

UNIVERSIDAD PÚBLICA DE EL ALTO

CARRERA INGENIERÍA DE SISTEMAS



PROYECTO DE GRADO

SISTEMA DE CONTROL DOMÓTICO BASADO EN ARDUINO Y APLICACIÓN MÓVIL

Caso: Sala de Reuniones de la Asociación Municipal de
Futbol “AMFEAL”

Para Optar al Título de Licenciatura en Ingeniería de Sistemas

MENCIÓN: GESTIÓN Y PRODUCCIÓN

Postulante:	Univ. Nelson Guarachi Mita
Tutor Metodológico:	Ing. Dionicio Henry Pacheco Ríos
Tutor Revisor:	Ing. Sergio Ramiro Rojas Saire
Tutor Especialista:	Lic. Edgar Salcedo Condori

EL ALTO – BOLIVIA

2020

Dedicatoria

El presente proyecto de grado va dedicado con mucho cariño a mis padres, familiares por el apoyo incondicional que me brindaron en cada momento de mi vida.

A mi querida novia que siempre ha estado en todo este tiempo apoyándome incondicionalmente.

A mis amigos que siempre me dieron el apoyo incondicional en todo momento.

A mis tíos que siempre me brindaron su apoyo, que demostraron su preocupación e interés por verme superar y triunfar en la vida.

Nelson Guarachi Mita

AGRADECIMIENTOS

Agradezco a Dios por hacer realidad mi sueño y cumplir con unas de mis metas.

A la Universidad Pública de El Alto UPEA por ser parte de ella y abrirme las puertas en su seno para poder estudiar mi Carrera Ingeniería de Sistemas, así como también a los diferentes docentes que brindaron sus conocimientos y apoyo para seguir adelante día a día. A mi padres y tíos, amigos que me acompañaron a lo largo de este camino, brindándome apoyo y ánimo para salir adelante. De la misma manera agradezco al Ing. Edgar Salceso Condori, Ing. Sergio Ramiro Rojas, quienes fueron mis guías durante el desarrollo del proyecto de grado. Al presidente de la Asociación Municipal de Fútbol de El Alto, Sr. Abdón Suntura Mamani Sea por la colaboración y aceptación para la elaboración del proyecto de grado. A todos quienes me apoyaron en los momentos que más necesite, viéndome como un ser capaz de alcanzar éxito para hacer de este sueño realidad.

RESUMEN

En la actualidad toda institución requiere de las tecnologías, sistemas de automatización e informáticos para tener un mejor desempeño dentro de su organización obteniendo mayores beneficios económicamente y en el ámbito de la competencia.

Los sistemas de control de automatización han llegado a ser una herramienta indispensable para estas entidades, por lo cual la mayor parte se inclina por hacer uso de la implementación de sistema domótico, que logre la mayor organización de su institución.

En el presente documento se describe el proceso de desarrollo del sistema de control domótico basado en Arduino y aplicación móvil, asimismo, se hace un análisis de la problemática que enfrenta la asociación de fútbol El Alto, para posteriormente describir la solución a los problemas identificados, también se hace énfasis en el uso de las herramientas a utilizarse y finalmente lograr un sistema de aplicación móvil confiable y eficiente.

Para modelar el sistema y documentarlo se utilizó las metodologías MOBILE-D, MODELO EN V, evaluación de calidad de software ISO 9001, estimación de costos COCOMO II y para su desarrollo programas del Arduino, lenguaje de programación y librerías de diseño para la parte de documentación.

Palabras Clave: Domótica, Arduino, metodologías, MOBILE-D, MODELO EN V, Automatización.

ABSTRACT

At present, every institution requires technologies, automation and computer systems to perform better within its organization, obtaining greater benefits economically and in the field of competition. Automation control systems have become an indispensable tool for these entities, which is why most of them are inclined to make use of the implementation of a home automation system, which achieves the greatest organization of their institution.

This document describes the development process of the home automation control system based on Arduino and mobile application, likewise, an analysis of the problems faced by the El Alto football association is made, to later describe the solution to the identified problems. emphasis is also placed on the use of the tools to be used and finally achieve a reliable and efficient mobile application system.

