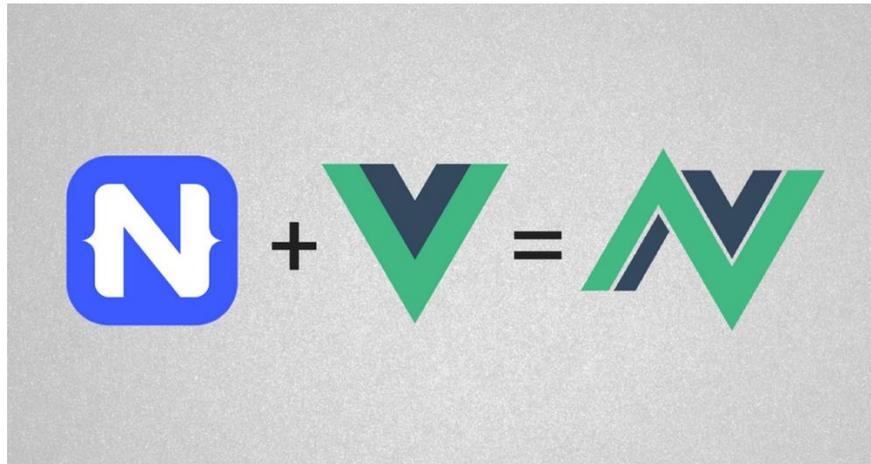


Figura 23. NativeScript-Vue.

Fuente: VanToll, 2018.

2.2.20. Base de datos

Una base de datos es un lugar donde se almacena datos que son parte de un solo contexto, estos datos son almacenados para sus usos futuros. En informática la base de datos es un sistema que está formado por una cantidad de información y a la vez esta información está almacenada en discos, a los cuales se puede acceder directamente, estos son manipulados por un software que los administra. Las bases de datos se conforman por tablas o entidades que tienen sus respectivos atributos y sus valores que se distribuyen en filas y columnas respectivamente.

Tabla 5.

Tabla de comparación entre base de datos relacional y no relacional

HERRAMIENTAS	CARACTERÍSTICAS
Base de datos relacional	Se usa una relación de identificadores, son más robustos y menos vulnerables.
Base de datos no relacional	No hay necesidad de un identificador, la información se organiza en documentos, son más flexibles, manejan los datos en tiempo real.

Fuente: Elaboración propia. Según Aukera.

2.2.20.1. *Firestore*

Es una base de datos no relacional, y una plataforma de desarrollo para aplicaciones móviles, tiene muchos servicios, entre ellos el servicio de base de datos en tiempo real y back-end, este servicio proporciona una API que sincroniza y almacena datos en la nube de Firestore. Firestore tiene integración con aplicaciones híbridas y nativas (Cyron, 2016).

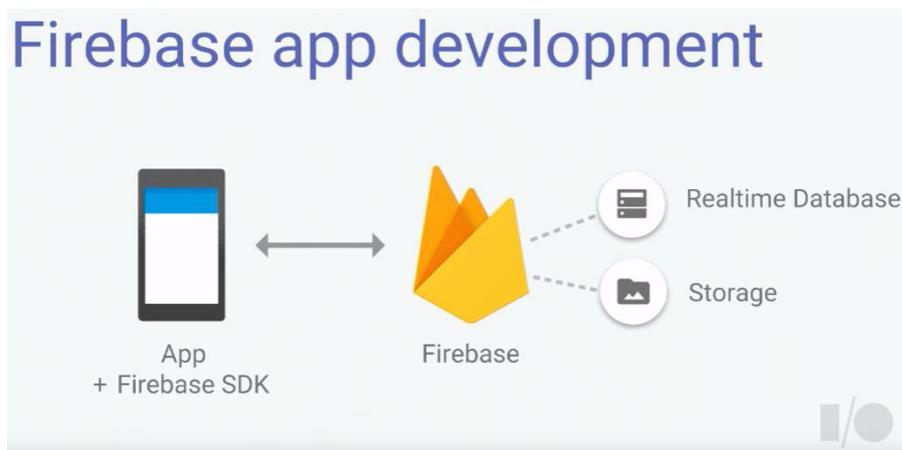


Figura 24. Funcionalidad de Firestore.

Fuente: Cyron, 2016.

Base de datos en tiempo real. Almacena y sincroniza datos NoSQL los aloja en la nube. Los datos se sincronizan con distintos clientes en tiempo real y siempre están disponibles cuando la aplicación no está en conexión. Los datos son almacenados en formato JSON y se sincronizan en tiempo real a los clientes conectados. Cuando se compila la aplicación multiplataforma con los SDK de iOS, Android y JavaScript, todos los clientes comparten una dato en tiempo real y también reciben actualizaciones automáticamente con los datos más recientes (Firestore, 2018).

2.2.21. *Metodología ágil XP*

Metodología XP, Borja (2013) para el desarrollo de software, y que se ajusta a reglas que se centran en la necesidad del cliente para lograr un producto de buena calidad en poco tiempo, diseñada para el desarrollo de aplicaciones que requieran un grupo de programadores pequeño, dónde la comunicación sea más factible que en grupos de desarrollo grandes.

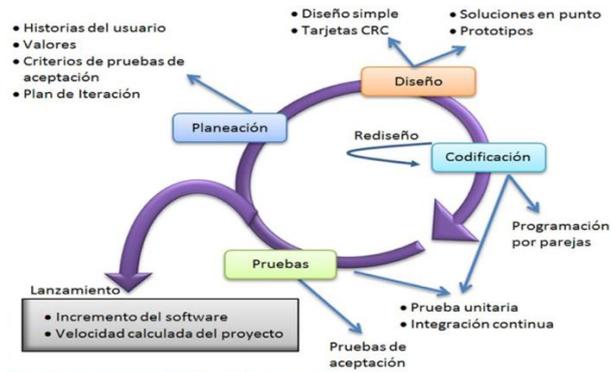


Figura 25. Metodología XP.

Fuente: Ramírez, 2017.

La metodología XP, tiene cuatro pasos, cada uno de estos pasos tiene actividades, esos cuatro pasos son los siguientes:

- Planificación.
- Diseño.
- Codificación.
- Pruebas.

Cevallos (2015) XP establece valores para realizar un trabajo. Cada uno de estos valores se usa como un motor para actividades, acciones y tareas específicas.

- Simplicidad.
- Comunicación.
- Retroalimentación.
- Coraje o valentía.
- Respeto.

Tabla 6.
Comparación de las metodologías ágiles SCRUM y XP.

Metodología SCRUM	Metodología XP
Enfocada a la administración del proyecto.	Enfocada en la programación y creación del producto.
Puede modificar el orden de prioridades establecido por el Product Owner en el Spring Backlog.	Sigue estrictamente el orden de prioridad de las actividades definidas por el cliente.
Cuenta con una estructura más jerárquica y organizada	Su estructura es más cambiante y menos organizada.
Las iteraciones de entrega se realizan cada 2-4 semanas.	Las iteraciones de entrega son de 1-3 semanas.
Al término de un Sprint, las tareas realizadas durante el Sprint Backlog y aprobadas por el cliente no se vuelven a modificar.	Las tareas entregadas al cliente son susceptibles a modificaciones durante el proyecto, incluso si funcionan correctamente.

Fuente: Elaboración propia según (Bahit, 2011) & (Ingeniería de Software, 2017).

CAPÍTULO III

MATERIALES Y MÉTODOS

3.1. Descripción del lugar de ejecución

El desarrollo de esta investigación se sitúa en una vivienda de la ciudad de Juliaca ubicada en la Urbanización Espinal, Salida Arequipa, la cual cuenta con autos, laptops, y más objetos de valor.

3.2. Metodología de la investigación

3.2.1. Tipo de investigación

Tipo de investigación propositiva y tecnológica. La investigación propositiva se sustenta con la siguiente fórmula:

$$\textit{Teoría} + \textit{Hecho} + \textit{Solución} = \textit{Investigación Propositiva}$$

Según Charaja (2011) “La investigación propositiva se fundamenta en una necesidad o vacío. Cuando se tiene la información descrita, se realiza una propuesta para contrarrestar la problemática actual y las deficiencias halladas”.

Según A. Dean (n.d.) afirma que “la investigación tecnológica resalta la presencia de un estado cognitivo propio de la ingeniería donde se destaca una potencial actitud innovadora de los involucrados en el proceso de la investigación”. Se refiere al diseñador que junta elementos para realizar combinaciones nuevas, tiene la capacidad de integrar y de manipular en su mente dispositivos, sistemas y aparatos que todavía no existen.

3.3. Materiales e insumos

A continuación, se mencionan los principales materiales que se usaron para esta investigación.

3.3.1. Hardware

- Arduino Uno R3
- Arduino Shield Ethernet
- Relé de un canal
- Sensor PIR
- Sensor Buzzer pasivo Arduino
- Cámara IP con resolución de 1920x1080p.
- Foco led
- Cables dupont
- Smartphone para prueba
- Cable UTP
- Conector ethernet estándar RJ45

Las especificaciones de estos se pueden apreciar en el *Anexo A*.

3.3.2. Software

- IDE Arduino versión 1.8.8
- IDE Visual Studio Code versión 1.33.1
- Librería SPI.h
- Librería Ethernet.h
- Framework NativeScript versión 5.2.0
- Plugin “nativescript-plugin-firebase” versión 7.1.6
- Plugin “nativescript-open-app” versión 0.2.0
- Plugin “nativescript-loading-indicator” versión 2.4.0
- Plugin “nativescript-vue” versión 2.0.0

3.3.3. Arquitectura de solución

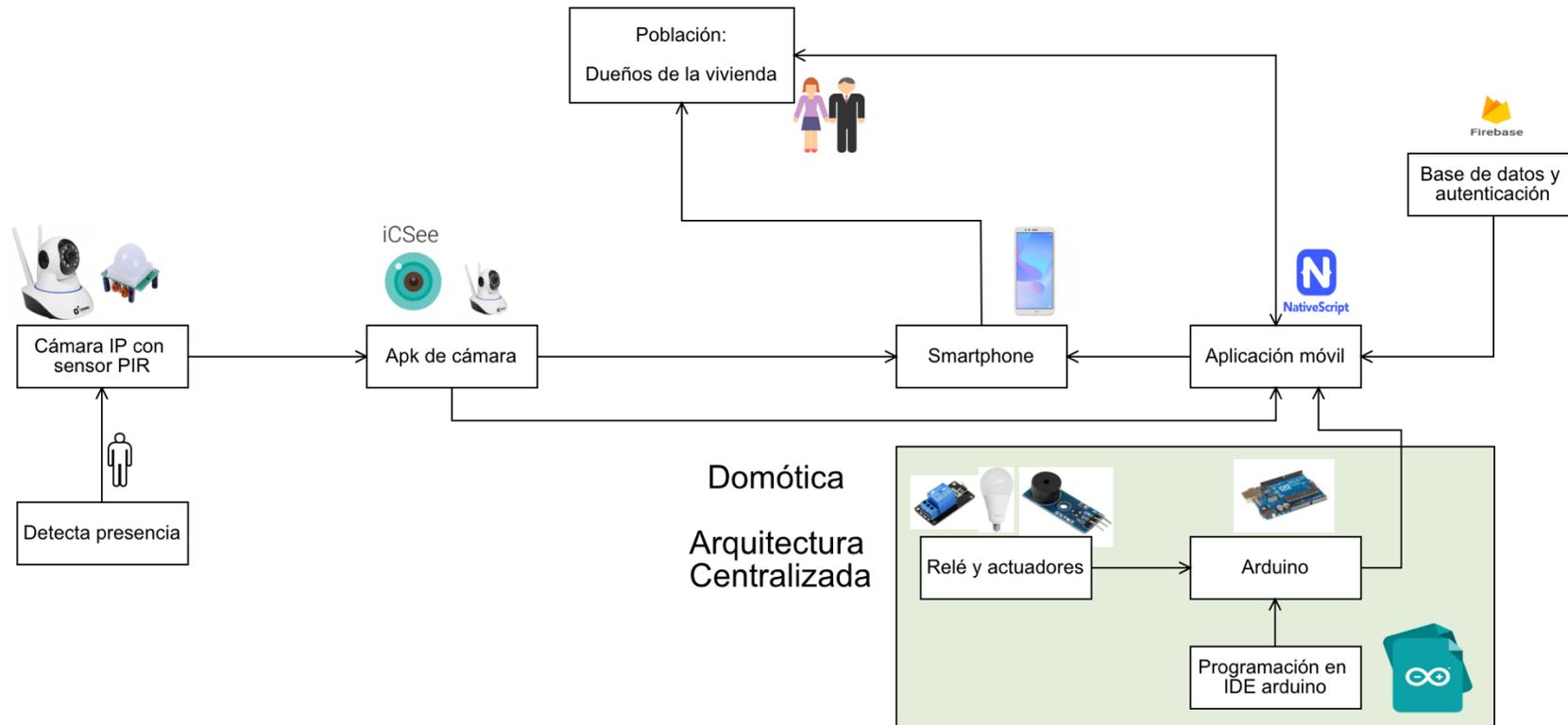


Figura 26. Arquitectura de solución de la investigación.