To model the system and document it, the methodologies MOBILE-D, V MODEL, ISO 9001 software quality evaluation, COCOMO II cost estimation were used and for its development Arduino programs, programming language and design libraries for the part of documentation.

Keywords: Home automation, Arduino, methodologies, MOBILE-D, V-MODEL, Automation.

INDÍCE GENERAL

Pág.

CAPÍTULO I	1
1.MARCO PRELIMINAR	1
1.1 INTRODUCCIÓN	2
1.2 ANTECEDENTES	2
1.2.1 Antecedentes de la Institución	2
1.2.2 Antecedentes de Trabajos Afines	3
1.3.3 Antecedentes Internacionales	3
1.3 Problema Principal	5
1.3.1 Problemas Secundarios	5
1.4 OBJETIVOS	5
1.4.1 Objetivo General	5
1.4.2 Objetivos Específicos	6
1.5.1 Justificación Técnica.	6
1.5.2 Justificación Económica	6
1.5.3 Justificación Social	6
1.6 METODOLOGÍA	7
1.6.1 Mobile-D	7
1.7 MÉTODO DE INGENIERÍA	7
1.7.1 Metodología de Costos (COCOMO II).	8
1.7.2 Seguridad ISO 9001	8
1.8 TÉCNICAS DE INVESTIGACIÓN	8
1.9 HERRAMIENTAS	8
1.10 LÍMITES Y ALCANCES	12
1.10.1 Límites	12
1.10.2 Alcances	12
1.11 APORTES	13
CAPITULO II	14
2.MARCO TEORICO	14

2. INTRODUCCIÓN.....	15
2.1 DOMÓTICA	15
2.2.1. Confort.....	15
2.2.2 Seguridad	16
2.2.3 Comunicación.....	16
2.2.4 Accesibilidad	16
2.3 ARDUINO.....	17
2.3.1 Características.....	18
2.3.2 Alimentación de un Arduino.....	19
2.3.3 Memoria.....	20
2.3.4 Entradas y Salidas.....	20
2.3.5 Comunicaciones.....	21
2.3.6 Programación.	22
2.3.7 Protección contra sobre corrientes en USB	23
2.3.8 Características físicas y compatibilidad de Shields.....	24
2.4 Seguridad.....	24
2.4.1 Gestión de Seguridad en el hogar	24
2.4.2 Vigilancia Interna y Externa.....	25
2.4.3 Seguridad Perimetral.....	26
2.4.5 Control de Acceso.....	27
2.4.6 Alarma de Agresión.....	27
2.4.7 Centrales de Alarma	28
2.5 SENSORES.....	28
2.5.1 Sensores de Infrarrojos	29
2.5.2 Sensores de Ultrasonidos	30
2.5.3 Sensores de Temperatura.....	30
2.5.4 Sensor de Movimiento	31
2.6 SERVOMOTORES.....	32
2.6.1 Partes de un servomotor	34
2.7 TECNOLOGÍAS DE SOFTWARE.....	34

2.8 METODOLOGÍA	35
2.8.1 Modelo en V	35
2.8.1 Mobile-D	37
2.8.2 Fases de la Metodología	37
2.8.3 METODO DE INGENIERIA	41
2.9 Metodología de Costos (COCOMO II).	43
2.10 Seguridad ISO/IEC 27000.....	43
CAPÍTULO III	44
3. DISEÑO METODOLOGICO	44
3.1. INTRODUCCION.....	45
3.2 Metodología de desarrollo en “V “	45
3.1.2 Fase 1 Definición de Especificaciones y Requerimientos	45
3.1.2 Fase 2 Diseño Global	46
3.1.3 Fase 3 Diseño en Detalle.....	47
3.1.4. Fase 4 Implementación	48
3.1.5 Fase 5 test Unitario	49
3.1.6 Fase 6 Integración	50
3.1.7. Fase 7 Test operacional del Sistema	51
3.3 MODELO DE SISTEMA.....	52
3.4 HARDWARE DEL SISTEMA	53
3.5 DESARROLLO DEL SISTEMA EN BASE A LA METODOLOGÍA MOBILE D.....	54
3.5.1 Fase I - Exploración.....	54
3.5.2 Definir Alcance	54
3.5.2 Fase II – Inicialización	56
3.5.3 Fase III - Fase del Producto	58
3.5.4 Fase IV - Fase de Estabilización y Fase de Pruebas – Corrección..	59
3.5.5 Fase 5	66
CAPÍTULO IV.....	69
4. Seguridad, Control	70

4.1 Seguridad.....	70
4.2 Identificación y Autenticación.....	70
4.5 Análisis Costo-Beneficio del Sistema	70
4.6 Modelo Base de Estimación COCOMO.....	71
CAPITULO V.....	75
CONCLUSIONES Y RECOMENDACIONES	75
5.1 Conclusiones.....	76
5.2 Recomendaciones.....	76
Bibliografía.....	78

Índice De Tablas

Tabla 2.1 : Características Técnicas de Arduino Mega 2560	19
Tabla 3.1 Procesos del prototipo	45
Tabla 3.2 Historia de usuario acceso al sistema de control	46
Tabla 3.3 Documentación de Pruebas 1	49
Tabla 3.4 Documentación de Pruebas	50
Tabla 3.13 Documentación de Pruebas 5	50
Tabla 3.6 Documentación de Pruebas	51
Tabla 3.7 Planeación Inicial Del Proyecto	55
Tabla 3.8 Requerimientos Iniciales	55
Tabla 3.9 Plan de entrenamiento	57
Tabla 3.10 Lista de verificación del proyecto.....	58
Tabla 3.11: Menú principal.....	60
Tabla 3.12: Menú principal.....	60
Tabla 3.13.: Usuario.....	62
Tabla 3.14: Menú principal.....	63
Tabla 3.15: Menú principal.....	65

Índice De Figuras

Figura 1: Organigrama de la A.M.F.E.A.L	3
Figura: 2 Faces Mobile - D	7
Figura: 2.1 Domótica	15
Figura 2.2. Arduino	18
Figura 2.4: Seguridad domótica en el Hogar.....	25
Figura 2. 5: Seguridad perimetral en una vivienda.....	26
Figura 2. 6: Controlador de acceso.....	27
Figura 2. 7: Sensores Domóticos figura 2.7	29
Figura 2. 8: Sensor de Ultrasonido	30
Figura 2. 9: Sensor de Temperatura	31
Figura 2. 10: Sensor de Movimiento PIR	32
Figura 2. 11: Servomotor de giro de 0° - 180°	33
Figura 2. 12: Estructura interna de un Servomotor	34
Figura 2. 13 fase de Exploración.....	38
Figura 2. 14 fase de Inicialización.....	39
Figura 2.15 fase de Productización Estados.....	39
Figura 2.16 fase de Estabilización Estados	40
Figura 2.17 etapa de Estabilización Estados	41
Figura 3. 1: Tipología centralizada de un sistema domótico	Error!
Bookmark not defined.	
Figura 3. 2: Proceso del sistema elemental	52
Figura 3. 3: Relación de unidades del sistema	53
Figura3.1 Faces de Movile-D	54
Figura3.2 menú principal	60
Figura3.3 menú principal	61
Figura 3. 4 actor del Sistema.....	62
Figura 3.5 diagrama Global del Sistema.....	63
Figura 3. 6 actor del Sistema.....	64

CAPÍTULO I

MARCO PRELIMINAR

1.1 INTRODUCCIÓN

El constante crecimiento de la tecnología, conlleva a desarrollar sistemas inteligentes cada vez más complejos para satisfacer las necesidades que existe en la sociedad, el poder brindar confort, seguridad, control, supervisión, automatización y optimización son procesos que mejoran la calidad de vida.

Los servicios que ofrece la domótica a nuestro diario vivir sin duda buscan la calidad y la seguridad entre sus principales tareas esta, brindar formas de comunicación que actúen en la interacción a través de dispositivos que realicen una acción en nuestro hogar. Todas estas ventajas se pueden resumir en un aumento de la calidad de vida del propietario del hogar.

Gracias al crecimiento de la tecnología que va aportando al desarrollo de numerosas áreas tal el caso de la Ingeniería de Sistemas que van teniendo un gran desarrollo en la domótica y debido a la gran cantidad de empresas que brinda este servicio a un costo muy alto y aún muy poco conocido en nuestro medio es necesario buscar alternativas que nos ayuden a tener el control de la vivienda.