Fuente: Elaboración propia, 2018.

En la *figura 27* representa a la arquitectura de solución de la investigación, se visualiza que la solución a la vigilancia remota, da inicio cuando la cámara IP integrada con un sensor PIR, detecta la presencia de un cuerpo en un radio de 6m, luego la aplicación de la cámara instalada en el Smartphone, notifica al usuario la detección de movimiento, una vez recepcionado la notificación, el usuario ingresa a la aplicación móvil desarrollada con NativeScript, dónde se accederá con su usuario y contraseña previamente registrado, ingresando a la aplicación el usuario podrá controlar la iluminación y la alarma, también podrá visualizar el video en vivo de la cámara, mediante la aplicación iCSee (apk de la cámara IP), de esta manera el usuario estará vigilando su vivienda de manera remota.

3.3.4. Metodología XP

En esta parte el investigador por ser XP una metodología ágil, hace uso de las actividades necesarias para cada fase de la metodología, esta personalización se hizo de acuerdo a las necesidades para el desarrollo de la solución.

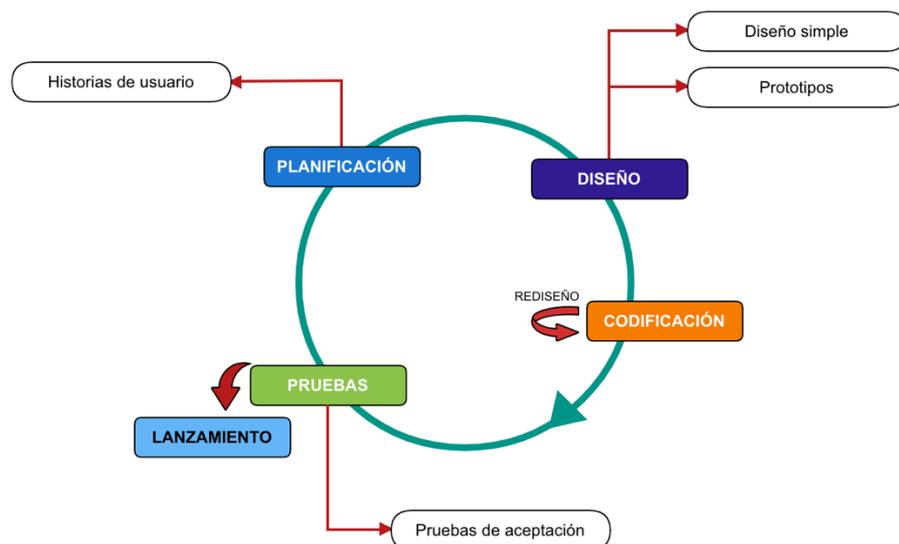


Figura 27. Metodología XP reducida.

Fuente: Propia, 2018.n

3.3.4.1. Planificación

En esta fase se define las historias de usuario con los clientes, estos constan de 2 o 3 líneas escritas por el cliente en un lenguaje que no es técnico, estas mismas se utiliza para la fase de las pruebas, para corroborar si el sistema cumple con lo que especifica la historia de usuario.

3.3.4.2. Diseño

En esta fase se define un diseño simple y sencillo del sistema, como por ejemplo la arquitectura tecnológica que se usará para el desarrollo, esto debe ser entendible y fácil de implementar para que demande menos tiempo y esfuerzo para desarrollar. Después de esto se diseña un prototipo del sistema, tanto la parte del hardware y software.

3.3.4.3. Codificación

En esta fase a la hora de la codificación de una historia de usuario, que es un requerimiento, la presencia del cliente es muy importante. Antes del desarrollo de cada historia de usuario el cliente debe especificar detalladamente que hará y que tendrá que contemplar. Para que a la hora de realizar las pruebas se compruebe si cumple la funcionalidad especificada.

3.3.4.4. Pruebas

En esta fase se hace la prueba de funcionalidad por cada historia de usuario, estas pruebas de funcionalidad se valida con la presencia de los clientes quienes definen si la prueba fue satisfactoria o deficiente.

CAPÍTULO IV

DESARROLLO DEL SISTEMA DOMÓTICO

4.1. Fase de planificación

En esta etapa el investigador desarrolló un formato para el levantamiento de las historias de usuario, lo cual en detalle es explicado en el Anexo 4, esto permitió al investigador recopilar los requerimientos funcionales para el sistema domótico y posteriormente la asignación de las tareas.

4.1.1. Historias de usuario

Mediante la técnica de entrevista realizada a los dueños de la vivienda, el investigador recopiló las siguientes historias de usuario:

- HDU-01: En mi domicilio quiero contar con cámaras de seguridad, sobre todo en los lugares que tienen puerta y ventana.
- HDU-02: Yo no cuento con ningún tipo de alarma, a veces si es que quieren ingresar por la ventana o por la puerta otras personas, entonces yo quisiera que mi Smartphone me avise.
- HDU-03: En las noches cuando estoy durmiendo si alguien intenta ingresar a mi casa quisiera que suene mi alarma, o cuando el perro ladre quisiera encender las luces y ver quien esta ingresando a mi propiedad.
- HDU-04: Controlar las luminarias y la alarma del hogar.
- HDU-05: Quisiera un sistema solo para las personas que vivimos en mi casa.
- HDU-06: Quisiera desde mi Smartphone poder ver lo que ocurre en mi casa desde cualquier lugar.
- HDU-07: Quisiera desde mi Smartphone controlar el foco y la alarma que se van a poner en mi casa, quiero hacer esto hasta cuando esté de viaje y lejos de mi casa.

4.1.1.1. Descripción de Historias de usuario para la aplicación móvil

Tabla 7.
Historia de usuario número 1

Historia de Usuario	
Número: HDU-01 Usuario: dueños de la vivienda	
Nombre Historia: En mi domicilio quiero contar con cámaras de seguridad, sobre todo en los lugares que tienen puerta y ventana.	
Prioridad: Alta	Puntos Estimados: 2
Riesgo: Bajo	Iteración Asignada: 1
Programador Responsable: Mireya Chambi	
Descripción: Instalación y configuración de cámaras de seguridad en lugares que tengan puerta y ventana de la vivienda.	

Fuente: Elaboración propia, 2019.

Tabla 8.
Historia de usuario número 2

Historia de Usuario	
Número: HDU-02 Usuario: Dueños de la vivienda	
Nombre Historia: Yo no cuento con ningún tipo de alarma, a veces si es que quieren ingresar por la ventana o por la puerta otras personas, entonces yo quisiera que mi Smartphone me avise.	
Prioridad: Alta	Puntos Estimados: 1
Riesgo: Bajo	Iteración Asignada: 1
Programador Responsable: Mireya Chambi	
Descripción: Instalación de alarmas en lugares de acceso de personas a la vivienda, como puerta principal y secundarias.	

Fuente: Elaboración propia, 2019.

Tabla 9.
Historia de usuario número 3

Historia de Usuario	
Número: HDU-03 Usuario: Dueños de la vivienda	
Nombre Historia: En las noches cuando estoy durmiendo si alguien intenta ingresar a mi casa quisiera que suene mi alarma, o cuando el perro ladre quisiera encender las luces y ver quien esta ingresando a mi propiedad.	
Prioridad: Media	Puntos Estimados: 1
Riesgo: Bajo	Iteración Asignada: 1
Programador Responsable: Mireya Chambi	
Descripción: Por el aviso de presencia de personas por las noches en la vivienda, se debe encender y apagar los focos externos e internos.	

Fuente: Elaboración propia, 2019.

Tabla 10.
Historia de usuario número 4

Historia de Usuario	
Número: HDU-04 Usuario: Dueños de la vivienda	
Nombre Historia: Controlar las luminarias y alarma del hogar.	
Prioridad: Alta	Puntos Estimados: 2
Riesgo en Desarrollo: Medio	Iteración Asignada: 1
Programador Responsable: Mireya Chambi	
Descripción: Mediante un sistema los dueños de la vivienda deben controlar las luminarias y alarmas.	

Fuente: Elaboración propia, 2019.

Tabla 11.
Historia de usuario número 5

Historia de Usuario	
Número: HDU-05 Usuario: Dueños de la vivienda	
Nombre Historia: Quisiera un sistema solo para las personas que vivimos en mi casa.	
Prioridad: Alta	Puntos Estimados: 3
Riesgo en Desarrollo: Alto	Iteración Asignada: 1
Programador Responsable: Mireya Chambi	
Descripción: Realización de un sistema donde puedan acceder solo los dueños de la vivienda, solo usuarios registrados.	
Fuente: Elaboración propia, 2019.	

Tabla 12.
Historia de usuario número 6

Historia de Usuario	
Número: HDU-06 Usuario: Dueños de la vivienda	
Nombre Historia: Quisiera desde mi Smartphone poder ver lo que ocurre en mi casa desde cualquier lugar.	
Prioridad: Alta	Puntos Estimados: 2
Riesgo en Desarrollo: Medio	Iteración Asignada: 1
Programador Responsable: Mireya Chambi	
Descripción: Implementar la visualización de la cámara en la aplicación móvil.	
Observaciones: Smartphone con acceso a internet y previa configuración de la cámara.	
Fuente: Elaboración propia, 2019.	

Tabla 13.
Historia de usuario número 7

Historia de Usuario	
Número: HDU-07 Usuario: Dueños de la vivienda	
Nombre Historia: Quisiera desde mi Smartphone controlar los focos y la alarma que se van a poner en mi casa, quiero hacer esto hasta cuando este de viaje y lejos de mi casa.	
Prioridad: Alta	Puntos Estimados: 2
Riesgo en Desarrollo: Medio	Iteración Asignada: 1
Programador Responsable: Mireya Chambi	
Descripción: Control de la luminaria y alarma instaladas en la vivienda desde cualquier lugar.	
Observaciones: Smartphone con acceso a internet.	
Fuente: Elaboración propia, 2019.	

4.1.1.2. *Requerimientos para el sistema domótico*

Tabla 14.
Requerimientos funcionales

CÓDIGO	REQUERIMIENTOS FUNCIONALES	ASIGNADO
<i>REQF-001</i>	Validación de usuarios.	Mireya Chambi
<i>REQF-002</i>	Permitir la visualización del video emitido por la cámara en tiempo real.	Mireya Chambi
<i>REQF-003</i>	Permitir la manipulación de los dispositivos integrados al sistema domótico.	Mireya Chambi

Fuente: Elaboración propia, 2019.

Tabla 15.
Requerimientos no funcionales

CÓDIGO	REQUERIMIENTOS NO FUNCIONALES	ASIGNADO
REQNF-001	Vistas amigables	Mireya Chambi
REQNF-002	Autenticación por usuario/email, contraseña y cuentas de google	Mireya Chambi

Fuente: Elaboración propia, 2019.

4.1.1.3. *Tareas asignadas para el sistema domótico*

Tabla 16.
Tarea 1, HDU-01

Tarea	
Número: 1	Numero de Historia: HDU-01
Nombre: Instalación y configuración de cámaras de seguridad	
Tipo de tarea: Configuración	Puntos Estimados: 2
Fecha Inicio: 08 de enero de 2019	Fecha fin: 18 de enero de 2019
Programador Responsable: Mireya Chambi	
Descripción: Se hace la adquisición de cámaras de seguridad, después se realiza la instalación en la vivienda y finalmente la configuración.	

Fuente: Elaboración propia, 2019.

Tabla 17.
Tarea 2, HDU-02 y HDU-03

Tarea	
Número: 2	Numero de Historia: HDU-02, HDU-03
Nombre: Instalación de alarmas	
Tipo de tarea: Instalación	Puntos Estimados: 1
Fecha Inicio: 08 de enero de 2019	Fecha fin: 23 de enero de 2019
Programador Responsable: Mireya Chambi	
Descripción: Se hace la adquisición de alarmas, después se realiza la instalación en puntos claves de la vivienda.	

Fuente: Elaboración propia, 2019.

Tabla 18.
Tarea 3, HDU-03, HDU-04 y HDU-07

Tarea	
Número: 3	Numero de Historia: HDU-03, HDU-04, HDU-07
Nombre: Control de luminaria y alarmas remotamente	
Tipo de tarea: Desarrollo	Puntos Estimados: 2
Fecha Inicio: 05 de febrero de 2019	Fecha fin: 28 de febrero de 2019
Programador Responsable: Mireya Chambi	
Descripción: Vista donde se visualicen los botones para el control de los dispositivos electrónicos.	

Fuente: Elaboración propia, 2019.

Tabla 19.
Tarea 4, HDU-05

Tarea	
Número: 4	Numero de Historia: HDU-05
Nombre: Acceso a aplicación móvil	
Tipo de tarea: Desarrollo	Puntos Estimados: 2
Fecha Inicio: 04 de marzo 2019	Fecha fin: 13 de marzo de 2019
Programador Responsable: Mireya Chambi	
Descripción: Vista de acceso al aplicativo móvil, autenticación mediante usuario/email, contraseña y cuentas de google.	

Fuente: Elaboración propia, 2019.

Tabla 20.
Tarea 5, HDU-06

Tarea	
Número: 5	Numero de Historia: HDU-06
Nombre: Visualización de la vivienda en tiempo real	
Tipo de tarea: Desarrollo	Puntos Estimados: 2
Fecha Inicio: 19 de marzo de 2019	Fecha fin: 19 de marzo 2019
Programador Responsable: Mireya Chambi	
Descripción: Vista que muestra el video en tiempo real emitido por la cámara instalada en la vivienda, conexión con la aplicación de la cámara.	

Fuente: Elaboración propia, 2019.

Tabla 21.
Tarea 6, HDU-01 hasta HDU-07

Tarea	
Número: 6	Numero de Historia: HDU-01 hasta HDU-07
Nombre: Integración del sistema domótico	
Tipo de tarea: Desarrollo	Puntos Estimados: 2
Fecha Inicio: 21 de marzo de 2019	Fecha fin: 26 de marzo 2019
Programador Responsable: Mireya Chambi	
Descripción: En esta tarea se hace la integración tanto del hardware (Arduino) y software (aplicación móvil) por los cuales está compuesto el sistema domótico.	
Fuente: Elaboración propia, 2019.	

4.2. Fase de diseño

4.2.1. Diseño simple - arquitectura tecnológica

En la *figura 29*, se muestra la arquitectura tecnología que se usó para el desarrollo del sistema domótico, por la parte del cliente está el aplicativo móvil desarrollado con NativeScript-Vue, un framework basado en componentes que facilita el desarrollo en un tiempo reducido y la reutilización de código, en la parte del servidor se contempla la parte del hardware del sistema domótico, en primer lugar se levantó el servicio desarrollado con Arduino, para el control de la iluminaria y alarma y otro de los servicios fue la aplicación de la cámara, finalmente para la parte de control de acceso a la aplicación móvil se usó la base de datos no relacional Firebase, que brinda distintos tipos de autenticación o accesos.

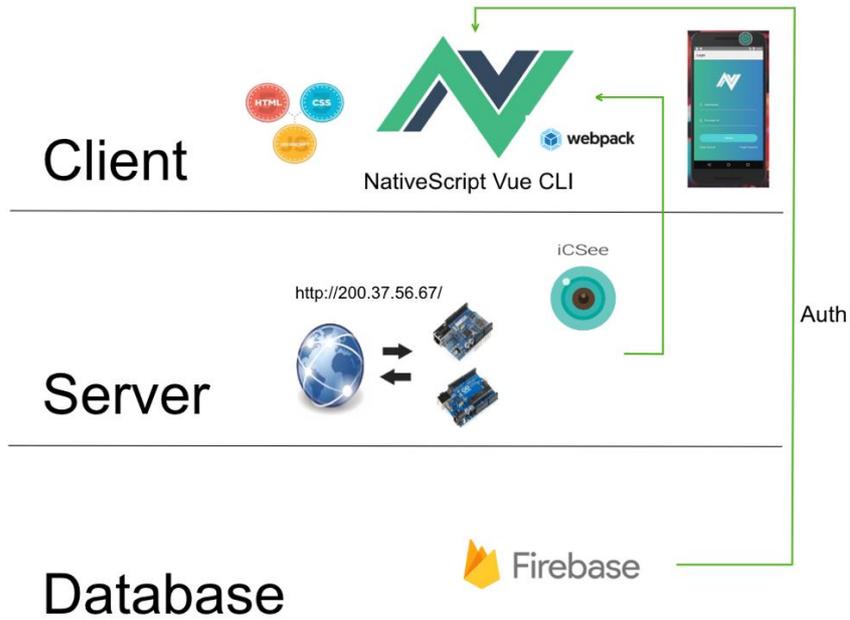


Figura 28. Arquitectura Tecnológica del Sistema Domótico.

Fuente: Elaboración propia, 2019.

4.2.2. Prototipo del esquema físico de las conexiones en Arduino

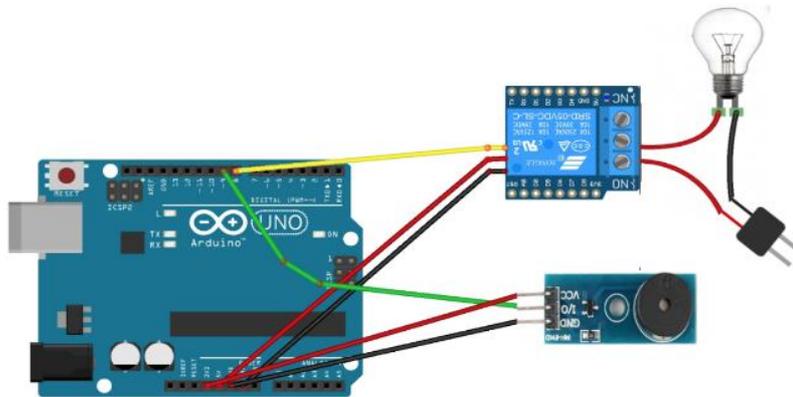


Figura 29. Diseño de esquema físico de conexiones en Arduino.

Fuente: Elaboración propia, 2019.

4.2.3. Prototipo del servicio web Arduino

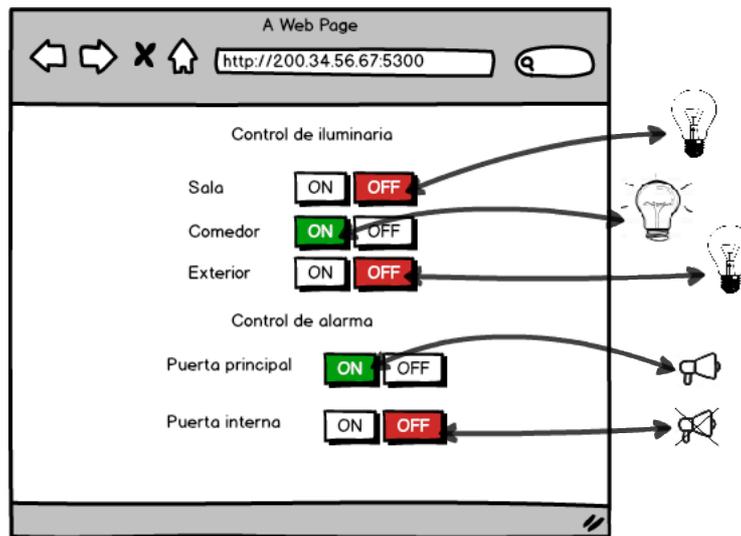


Figura 30. Vista del servicio web Arduino.

Fuente: Elaboración propia, 2019.

4.2.4. Prototipo de la aplicación móvil

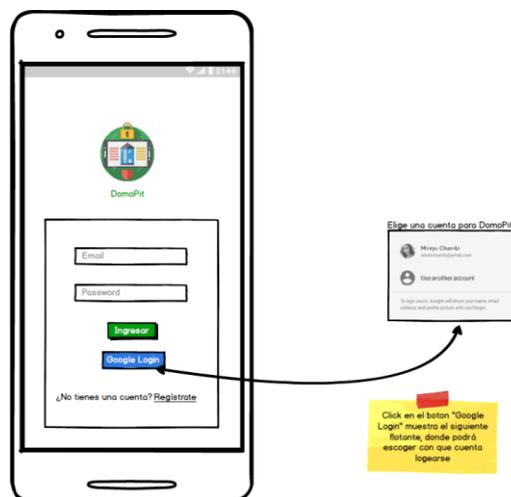


Figura 31. Vista de Acceso a la aplicación.

Fuente: Elaboración propia, 2019.

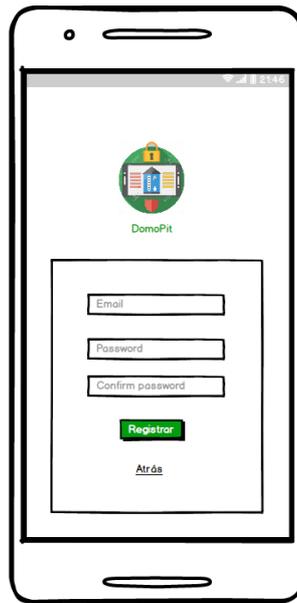


Figura 32. Vista para registro de usuario.

Fuente: Elaboración propia, 2019.

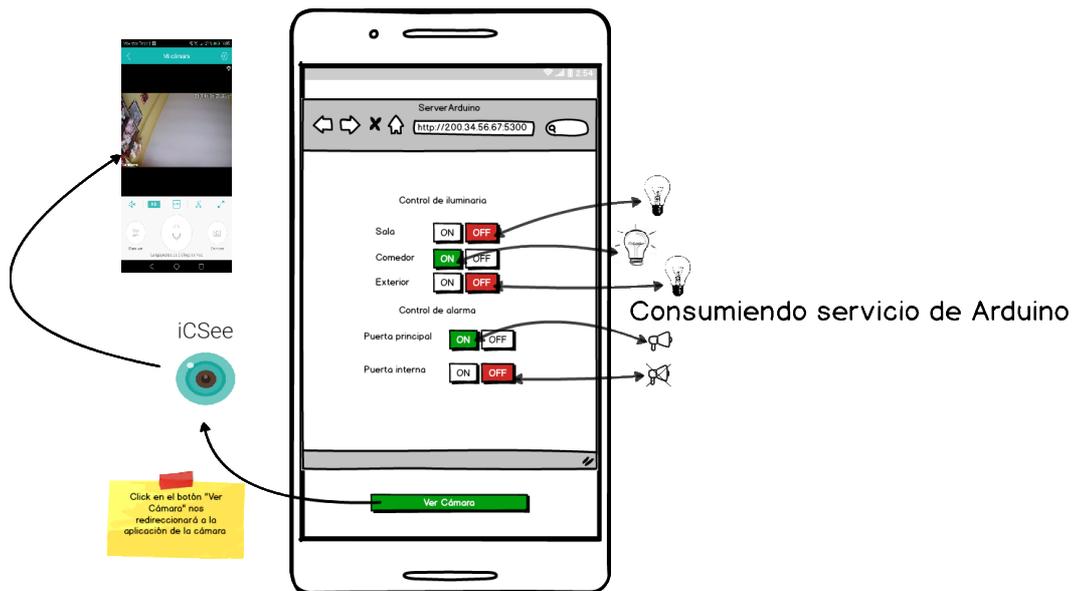


Figura 33. Vista principal de la aplicación.

Fuente: Elaboración propia, 2019.