El uso diario de Teléfonos inteligentes ha ido generando un consumo masivo en la población, donde un gran porcentaje de esta, posee un dispositivo móvil. La preferencia por uno u otro dispositivo varía de las funcionalidades que requiere el usuario o simplemente de la condición económica que se posea, es por esto que debemos utilizar todos los recursos que nos brindan estos aparatos para elevar nuestra calidad de vida a través de aplicaciones que podamos utilizar fácilmente y también utilizando plaquetas electrónicas llamadas Arduino programables.

Es por ello que la propuesta de este Proyecto de Grado es, modelar, diseñar y construir un sistema integrado de control y seguridad compuesto de elementos computacionales para la Asociación Municipal de futbol de El Alto.

1.2 ANTECEDENTES

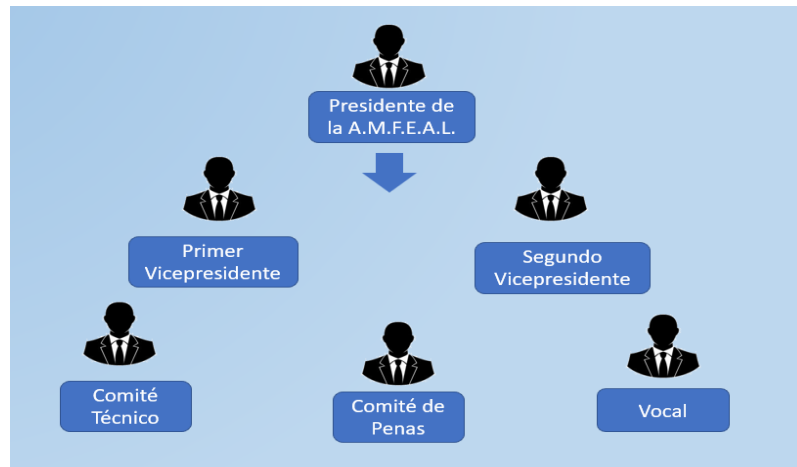
1.2.1 Antecedentes de la Institución

La Asociación Municipal Futbol de El Alto fundado el 23 de abril de 1988, con el nombre de AFEAL, posteriormente fue cambiado a “AMFEAL” en 2015 con

Personería Jurídica R.A.D .162/2015.

La asociación municipal futbol de El Alto tiene diferentes categorías como ser: Divisiones mayores y menores en dama y varones, también tiene un representante en la Copa Bolivia que se juega a nivel nacional.

Figura 1: Organigrama de la A.M.F.E.A.L



Fuente: (A.M.F.E.A.L., 2020)

1.2.2 Antecedentes de Trabajos Afines

1.3.3 Antecedentes Internacionales

- **Título:** Diseño e implementación de un sistema de automatización domótico para un salón prototipo en la facultad de ingeniería de la universidad distrital francisco José de caldas

Autor: Felipe Morales

Metodología: Karl Ulrich

Herramientas: Arduino, kotlin.

Año: 2017

Descripción: Actualmente los sistemas de domótica se enfocan en aspectos más concretos debido al ambiente de inseguridad o a las nuevas necesidades que surgen de tener un control maestro de los dispositivos. (Morales, 2017).

- **Título:** Diseño e Implementación de un Sistema de Automatización Domótico para un salón prototipo en la facultad de Ingeniería De La Universidad Distrital Francisco José de caldas.

Autor: Felipe Morales

Metodología: programación Extrema o XP.

Herramientas:

Año: 2017

Descripción: Últimamente se hace necesaria una automatización de las viviendas por varios motivos, uno de ellos es la comodidad que este tipo de instalaciones proporciona a sus usuarios. Otro de los motivos es, por supuesto, el ahorro energético que supone poder controlar todos los dispositivos electrónicos de la vivienda en cualquier momento del día. Además de ofrecer un alto grado de seguridad y comunicación a sus inquilinos. (Redondo, 2016).

Antecedentes Nacionales

- **Título:** Automatización de un Sistema de banco de agua helada en la Empresa Pil Andina S.A.

Autor: German Mario Mamani Álvarez

Metodología: Carl Ulrich

Herramientas: Condensadores Evaporativos, Compresores Rotativos.

Año: 2013.

Descripción: Automatizar el funcionamiento del banco de agua helada mediante la utilización de un sistema adecuado que permita mejorar y optimizar las Instalaciones, dentro el ciclo de operación y control de los distintos elementos y procesos que son parte de este sistema. (ÁLVAREZ, 2013)

- **Título:** Sistema Domótico para un hogar basado en Software Y Hardware Libre”

Autor: Juan Guillermo Chávez Blanco

Metodología: Modelo en V.

Herramientas: Arduino, Relé, sensores.

Año: 2015.

Descripción: El acceso a la tecnología de automatización de viviendas (domótica), es limitado debido a la carencia de recursos económicos, ya que los sistemas domóticos son costosos y sus accesorios también y difíciles de ser implementados en viviendas ya construidas, por otro lado, lo que se quiere conseguir es realizar un sistema domótico accesible a todas las familias con ciertas limitaciones utilizando software y hardware libre. (Blanco, 2015)

1.3 Problema Principal

La asociación municipal de fútbol de El Alto, actualmente cuenta con varios dispositivos electrónicos, que funcionan cada uno de ellos de forma manual, lo cual no permite controlar, automatizar e integrar la tecnología de los sistemas de seguridad, gestión energética, bienestar o comunicaciones. Y a la vez mayor comodidad para las personas, debido a que se pueda manipular las instalaciones de la institución por medio de un dispositivo inteligente.

1.3.1 Problemas Secundarios

Los principales problemas identificados son detallados a continuación:

- Tareas rutinarias no automatizadas como por ejemplo el controlar el sistema de iluminación, apertura y cierre de puertas y ventanas, implica una demora en el inicio de una actividad de trabajo en la institución.
- La falta de personal encargado de los ambientes de la institución en cuanto control de dispositivos electrónicos.
- Los dueños de la institución no tienen conocimiento sobre las nuevas tecnologías acerca de la domótica y sus beneficios.

1.4 OBJETIVOS

1.4.1 Objetivo General

Desarrollar un prototipo de control domótico basado en Arduino, por el cual será controlado por una aplicación móvil en Android, que permita automatizar y gestionar

luces, puertas, ventilador y alarma de la institución.

1.4.2 Objetivos Específicos

- Analizar la información y requerimientos de la institución para definir el alcance y la arquitectura del sistema domótico.
- Recibir datos por medio de Bluetooth y procesarlos a través de Arduino para control de la aplicación móvil.
- Desarrollar una interfaz de control sencillo en Android para el usuario.
- Integrar los elementos software, hardware y la aplicación móvil, para probar y evaluar el funcionamiento del prototipo del sistema de control domótica.
- Desarrollar el diseño lógico y físico utilizando las metodologías MOBILE-D y MODELO EN V, basado en los requerimientos que permitirá generar un análisis amplio.

1.5.1 Justificación Técnica.

El hardware o software necesario para el desarrollo del prototipo y automatización se vuelve cada vez más accesible a las personas y su costo es mucho más menor en comparación hace algunos años atrás.

1.5.2 Justificación Económica.

El presente proyecto planteada contemplara el uso de equipos tecnológicos y dispositivos móviles existentes en el medio local. Con la automatización de estos componentes se busca el aspecto de minimizar costos a comparación de empresas que ofrecen estos servicios de domótica a precios muy elevados y que incluso necesitan edificaciones adecuadas para su instalación.