4.3. Fase de codificación

4.3.1. Desarrollo servidor web - Arduino

Para la configuración del servidor web en Arduino se requirió incluir la librería *SPI.h* que permitió la comunicación entre el Arduino y el Shield Ethernet. También se requirió incluir la librería *Ethernet.h* que es la que conectó el Arduino a internet. Después se declaró la dirección MAC del Arduino, el IP, el puerto y los pines de salida de los dispositivos electrónicos.

```
#include <SPI.h> //Aquí incluimos la librería SPI
#include <Ethernet.h> //Aquí incluimos la librería Ethernet
Byte mac[]={0xDE,0xAD,0xBE,0xEF,0xFE,0xED}; //Declaración de la dirección MAC
IPAddress ip (192, 168, 1, 70); //Declaramos la IP
EthernetServer servidor (5300); //Declaración del puerto 5300
int PIN_LED=8; //Aquí establecemos la variable PIN_LED como un valor entero
int PIN_BUZZER=9; //Aquí establecemos la variable PIN_BUZZER como un valor entero
```

En este sector del código se muestra el control de iluminación y alarma, en un archivo .ino y una estructura HTML.

```
cliente.println("<h1>Control de Iluminacion</h1>");
cliente.print("<br><br>");
cliente.print("Estado del FOCO: ");
cliente.print(state);
cliente.print("<br><br><br><br>");
cliente.println("<input type=submit value=ON style=width:100px;height:38px onClick=location.href='./?LED=T'\>");
cliente.println("<input type=submit value=OFF style=width:100px;height:38px onClick=location.href='./?LED=F'\>");
cliente.println("</center>");

cliente.println("<center>");
cliente.println("<h1>Control de Alarma</h1>");
cliente.print("<br><br>");
cliente.print("Estado del ALARMA: ");
cliente.print(state);
cliente.print("<br><br><br><br>");
cliente.println("<input type=submit value=ON style=width:100px;height:38px
onClick=location.href='./?BUZZER=T'\>");
cliente.println("<input type=submit value=OFF style=width:100px;height:38px
onClick=location.href='./?BUZZER=F'\>");
cliente.println("</center>");
```

Para ver todo el código de Arduino, revise el *Anexo B*.

4.3.2. Desarrollo móvil – NativeScript-Vue

Para el desarrollo de la aplicación móvil de forma ágil se usó el framework NativeScript-Vue, y con la utilización de la base de datos Firebase se permitió acceder al aplicativo mediante dos métodos, por email/contraseña y por cuentas existentes de google. Tal como se muestra en el siguiente código. Componente LoginPage.vue, ver en *Anexo C*.

Correo electrónico y contraseña

```
import firebase from "nativescript-plugin-firebase";
const userService = {
  async register(user) {
    return await firebase.createUser({
      email: user.email,
      password: user.password
    });
  },
  async login(user) {
    return await firebase.login({
      type: firebase.LoginType.PASSWORD,
      passwordOptions: {
        email: user.email,
        password: user.password
      }
    });
  },
};
```

Cuenta existente de Google

```
async loginGoogle(user) {
  await firebase
    .login({
      type: firebase.LoginType.GOOGLE
    })
    .then(result => {
      return Promise.resolve(JSON.stringify(result));
    })
    .catch(error => {
      console.error(error);
      return Promise.reject(error);
    });
},
async resetPassword(email) {
  return await firebase.resetPassword({
    email: email
  });
}
};
```

Se hizo la integración del servidor de Arduino en un componente `<WebView/>` y para poder controlar la cámara se usó un componente `<Button/>`. Componente `Player.vue`, ver en *Anexo C*.

```
<template>
  <Page class="page">
    <ActionBar title="Video" class="action-bar" />
    <ScrollView>
      <StackLayout class="home-panel">
        <WebView height="1000px" src="http://192.168.1.70:5300"/>
        <Button text="Ver Cámara" @tap="submit" class="buttoncam" />
      </StackLayout>
    </ScrollView>
  </Page>
</template>
```

Para acceder a la aplicación de la cámara mediante nuestra aplicación se usó el plugin “*nativescript-open-app*”, brindando el ID único “*com.xm.csee*”. Tal como se muestra en las siguientes líneas de código:

```
<script>
var openApp = require("nativescript-open-app").
openApp;

export default {
  methods: {
    submit() {
      var installed = openApp("com.xm.csee",false);
      console.log("Is it installed? " + installed);
    }
  },
  data () {
    return {
    };
  },
}
</script>
```

4.4. Fase de pruebas

4.4.1. Pruebas de configuración del hardware

Tabla 22.

Caso de prueba - Correcto funcionamiento de los dispositivos electrónicos integrados con la placa Arduino y Shield Ethernet

Caso de Prueba	
Número de Caso de Prueba:1	Número Historia de Usuario: HDU-01
Nombre Caso de Prueba: Correcto funcionamiento de los dispositivos electrónicos integrados con la placa Arduino y Shield Ethernet	
Descripción: prueba para el correcto funcionamiento de los dispositivos electrónicos integrados con la placa Arduino y Shield Ethernet.	
Condiciones de ejecución: se debe tener las conexiones físicas realizadas, entre dispositivos electrónicos, placa Arduino y Shield ethernet.	
Entradas: Una vez compilado el programa en la placa Arduino.	
Resultado esperado: Se logra establecer las conexiones y tener la integración de toda la parte de hardware del sistema domótico.	
Evaluación: prueba satisfactoria	

Fuente: Elaboración propia, 2019.



Figura 34. Integración de Arduino, Shield Ethernet, relé, sensor de alarma y bombilla.

Fuente: Elaboración propia, 2019.

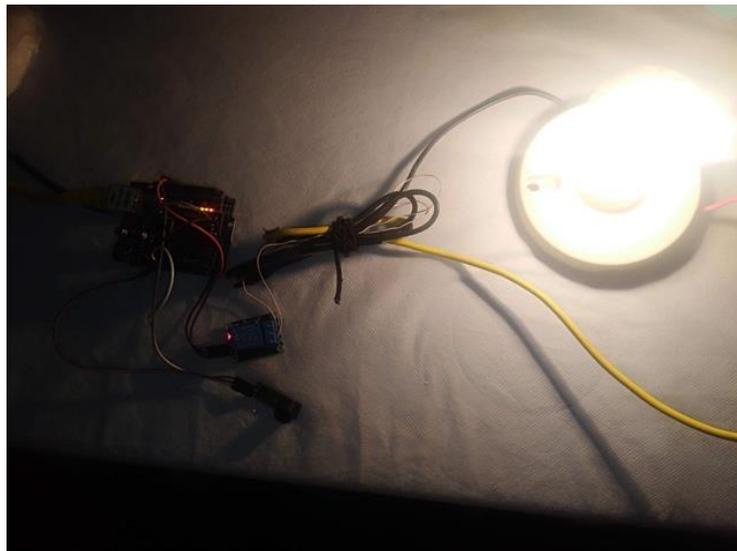


Figura 35. Correcto funcionamiento de la integración.

Fuente: Elaboración propia, 2019.

4.4.2. Pruebas de aceptación del aplicativo móvil

Para la realización de las pruebas de funcionalidad de la aplicación móvil DomoPit, se definió las características mínimas con las que debe contar el Smartphone. Ver *Anexo A*.

Tabla 23.
Caso de prueba - Validar usuarios

Caso de Prueba	
Número de Caso de Prueba:2	Número Historia de Usuario: REQF-001, REQNF-001, REQNF-002
Nombre Caso de Prueba: Validar usuarios	
Descripción: prueba para la funcionalidad validar usuarios	
Condiciones de ejecución: el usuario tiene que estar previamente registrado y si no debe registrarse	
Entradas: el usuario accede al aplicativo mediante un email y contraseña o con una cuenta de existente de google	
Resultado esperado: muestra la pantalla principal de la aplicación	
Evaluación: prueba satisfactoria	
Fuente: Elaboración propia, 2019.	

Se desarrolló dos métodos de acceso al sistema domótico, el número 1 es usando un correo electrónico y contraseña previamente registrado, el número 2 es usando una cuenta de Google, como se muestra en la *figura 38*, los dos métodos de autenticación de usuarios son proveídos por Firebase, como se puede apreciar en la *figura 37*. Las pruebas de aceptación se pueden ver en la parte de la metodología (fase de pruebas).

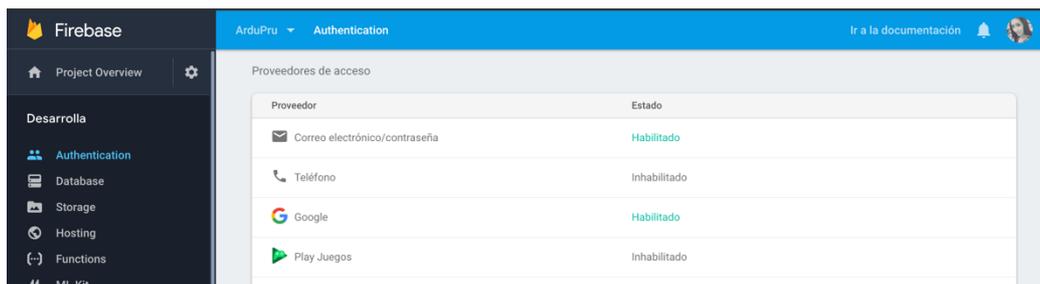


Figura 36. Habilitando los servicios de autenticación por correo electrónico/contraseña y google.

Fuente: Elaboración propia, 2019.

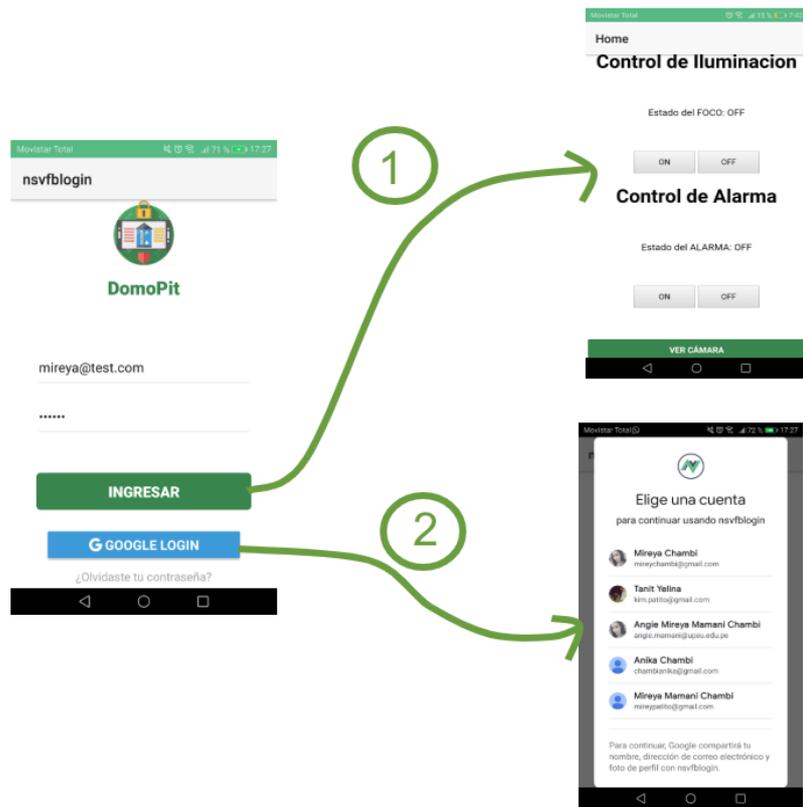


Figura 37. Métodos de acceso al aplicativo del sistema domótico DomoPit.

Fuente: Elaboración propia, 2019.