1.5.3 Justificación Social

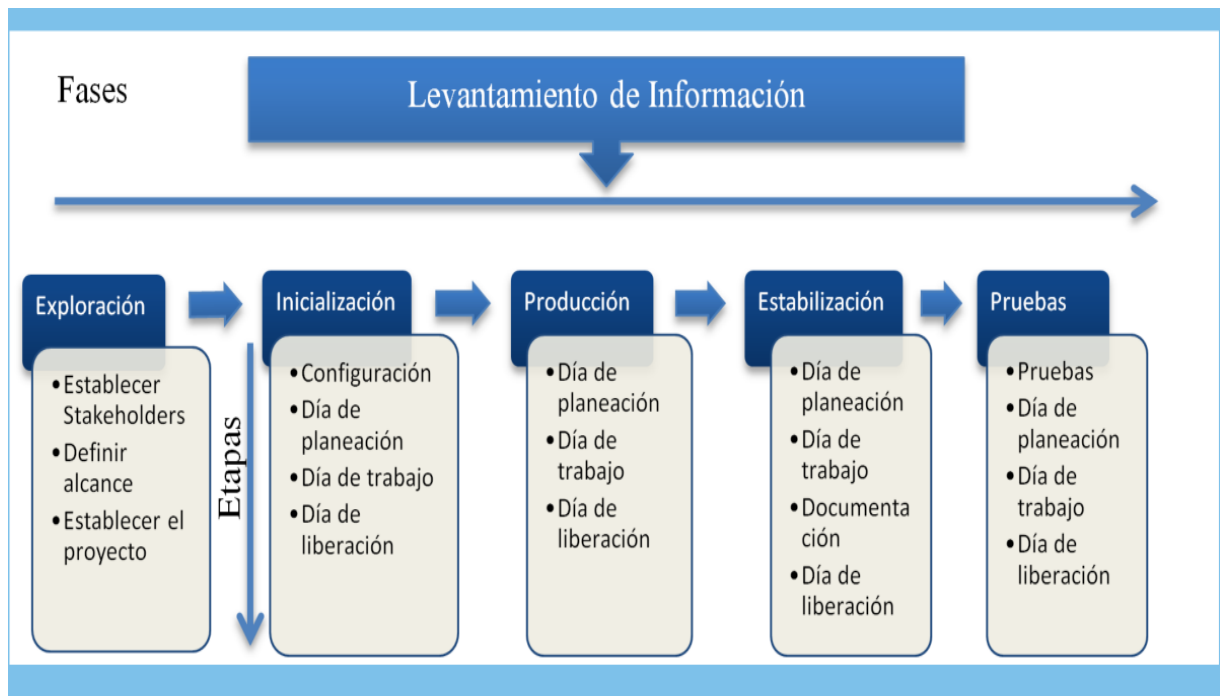
El presente proyecto va más enfocada al personal de la Asociación de futbol A.M.F.E.A.L. que mediante el uso de un teléfono inteligente pueda manejar más fácilmente su oficina, sala y también puede ser manejada por personas con alguna discapacidad de movimiento de piernas que le impida trasladarse con facilidad dentro de la institución.

1.6 METODOLOGÍA

1.6.1 Mobile-D

Fue creado en un proyecto finlandés en 2005, pero sigue estando vigente. Basado en metodologías conocidas pero aplicadas de forma estricta como: extreme programming, Crystal Methodologies y Rational Unified Process. Esta metodología está enfocada al desarrollo de software móvil. (Fernandez, 2019)

Figura: 2 Facas Mobile - D



Fuente: (Mobile - D NET)

1.7 MÉTODO DE INGENIERÍA

El método ingeniería se define como una estrategia para producir el mejor cambio, con los recursos disponibles, en una situación deficientemente entendida o incierta.

Este enfoque heurístico, propuesto por muchos autores, ha sido ampliamente preconizado

por Koen, quien afirma que el método ingenieril consiste en el uso de heurismos para producir el mejor cambio, con los recursos disponibles, en una situación deficientemente entendida. Es decir, que el método ingenieril es el uso de

heurismos de ingeniería

1.7.1 Metodología de Costos (COCOMO II).

Los objetivos principales que se tienen en cuenta para construir el modelo COCOMO II son:

Desarrollar un modelo de estimación de costo y cronograma de proyectos de software que se adaptara tanto a las prácticas de desarrollo de la década del 90 como a las futuras.

Construir una base de datos de proyectos de software que permitiera la calibración continua del modelo, y así incrementar la precisión en la estimación.

Implementar una herramienta de software que soportara el modelo.

Proveer un marco analítico cuantitativo y un conjunto de herramientas y técnicas que evaluaran el impacto de las mejoras tecnológicas de software sobre los costos y tiempos en las diferentes etapas del ciclo de vida de desarrollo. (BARRAGÁN., 2011)

1.7.2 Seguridad ISO 9001

Hoy en día, muchas organizaciones están optando por apoyarse en herramientas que les permitan automatizar los procesos para ser más ágiles y centrarse en la calidad y mejora de sus productos/servicios y no sólo en el proceso que conlleva la propia certificación de una norma ISO y el mantenimiento del Sistema de Gestión de la Calidad. Las opciones tradicionales para implementar y mantener la Gestión de la Calidad en base a los requerimientos que establece la ISO 9001 presentaban una serie de desventajas que es posible paliar trabajando con un Software ISO.