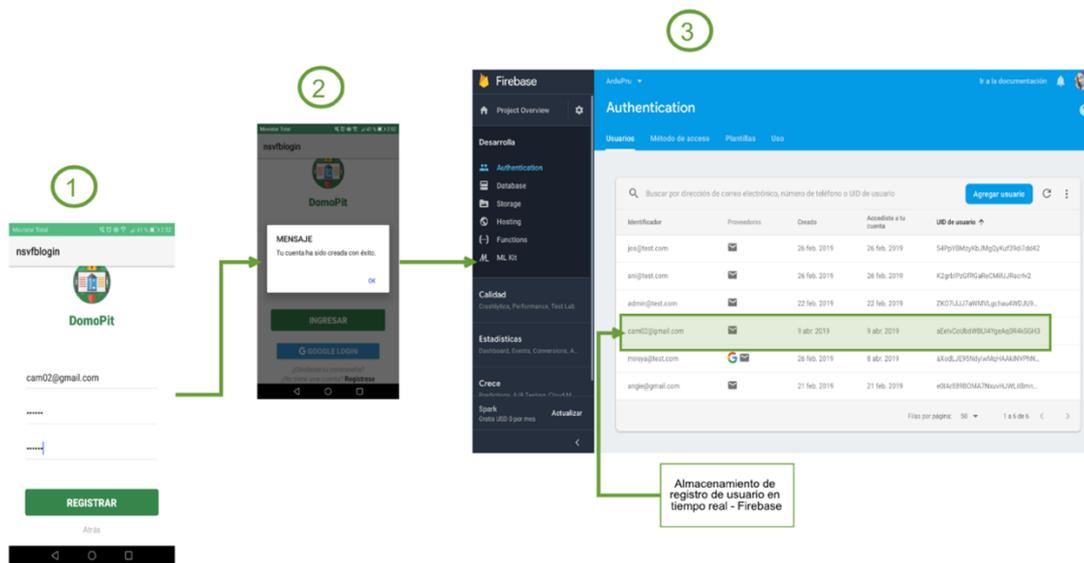


Figura 38. Registro de nuevo usuario, almacenamiento en tiempo real Firebase.

Fuente: Elaboración propia, 2019.

Tabla 24.
Caso de prueba - Visualizar video en tiempo real

Caso de Prueba	
Número de Caso de Prueba:3	Número Historia de Usuario: REQF-002, REQNF-002
Nombre Caso de Prueba: Visualizar video en tiempo real	
Descripción: prueba para la funcionalidad de visualización de video en tiempo real	
Condiciones de ejecución: el usuario tiene que ingresar a la aplicación	
Entradas: una vez que el usuario ingrese a la aplicación presiona el botón ver cámara	
Resultado esperado: muestra el video en tiempo real emitido por la cámara	
Evaluación: prueba satisfactoria	

Fuente: Elaboración propia, 2019.



Figura 39. Funcionalidad de la vista del video en tiempo real.

Fuente: Elaboración propia, 2019.

Tabla 25.

Caso de prueba - Encender y apagar los dispositivos integrados al sistema domótico

Caso de Prueba	
Número de Caso de Prueba: 4	Número Historia de Usuario: REQF-003, REQNF-002
Nombre Caso de Prueba: Encender y apagar los dispositivos integrados al sistema domótico	
Descripción: prueba para la funcionalidad del encendido y apagado de los dispositivos integrados al sistema domótico	
Condiciones de ejecución: el usuario tiene que ingresar a la aplicación	
Entradas: una vez que el usuario ingrese a la aplicación presiona los botones de apagado y encendido de los dispositivos integrados al sistema domótico	
Resultado esperado: los dispositivos se encienden y apagan correctamente	
Evaluación: prueba satisfactoria	

Fuente: Elaboración propia, 2019.

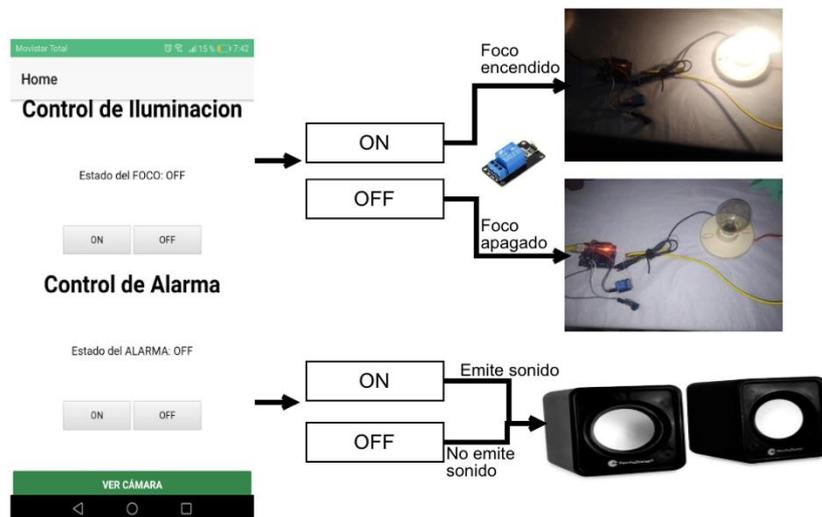


Figura 40. Funcionalidad del control para la iluminación y alarma.

Fuente: Elaboración propia, 2019.

CAPÍTULO V

RESULTADOS Y DISCUSIÓN

5.1. Resultado del objetivo específico 1

En esta parte se levantó las historias de usuario mediante la técnica de la entrevista, a través de esto se logró concretar los requerimientos funcionales y no funcionales del sistema domótico, ver en las *tablas 14 y 15*, posteriormente se logró diseñar los prototipos para la parte del hardware y software del sistema domótico descritos en el capítulo IV.

5.1.1. Discusión

Uno de los propósitos de la presente investigación fue analizar y diseñar los requerimientos levantados para el sistema domótico, en investigaciones anteriores Martínez (2015) levantó sus requerimientos mediante la declaración de variables potenciales que necesita el sistema domótico, el no diseñó un prototipo sino desarrolló la solución de forma directa, a la vez Paiz (2015) hizo el análisis y diseño de sus requerimientos mediante la definición de parámetros y diagramas de bloques buscando la mejor solución a su problemática. En esta investigación se usó la entrevista para la definición de los requerimientos, que inicialmente fueron las historias de usuario, y se diseñó un prototipo para guía en la etapa de la codificación o desarrollo del sistema domótico.

5.2. Resultado del objetivo específico 2

Como resultado se obtuvo la integración del hardware y software. En lo que concierne a hardware se levantó un servidor en Arduino para el control de la iluminación y las alarmas, y en software se desarrolló una aplicación móvil llamada DomoPit con accesos de Firebase, asimismo la visualización del video emitido en tiempo real mediante la aplicación iCSee propia de la cámara. Estas herramientas garantizaron la integración de un sistema domótico estable, que cumplió con las expectativas de los dueños de la vivienda. En la *figura 42* se muestra el esquema general de la integración de las tecnologías de automatización de viviendas.

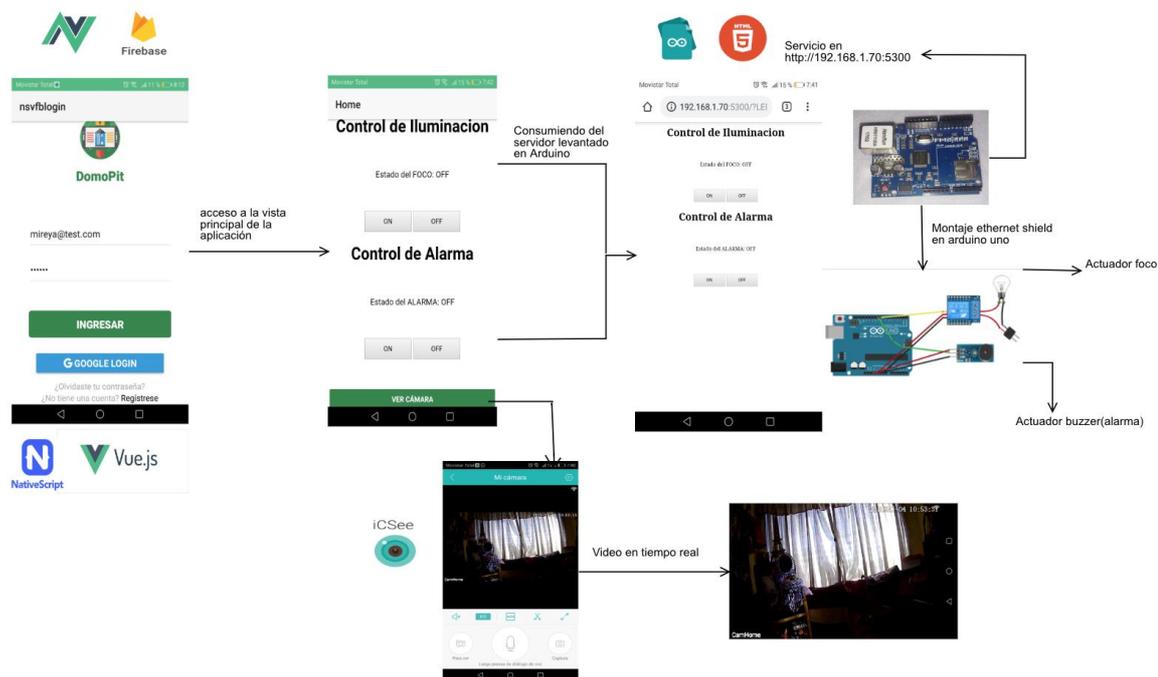


Figura 41. Esquema general de la integración de las tecnologías de automatización de viviendas.

Fuente: Elaboración propia, 2019.

5.2.1. Discusión

Otro de los propósitos de la presente investigación fue integrar las tecnologías de automatización en las viviendas, en investigaciones anteriores como la de Martínez (2015) quién integró un sistema domótico haciendo uso de un microcontrolador Raspberry Pi, sensor de movimiento, sensor de consumo eléctrico, sensor de control de iluminación los cuales son controlados mediante un aplicativo móvil desarrollado en JavaScript, también Mejía (2016) integró un sistema domótico con placa Arduino, dispositivos electrónicos los cuales son controlados mediante un teléfono móvil con sistema operativo Android esta conexión usando la tecnología Bluetooth y Guerra (2014) integró un sistema domótico haciendo uso de Raspberry Pi, sistemas de vigilancia IP y telefonía IP, sensor PIR, relés estos controlador por un aplicación web, desarrollada en HTML y JQuery Mobile. Tomando en cuenta los antecedentes mencionados, en esta investigación se realizó la integración de las tecnologías de automatización de viviendas con el uso de un shield Ethernet montado en una placa Arduino para el levantamiento de un servidor y la salida a internet mediante un

protocolo HTTP, una cámara IP con sensor de movimiento PIR para detectar presencia, un aplicativo móvil desarrollado en NativeScript-Vue como puente para el control de la iluminación, alarmas y video en tiempo real.

5.3. Resultado del objetivo específico 3

Se codificó los requerimientos del sistema domótico basado en las historias de usuario recopiladas, ver *anexo B* y *C*. Para la realización de las pruebas se consideró un indicador importante que son las notificaciones enviadas al Smartphone del dueño de la vivienda, notificando la presencia de personas en su propiedad. En cuanto a las funcionalidades del sistema, se obtuvo un resultado satisfactorio, como se puede ver en la fase de pruebas descrito en el capítulo IV, dando como resultado una versión de producción del aplicativo móvil DomoPit.

5.3.1. Discusión

Otro de los propósitos de la presente investigación fue codificar y realizar pruebas del sistema domótico Domopit, Aviles & Cobeña (2015) realizó sus pruebas de aceptación mediante la experiencia de los usuarios quienes le indicaron si su sistema cumplía con sus requerimientos, no desarrolló ningún formato, a la vez Núñez (2017) quién realizó las pruebas de sus prototipado mediante una tabla donde almacena los campos de sensor, estímulo, trama y esto es reflejado en la aplicación de PC. En ambos casos no se desarrolló un aplicativo móvil sino hicieron uso de otras tecnologías como GSM, WiFi, en esta investigación se codificó un aplicativo móvil y un servidor web que ambos al ser integrados conformaron el sistema domótico, se usó conexiones ethernet, y para la realización de las pruebas se usó un formato donde el dueño de la vivienda dio el visto bueno a las funcionalidades del sistema domótico.

CAPÍTULO VI

CONCLUSIONES Y RECOMENDACIONES

6.1. Conclusiones

Conforme a los resultados expuestos en esta investigación de desarrollo de un sistema domótico controlado mediante dispositivos móviles para la vigilancia remota de viviendas familiares, se concluyó lo siguiente:

Con respecto al objetivo general, se desarrolló un sistema domótico controlado mediante dispositivos móviles para vigilancia remota de viviendas, validado con las pruebas de aceptación de parte del dueño de la vivienda tanto en hardware como en software.

Con respecto al primer objetivo específico se analizó y diseñó los requerimientos levantados para el sistema domótico, mediante las historias de usuario que vienen siendo los requerimientos, y los prototipos tanto para la parte del hardware y software del sistema domótico DomoPit.

Con respecto al segundo objetivo específico se concluyó que, si es factible integrar hardware con software tratándose de tecnologías de automatización en las viviendas, mediante el levantamiento de un servicio en Arduino con Shield ethernet, conectado al router que está en la vivienda, para permitir el control de la iluminación/alarma y una aplicación móvil donde se integra este servicio juntamente con la visualización del video en vivo emitido por la aplicación iCSee, propio de la cámara.

Con respecto al tercer objetivo específico se codificó y realizó pruebas al sistema domótico DomoPit, y se concluyó que, si se puede hacer uso de las nuevas herramientas de desarrollo móvil como NativeScript-Vue y Firebase, en sistemas domóticos, permitiendo realizar la autenticación del usuario, integración del servidor web de Arduino y la aplicación iCSee de la cámara IP. Luego la realización de las pruebas de funcionalidad con los dueños de la vivienda, las cuales tuvieron un resultado satisfactorio.

6.2. Recomendaciones

Para investigaciones futuras se propone hacer el uso de No-IP para obtener un dominio público estático, donde esté almacenado el servidor web de Arduino, con esto por mas que la IP pública del router sea dinámica, el domino público creado no cambiará.

En caso de corte de fluido eléctrico se recomienda la instalación de alarmas de emergencia y la instalación de las cámaras con UPS.

La población peruana en general debería optar por integrar estos tipos de sistema en sus viviendas, para no tener preocupación al momento de dejar sus viviendas solas, sin la presencia de una persona, o en el caso de un viaje familiar, se vayan con la confianza de poder vigilar su vivienda desde cualquier lugar donde tengan conexión Internet.

Se necesita mas investigaciones relacionadas y enfocadas a la seguridad ciudadana, por el incremento de los actos delictivos en estos últimos años, para así salvaguardar la integridad de la población.

REFERENCIAS

- IEP, I. for E. and P. (2018). 2018 GLOBAL PEACE INDEX. Retrieved from <http://visionofhumanity.org/app/uploads/2018/06/Global-Peace-Index-2018-2.pdf>
- 4BP. (n.d.). RISC. Retrieved from http://4.bp.blogspot.com/-EcJapT9Fkq4/TdJ3tvISWAI/AAAAAAAAAIs/Ct_8G9tXrcI/s1600/RISC+2.JPG
- A. Dean, R. (n.d.). la investigación tecnológica en las ciencias de la ingeniería y la innovación tecnológica. Retrieved from <https://www.unrc.edu.ar/publicar/23/dossidos.html>
- Adaptix. (2019). Dirección IP Privada, Pública, Dinámica, Estática. Retrieved from <https://www.adaptixnetworks.com/direccion-ip/>
- Albusac, J. (2008). *Vigilancia Inteligente: Modelado de Entornos Reales e Interpretación de Conductas para la Seguridad*. Universidad de Castilla - La Mancha. Retrieved from <http://www.esi.uclm.es/www/jalbusac/doc/masteralbusac.pdf>
- Arduino, A. (2018). ¿Qué es Arduino? Retrieved from <http://arduino.cl/que-es-arduino/>
- Atiencia, K. (2017). *Aplicación móvil multiplataforma que brinda información de las unidades de salud públicas y privadas de la ciudad de Loja*. Universidad Nacional de Loja. Retrieved from <http://dspace.unl.edu.ec/jspui/handle/123456789/18564>
- Aviles, A., & Cobeña, K. (2015). *Diseño e implementación de un sistema de seguridad a través de cámaras, sensores y alarma, monitorizado y controlado teleméricamente para el centro de acogida "patio mi pana" perteneciente a la fundación proyecto salesiano. Facultad de Ingeniería Electronica*. Universidad Politécnica Salesiana Ecuador. Retrieved from <https://dspace.ups.edu.ec/bitstream/123456789/10401/1/UPS-GT001444.pdf>
- Bahit, E. (2011). *Artefactos en Scrum: claves para una organización diaria*.

- BBC Mundo, J. (2013). Por qué hay tan pocos crímenes en Islandia. Retrieved from https://www.bbc.com/mundo/noticias/2013/05/130526_islandia_crmenes_raros_armas_lav
- Borja, Y. (2013). Metodología Ágil de Desarrollo de Software – XP, 10. Retrieved from http://www.runayupay.org/publicaciones/2244_555_COD_18_290814203015.pdf
- Cevallos, K. (2015). Metodología de Desarrollo Ágil: XP y Scrum. Retrieved from <https://ingsoftwarekarlacevallos.wordpress.com/2015/05/08/metodologia-de-desarrollo-agil-xp-y-scrum/>
- Charaja, F. (2011). *El MAPIC en la Metodología de la Investigación*. (P. : s. E., Ed.) (2a ed.). Retrieved from http://biblioteca.unap.edu.pe/opac_css/index.php?lvl=notice_display&id=89794
- CircuitDigest. (2018). Relay types operation applications. Retrieved from <https://circuitdigest.com/article/relay-types-operation-applications>
- Cyron, P. (2016). Quick start with Firebase. Retrieved from <https://geeksdeck.com/quick-start-firebase/>
- DCervantes. (2014). Instalar Arduino IDE en Ubuntu. Retrieved from <https://scidle.com/es/instalar-arduino-ide-en-ubuntu/>
- Developers. (2018). Arquitectura de la plataforma. Retrieved from <https://developer.android.com/guide/platform/?hl=es-419>
- Domoprac, D. practica paso a paso. (2018). CONTROL DEL SISTEMA DE ILUMINACIÓN CON DOMÓTICA. Retrieved from <http://www.domoprac.com/domoteca/domoteca/integracion-domotica/control-del-sistema-de-iluminacion-con-domotica.html>
- Firebase. (2018). Firebase Realtime Database. Retrieved from <https://firebase.google.com/docs/database/>

- Flotante, P. (2018). Pruebas al módulo sensor pasivo infrarrojo PIR HC-SR501. Sensor de movimiento (body motion). Retrieved from <http://www.puntofotante.net/MODULO-SENSOR-PASIVO-INFRRARROJO-PIR-HC-SR501.htm>
- González, C., & Salcedo, O. (2017). Sistema de seguridad para locales comerciales mediante Raspberry Pi, cámara y sensor PIR. *Revista Virtual*, 175–193. Retrieved from <http://eds.a.ebscohost.com/eds/pdfviewer/pdfviewer?vid=2&sid=17134992-2213-4b3f-a001-09ad2a67c2d2%40sdc-v-sessmgr03>
- Guerra, F. (2014). *Diseño de un sistema de control domotico y video vigilancia supervisado por un telefono movil*. Pontificia Universidad Católica del Perú. Retrieved from <http://tesis.pucp.edu.pe/repositorio/handle/123456789/5375>
- Hernandez, M., & Parrales, G. (2011). *INSTALACIÓN Y SEGURIDAD DE LAS REDES INALAMBRICAS EN LA CLINICA “DR. RAFAEL HERNANDEZ TROYA”*. UNIVERSIDAD TECNICA DE BABAHOYO. Retrieved from <http://dspace.utb.edu.ec/bitstream/49000/99/1/T-UTB-FAFI-SIT-000001.pdf>
- INEI. (2017). Robo en la vivienda. Retrieved from https://www.inei.gob.pe/media/MenuRecursivo/publicaciones_digitaless/Est/Lib1519/cap03.pdf
- INEI & MININTER. (2018). N° 092 – 04 junio 2018, pp. 29–31. Retrieved from <https://www.inei.gob.pe/media/MenuRecursivo/noticias/nota-de-prensa-no-092-2018-inei.pdf>
- Ingenieria de Software. (2017). PROGRAMACION EXTREMA XP.
- IONOS. (2019). Conoce los tipos de redes más importantes. Retrieved from <https://www.ionos.es/digitalguide/servidores/know-how/los-tipos-de-redes-mas-conocidos/>
- KJURADOE. (2013). LA ARQUITECTURA. Retrieved from <https://domoticacesmag.wordpress.com/2013/10/15/la-arquitectura/>

- Llamas, L. (2016). ALARMA CON ARDUINO Y BUZZER ACTIVO (ZUMBADOR). Retrieved from <https://www.luisllamas.es/arduino-buzzer-activo/>
- Lledó, E. (2012). *Diseño de un sistema de control domótico basado en la plataforma Arduino*. Universidad Politécnica de Valencia. Retrieved from <https://riunet.upv.es/bitstream/handle/10251/18228/Memoria.pdf>
- LOWES. (2016). FOCO LED 10WATTS A19 E27 LUZ CALIDA 3PACK LINVAL. Retrieved from <https://www.lowes.com.mx/home/6411-foco-led-10-w-a19-e27-luz-calida-3-piezas-linval.html>
- Martínez, D. (2015). *Sistema De Domótica Para Control Y Supervisión De Una Habitación De Manera Remota*. Pontificia Universidad Javeriana Bogotá. Retrieved from <http://hdl.handle.net/10554/16522>
- Mejía, L. (2016). *Sistema De Integración De Dispositivos Electrónicos Automatizados Con Android Y Arduino a Través De Bluetooth*. Universidad Católica De Santa María Facultad De Ciencias E Ingenierías Físicas Y Formales Escuela Profesional De Ingeniería De Sistemas Sistema. Universidad Católica de Santa María. Retrieved from <https://tesis.ucsm.edu.pe/repositorio/handle/UCSM/5452>
- MercadoLibre. (2018). Camara Seguridad Ip Inalámbrica Wifi Monitorio De Celular. Retrieved from https://articulo.mercadolibre.com.pe/MPE-430994945-camara-seguridad-ip-inalambrica-wifi-monitorio-de-celular-_JM?quantity=1
- MundoGeek. (2019). No-IP. Retrieved from <http://mundogeek.net/archivos/2005/09/28/no-ip/>
- NativeScript. (2018a). Crea aplicaciones nativas para iOS y Android con JavaScript. Retrieved from <https://www.nativescript.org/>
- NativeScript. (2018b). NativeScript y Vue.js. Retrieved from <https://www.nativescript.org/vue>
- No-ip. (2019). Remote Access with Dynamic DNS. Retrieved from <https://www.noip.com/remote-access?gclid=CjwKCAjw->

Zv1BRBbEiwANw9UWvnnXJ-WL7R6Mvm09D-
sQnOydAIDTQtxwCfxZ8G0YPTkclNFK2F6BoC7lsQAvD_BwE&utm_campaign=
US Canada Mexico google&utm_content=US Canada Mexico Google Plus
Form&utm_medium=cpc&utm_source=google&utm_te

Núñez, F. (2017). *IMPLEMENTACIÓN DE UN PROTOTIPO DE UN SISTEMA DOMÓTICO BASADO EN ARDUINO PARA CONTROL, MONITOREO Y ASISTENCIA EN HOGARES PARA PERSONAS DE LA TERCERA EDAD*. Universidad de Las Américas. Retrieved from <http://dspace.udla.edu.ec/handle/33000/7396>

Paiz, J. (2015). *DISEÑO E IMPLEMENTACIÓN DE UNA CONEXIÓN REMOTA BIDIRECCIONAL USANDO TELÉFONO MÓVIL CON SISTEMA OPERATIVO ANDROID PARA TERMINALES DE POTENCIA APLICADOS EN DOMÓTICA*. Universidad Nacional de Ingeniería. Retrieved from <http://cybertesis.uni.edu.pe/handle/uni/3198>

Paz, G., Rodarte, J., & Pérez, R. (n.d.). *Casa inteligente y segura*. Universidad Autónoma de Ciudad Juárez. Retrieved from <http://www.uacj.mx/DGDCDC/SP/Documents/RTI/RTI/4. Casa inteligente.pdf>

Prieto, F. (2007). *Transmisión de imágenes de vídeo mediante Servicios Web XML sobre J2ME*. UNIVERSIDAD DE SEVILLA. Retrieved from <http://bibing.us.es/proyectos/abreproy/11372/fichero/Memoria%252FMemoria+completa.pdf>

Ramírez, D. (2017). Comparativa entre XP y SCRUM. Retrieved from <http://davidrtmetodosagiles.blogspot.com/2017/02/comparativa-entre-xp-y-scrum.html>

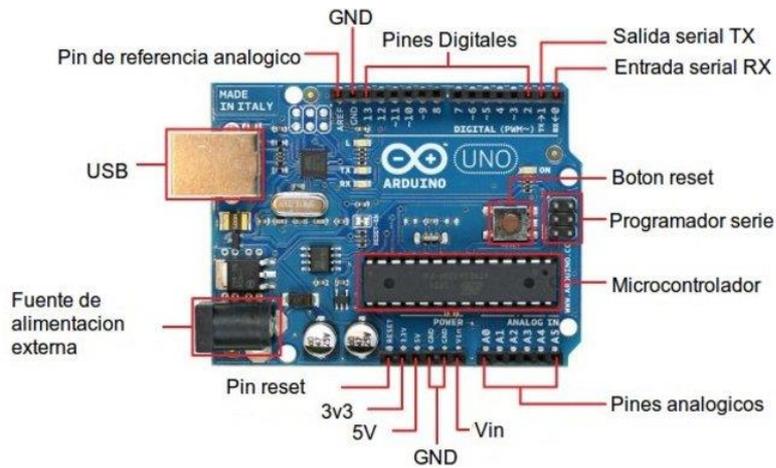
Scoello. (2012). SISTEMA ANDROID. Retrieved from <https://scoello12.wordpress.com/ventajas-y-desventajas/>

- SHERLIN. (2018). Arquitectura de microcontroladores. Retrieved from <http://sherlin.xbot.es/microcontroladores/introduccion-a-los-microcontroladores/arquitectura-de-microcontroladores>
- Spira, J. y R. (1961). Fundador de Lutron Electronics. Retrieved from https://engineering.purdue.edu/ECE/People/Teaching_awards/spira_award
- Techopedia. (2019). Mobile Application (Mobile App). Retrieved from <https://www.techopedia.com/definition/2953/mobile-application-mobile-app>
- tecnologia. (2014). DOMÓTICA. Retrieved from <http://tecnometal2.blogspot.com/2014/07/domotica.html>
- Thonti, V. (2017). Relay: Construction, Working and Types ELECTRONICS. Retrieved from <https://circuitdigest.com/article/relay-working-types-operation-applications>
- Uriel. (2017). Aplicación híbrida o nativa: ¿Cuál es mejor? Retrieved from <https://platzi.com/blog/hibrida-o-nativa/>
- VanToll, T. (2018). Getting Started with NativeScript-Vue 1.0. Retrieved from <https://medium.com/js-dojo/getting-started-with-nativescript-vue-1-0-34cacf3f3a1e>
- Veselý, D. (2017). *Analysis and experiments with NativeScript and React Native framework. Masaryk University. Faculty of Informatics Analysis.* Masaryk University. Retrieved from <https://is.muni.cz/th/bkg22/thesis.pdf>

ANEXOS

Anexo A. Especificaciones técnicas de hardware usado

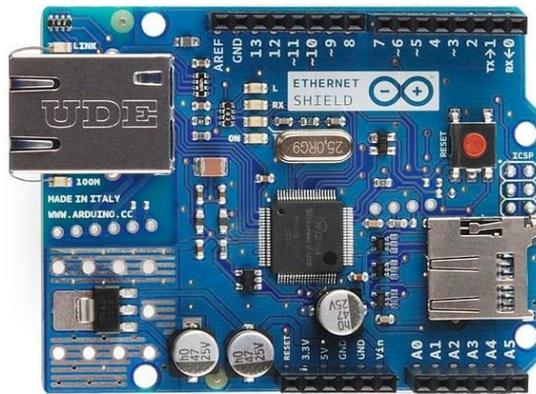
Arduino Uno R3



Especificaciones técnicas:

- Microcontrolador ATmega328.
- Voltaje de entrada 7-12V.
- 14 pines digitales de I/O (6 salidas PWM).
- 6 entradas análogas.
- 32k de memoria Flash.
- Reloj de 16MHz de velocidad.

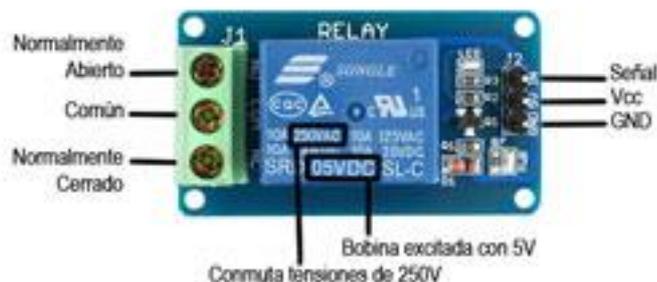
Arduino shield ethernet



Especificaciones técnicas:

- El shield provee un conector ethernet estándar RJ45.
- La ethernet shield dispone de unos conectores que permiten conectar a su vez otras placas encima y apilarlas sobre la placa Arduino.
- Arduino usa los pines digitales 10, 11, 12, y 13 (SPI) para comunicarse con el W5100 en la ethernet shield. Estos pines no pueden ser usados para e/s genéricas.
- El botón de reset en la shield resetea ambos, el W5100 y la placa Arduino.
- ON: indica que la placa y la shield están alimentadas
- LINK: indica la presencia de un enlace de red y parpadea cuando la shield envía o recibe datos
- 100M: indica la presencia de una conexión de red de 100 Mb/s (de forma opuesta a una de 10Mb/s)
- RX: parpadea cuando el shield recibe datos
- TX: parpadea cuando el shield envía datos

Relé



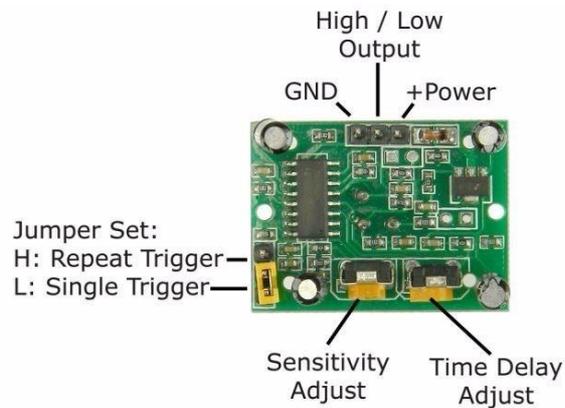
Especificaciones técnicas:

- Corriente de excitación.- Intensidad, que circula por la bobina, necesaria para activar el relé.
- Tensión nominal.- Tensión de trabajo para la cual el relé se activa.
- Tensión de trabajo.- Margen entre la tensión mínima y máxima, garantizando el

funcionamiento correcto del dispositivo.

- Consumo nominal de la bobina.- Potencia que consume la bobina cuando el relé está excitado con la tensión nominal a 20°C.
- Tensión de conexión.- Tensión entre contactos antes de cerrar o después de abrir.
- Intensidad de conexión.- Intensidad máxima que un relé puede conectar o desconectarlo.
- Intensidad máxima de trabajo.- Intensidad máxima que puede circular por los contactos cuando se han cerrado.

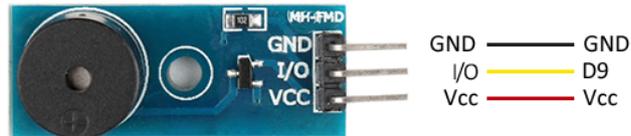
Sensor PIR



Especificaciones técnicas:

- Sensor piroeléctrico (Pasivo) infrarrojo (También llamado PIR)
- El módulo incluye el sensor, lente, controlador PIR BISS0001, regulador y todos los componentes de apoyo para una fácil utilización
- Rango de detección: 3 m a 7 m, ajustable mediante trimmer (Sx)
- Lente fresnel de 19 zonas, ángulo < 100°
- Salida activa alta a 3.3 V
- Tiempo en estado activo de la salida configurable mediante trimmer (Tx)
- Redisparo configurable mediante jumper de soldadura
- Consumo de corriente en reposo: < 50 μ A
- Voltaje de alimentación: 4.5 VDC a 20 VDC

Sensor buzzer pasivo arduino



Especificaciones técnicas:

- Zumbador pasivo a bordo
- Unidad de triodo 8550 a bordo
- Puede controlar con microcontrolador de un solo chip IO directamente
- Voltaje de trabajo: 5V
- Tamaño del tablero: 22 (mm) x 12 (mm)

Cámara IP



Requisitos:

- Celular con sistema ANDROID o IOS APPLE.
- La cámara debe tener acceso internet en el lugar donde se va a instalar (sea cableado o wifi).
- Punto de energía 220V AC o 5V DC usb para conectar la cámara.

Características:

- Giro Horizontal 355°
- Giro vertical 120°
- Visión en vivo hacia el celular.
- Cuenta con infrarojos, para vigilancia nocturna.

- Soporta micro sd de hasta 64gb.

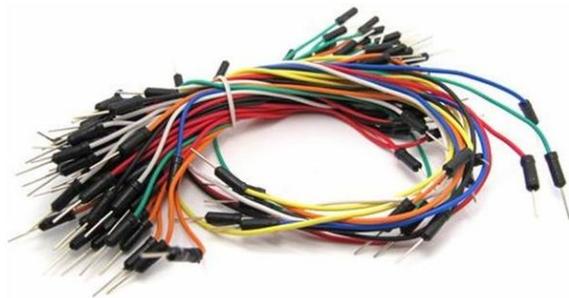
Foco LED



Especificaciones técnicas:

- Potencia: 19W
- Flujo luminoso: 2300 lm
- Voltaje: 100-240 V
- Temperatura de color: Luz Blanca(6500k)
- Frecuencia: 60 HERTZ
- Base: E26, E27
- Vida: 25, 000 Horas

Cables Dupont



Especificaciones técnicas:

- Tipo: Hembra a hembra, hembra a macho
- Espaciado: 2.54mm (0.1 pulgada)
- Longitud: 20cm, 10cm
- Cantidad: 6 unidades

Smartphone para prueba



Características:

- Procesador: Snapdragon 435 1.4GHz
- RAM: 2GB
- Almacenamiento: 16GB
- OS: Android 7.0
- Pantalla: 5.5", 720 x 1280 pixels

Anexo B. Código de Arduino

```
#include <SPI.h> //Aqui incluimos la libreria SPI
#include <Ethernet.h> //Aqui incluimos la libreria Ethernet
byte mac[]={0xDE,0xAD,0xBE,0xEF,0xFE,0xED}; //Declaracion de la direccion MAC

IPAddress ip(192, 168, 1, 70);

EthernetServer servidor(5300); //Declaracion del puerto 5300

int PIN_LED=8; //Aqui establecemos la variable PIN_LED como un valor entero
int PIN_BUZZER=9; //Aqui establecemos la variable PIN_BUZZER como un valor entero
String readString=String(30); //lee los caracteres de una secuencia en una cadena.
//Los strings se representan como arrays de caracteres (tipo char)
String state=String(3);

void setup() {
  Ethernet.begin(mac, ip); //Inicializamos con las direcciones asignadas
  servidor.begin(); //inicia el servidor

  pinMode(PIN_LED,OUTPUT);
  digitalWrite(PIN_LED,LOW);
  state="OFF";

  pinMode(PIN_BUZZER,OUTPUT);
```

```

digitalWrite(PIN_BUZZER,HIGH);
state="OFF";
}

void loop() {
//EthernetClient Crea un cliente que se puede conectar a
//una dirección específica de Internet IP
EthernetClient cliente= servidor.available();
if(cliente) {
boolean lineaenblanco=true;
while(cliente.connected()) {
if(cliente.available()) {
char c=cliente.read();
if(readString.length()<30) {
readString.concat(c);
//Cliente conectado
//Leemos petición HTTP caracter a caracter
//Almacenar los caracteres en la variable readString
}
if(c=='\n' && lineaenblanco) //Si la petición HTTP ha finalizado
{
int LED = readString.indexOf("LED=");
if(readString.substring(LED,LED+5)=="LED=T") {
digitalWrite(PIN_LED,HIGH);
state="ON"; }
else if (readString.substring(LED,LED+5)=="LED=F") {
digitalWrite(PIN_LED,LOW);
state="OFF";
}
}

int BUZZER = readString.indexOf("BUZZER=");
if(readString.substring(BUZZER,BUZZER+8)=="BUZZER=T") {
digitalWrite(PIN_BUZZER,LOW);
state="ON"; }
else if (readString.substring(BUZZER,BUZZER+8)=="BUZZER=F") {
digitalWrite(PIN_BUZZER,HIGH);
state="OFF";
}
}

//Cabecera HTTP estándar
cliente.println("HTTP/1.1 200 OK");
cliente.println("Content-Type: text/html");
cliente.println(); //Página Web en HTML
cliente.println("<html>");
cliente.println("<head>");
cliente.println("<title>ArduinoServer</title>");
cliente.println("</head>");

```

```

cliente.println("<body width=80% height=80%>");
cliente.println("<center>");
cliente.println("<h1>Control de Iluminacion</h1>");
cliente.print("<br><br>");
cliente.print("Estado del FOCO: ");
cliente.print(state);
cliente.print("<br><br><br><br>");
cliente.println("<input type=submit value=ON style=width:100px;height:38px
onClick=location.href='./?LED=T\>");
cliente.println("<input type=submit value=OFF style=width:100px;height:38px
onClick=location.href='./?LED=F\>");
cliente.println("</center>");

cliente.println("<center>");
cliente.println("<h1>Control de Alarma</h1>");
cliente.print("<br><br>");
cliente.print("Estado del ALARMA: ");
cliente.print(state);
cliente.print("<br><br><br><br>");
cliente.println("<input type=submit value=ON style=width:100px;height:38px
onClick=location.href='./?BUZZER=T\>");
cliente.println("<input type=submit value=OFF style=width:100px;height:38px
onClick=location.href='./?BUZZER=F\>");
cliente.println("</center>");

cliente.println("</body>");
cliente.println("</html>");
cliente.stop();
//Cierro conexión con el cliente
readString="";
}
}
}
}
}
}
}

```

Anexo C. Código NativeScript-Vue

LoginPage.vue

```
<template>
  <Page>
    <ScrollView>
      <FlexboxLayout class="page">
        <StackLayout class="form">
          <Image class="logo" src="~/assets/images/dm_logo.png" />
          <Label class="header" text="DomoPit" />

          <StackLayout class="input-field" marginBottom="25">
            <TextField class="input" hint="Email" keyboardType="email" autocorrect="false"
autocapitalizationType="none" v-model="user.email"
            returnKeyType="next" @returnPress="focusPassword" fontSize="18" />
            <StackLayout class="hr-light" />
          </StackLayout>

          <StackLayout class="input-field" marginBottom="25">
            <TextField ref="password" class="input" hint="Password" secure="true" v-
model="user.password" :returnKeyType="isLoggingIn ? 'done' : 'next' "
            @returnPress="focusConfirmPassword" fontSize="18" />
            <StackLayout class="hr-light" />
          </StackLayout>

          <StackLayout v-show="!isLoggingIn" class="input-field">
            <TextField ref="confirmPassword" class="input" hint="Confirm password"
secure="true" v-model="user.confirmPassword" returnKeyType="done"
            fontSize="18" />
            <StackLayout class="hr-light" />
          </StackLayout>

          <Button :text="isLoggingIn ? 'Ingresar' : 'Registrar'" @tap="submit" class="btn btn-
primary m-t-20" />
          <Button v-show="isLoggingIn" :text="\uf1a0' +' Google login' "
@tap="loginGoogle" class="fab btn btn-active" />
          <Label v-show="isLoggingIn" text="¿Olvidaste tu contraseña?" class="login-label"
@tap="forgotPassword" />
        </StackLayout>

        <Label class="login-label sign-up-label" @tap="toggleForm">
          <FormattedString>
            <Span :text="isLoggingIn ? '¿No tiene una cuenta?' : 'Atrás'" />
            <Span :text="isLoggingIn ? 'Regístrese' : ''" class="bold" />
          </FormattedString>
        </Label>
      </FlexboxLayout>
    </ScrollView>
  </Page>
</template>
```

```

</Page>
</template>
<script>
// A stub for a service that authenticates users.
import Player from "@/components/Player/Player";
import firebase from "nativescript-plugin-firebase";
const userService = {
  async register(user) {
    return await firebase.createUser({
      email: user.email,
      password: user.password
    });
  },
  async login(user) {
    return await firebase.login({
      type: firebase.LoginType.PASSWORD,
      passwordOptions: {
        email: user.email,
        password: user.password
      }
    });
  },
  async loginGoogle(user) {
    await firebase
      .login({
        type: firebase.LoginType.GOOGLE
      })
      .then(result => {
        return Promise.resolve(JSON.stringify(result));
      })
      .catch(error => {
        console.error(error);
        return Promise.reject(error);
      });
  },
  async resetPassword(email) {
    return await firebase.resetPassword({
      email: email
    });
  }
};
// A stub for the main page of your app. In a real app you'd put this page in its own .vue
file.
var LoadingIndicator = require("nativescript-loading-indicator")
  .LoadingIndicator;
var loader = new LoadingIndicator();
export default {

```

```

component: {
  Player
},
data() {
  return {
    isLoggingIn: true,
    user: {
      email: "mireya@test.com",
      password: "123456",
      confirmPassword: ""
    }
  };
},
methods: {
  toggleForm() {
    this.isLoggingIn = !this.isLoggingIn;
  },
  submit() {
    if (!this.user.email || !this.user.password) {
      this.alert("Please provide both an email address and password.");
      return;
    }
    loader.show();
    if (this.isLoggingIn) {
      this.login();
    } else {
      this.register();
    }
  },
  login() {
    userService
      .login(this.user)
      .then(() => {
        loader.hide();
        this.$navigateTo(Player);
      })
      .catch(err => {
        console.error(err);
        loader.hide();
        this.alert(err);
      });
  },
  loginGoogle(){
    //loader.show();//Don't use this for google logins, as the popup covers the UI on IOS
    userService
      .loginGoogle(this.user)
      .then((result) => {

```

```

    //loader.hide();
    this.$navigateTo(Player);
  })
  .catch((error) => {
    //loader.hide();
    console.error(err);
    this.alert(error)
  });
},
register() {
  if (this.user.password !== this.user.confirmPassword) {
    loader.hide();
    this.alert("Tus contraseñas no coinciden.");
    return;
  }
  if (this.user.password.length < 6) {
    loader.hide();
    this.alert("Tu contraseña debe tener al menos 6 caracteres.");
    return;
  }
  var validator = require("email-validator");
  if (!validator.validate(this.user.email)) {
    loader.hide();
    this.alert("Ingrese un email válido.");
    return;
  }
  userService
    .register(this.user)
    .then(() => {
      loader.hide();
      this.alert("Tu cuenta ha sido creada con éxito.");
      this.isLoggingIn = true;
    })
    .catch(err => {
      console.error(err);
      loader.hide();
      this.alert(err);
    });
},
forgotPassword() {
  prompt({
    title: "Forgot Password",
    message:
      "Enter the email address you used to register for APP NAME to reset your
password.",
    inputType: "email",
    defaultText: "",
  });
}

```

```

        okButtonText: "Ok",
        cancelButtonText: "Cancel"
    }).then(data => {
        if (data.result) {
            loader.show();
            userService
                .resetPassword(data.text.trim())
                .then(() => {
                    loader.hide();
                    this.alert(
                        "Your password was successfully reset. Please check your email for instructions
on choosing a new password."
                    );
                })
                .catch(err => {
                    loader.hide();
                    this.alert(err);
                });
        }
    });
},
focusPassword() {
    this.$refs.password.nativeView.focus();
},
focusConfirmPassword() {
    if (!this.isLoggingIn) {
        this.$refs.confirmPassword.nativeView.focus();
    }
},
alert(message) {
    return alert({
        title: "MENSAJE",
        okButtonText: "OK",
        message: message
    });
}
};
</script>

<style scoped>
.page {
    align-items: center;
    flex-direction: column;
}

.form {

```

```
margin-left: 30;
margin-right: 30;
flex-grow: 2;
vertical-align: middle;
}
```

```
.logo {
margin-bottom: 12;
height: 90;
font-weight: bold;
}
```

```
.header {
horizontal-align: center;
font-size: 25;
font-weight: 600;
margin-bottom: 70;
text-align: center;
color: #36864C;
}
```

```
.input-field {
margin-bottom: 25;
}
```

```
.input {
font-size: 18;
placeholder-color: #a8a8a8;
}
```

```
.input-field input {
font-size: 54;
}
```

```
.btn-primary {
height: 50;
margin: 30 5 15 5;
background-color: #36864C;
border-radius: 5;
font-size: 20;
font-weight: 600;
}
```

```
.login-label {
horizontal-align: center;
color: #a8a8a8;
font-size: 16;
```

```

}

.btn-active {
  font-weight: bold;
  color: #ffffff;
  background-color: #3699DB;
}

.sign-up-label {
  margin-bottom: 20;
}

.bold {
  color: #000000;
}
</style>

```

Player.vue

```

<template>
  <Page class="page">
    <ActionBar title="Home" class="action-bar" />
    <ScrollView>
      <StackLayout class="home-panel">
        <WebView height="1000px" src="http://192.168.1.70:5300/">
        <Button text="Ver Cámara" @tap="submit" class="buttoncam" />
      </StackLayout>
    </ScrollView>
  </Page>
</template>

<script>

var openApp = require("nativescript-open-app").
openApp;

export default {
  methods: {
    submit() {
      var installed = openApp("com.xm.csee",false);
      console.log("Is it installed? " + installed);
    }
  },
  data () {
    return {

```

```
    };
  },
}
</script>
```

```
<style scoped>
```

```
.buttoncam {
  background-color: #36864C;
  font-weight: bold;
  color: #ffffff;
}
</style>
```

Main.js

```
import Vue from 'nativescript-vue'
import VueDevtools from 'nativescript-vue-devtools'
import LoginPage from './components/Login/LoginPage'
import Player from './components/Player/Player'
if(TNS_ENV !== 'production') {
  Vue.use(VueDevtools)
}
// Prints Vue logs when --env.production is *NOT* set while building
Vue.config.silent = (TNS_ENV === 'production')

var firebase = require("nativescript-plugin-firebase");
firebase
  .init({
    // Optionally pass in properties for database, authentication and cloud messaging,
    // see their respective docs.
  })
  .then(
    function(instance) {
      console.log("firebase.init done");
    },
    function(error) {
      console.log("firebase.init error: " + error);
    }
  );

new Vue({
  render: h => h('frame', [h(LoginPage), Player])
}).$start()
```

Anexo D. Formatos

Formato de historias de usuario

Historia de Usuario	
Número:	Nombre:
Usuario:	
Modificación de Historia Número:	Iteración Asignado:
Prioridad en Negocio: (Alta/Media/Baja)	Puntos Estimados:
Riesgo en Desarrollo: (Alto/Medio/Bajo)	Puntos Reales:
Descripción:	
Observaciones:	

Formato de asignación de tareas

Tarea	
Número:	Numero de Historia:
Nombre:	
Tipo de tarea:	Puntos Estimados:
Fecha Inicio:	Fecha fin:
Programador Responsable:	

Formato de pruebas de aceptación

Caso de Prueba	
Número de Caso de Prueba:	Número Historia de Usuario:
Nombre Caso de Prueba:	
Descripción:	
Condiciones de ejecución:	
Entradas:	
Resultado esperado:	
Evaluación:	

Anexo E. Código en repositorio Github

<https://github.com/AngieChambi/DomoPitAppMovil>

The screenshot shows a GitHub repository page for 'AngieChambi / DomoPitAppMovil'. The page includes a navigation bar with 'Code', 'Issues', 'Pull requests', 'Projects', 'Security', and 'Insights'. A 'Join GitHub today' banner is present. Below the banner, the repository statistics are shown: 4 commits, 1 branch, 0 releases, and 1 contributor. The 'Branch: master' is selected, and there are buttons for 'New pull request', 'Find File', and 'Clone or download'. The commit history is displayed, showing the latest commit by AngieChambi with the message 'Subo el app movil final' 8 hours ago. The commit details list files: 'app', '.gitignore', 'README.md', 'babel.config.js', and 'firebase.nativescript.json', all with the message 'Subo la estructura del proyecto al repositorio de GitHub' 8 hours ago.