1.8 TÉCNICAS DE INVESTIGACIÓN

Para la elaboración del presente proyecto se analizarán circuitos digitales y se harán pruebas experimentales, buscando relacionar la práctica con la teoría además se elaborará un cuestionario con preguntas que recaben información sobre la seguridad en la institución donde se realizará el presente proyecto.

1.9 HERRAMIENTAS

- **Arduino**

Lenguaje de programación propio de Arduino Ver. 1.6.4 de 64 bits para programar

las instrucciones en el microcontrolador.

- **Proteus.**

Es software Proteus Design Suite 8.5, este permite hacer la simulación de todo el esquema del circuito a realizar haciendo pruebas de corrida

- **Android Studio**

Android es un sistema operativo móvil desarrollado por Google, basado en Kernel de Linux y otros softwares de código abierto. Fue diseñado para dispositivos móviles con pantalla táctil, como teléfonos inteligentes, tabletas, relojes inteligentes, automóviles y televisores.

- **El Arduino Mega 2560**

El Arduino Mega 2560 es una placa de desarrollo basada en el microcontrolador ATmega2560. Tiene 54 entradas/salidas digitales (de las cuales 15 pueden ser usadas como salidas PWM), 16 entradas analógicas, 4 UARTs, un cristal de 16Mhz, conexión USB, jack para alimentación DC, conector ICSP, y un botón de reseteo

- **Módulo HC-05**

El módulo de bluetooth HC-05 es el que ofrece una mejor relación de precio y características, ya que es un módulo Maestro-Esclavo, quiere decir que, además de recibir conexiones desde una PC o Tablet, también es capaz de generar conexiones hacia otros dispositivos bluetooth.

- **Protoboard (Placa de pruebas)**

La Protoboard, llamada en inglés breadboard, es una placa de pruebas en los que se pueden insertar elementos electrónicos y cables con los que se arman circuitos sin la necesidad de soldar ninguno de los componentes. Las Protoboards tienen orificios conectados entre sí por medio de pequeñas laminas metálicas.

- **Sensor Infrarrojo**

El infrarrojo es una luz compuesta de energía electromagnética. Sus ondas son más cortas que las de un microondas, pero más largas que los rayos de luz visibles.

El infrarrojo es invisible para el ojo humano. El cuerpo humano, recibe y emite naturalmente esta luz.

- **Servo Motor MG996R**

Este servo digital MG996R de alto torque cuenta con engranajes metálicos que resultan en un peso extra de 10 kg. Par de paro en un pequeño paquete. El MG996R es esencialmente una versión actualizada del Famoso servo MG995, y características mejoradas a prueba de golpes y un PCB rediseñado y IC Sistema de control que lo hacen mucho más preciso que su predecesor. El engranaje y el motor

También se han actualizado para mejorar el ancho de banda y el centrado muertos.

- **Sensor de Temperatura**

Los sensores de temperatura son dispositivos que transforman los cambios de temperatura en cambios en señales eléctricas que son procesados por equipo eléctrico o electrónico. Hay tres tipos de sensores de temperatura, los termistores, los RTD y los termopares

Capacitores

El capacitor es un dispositivo eléctrico que permite almacenar energía en forma de campo eléctrico. Es decir, es un dispositivo que almacena cargas en reposo o estáticas.

- **Leds**

Un LED (acrónimo del concepto inglés light-emitting diodo) es un diodo emisor de luz. En su interior hay un semiconductor que, al ser atravesado por una tensión continua, emite luz, lo que se conoce como electroluminiscencia.

- **Bocinas (Buzzers)**

El módulo zumbador KY- 006 consiste en un zumbador piezoeléctrico pasivo, que puede producir una gama de tonos de sonido dependiendo de la frecuencia de

entrada, este mismo puede generar tonos de entre 1.5 a 2.5 kHz al encenderlo y apagarlo a diferentes frecuencias, ya sea mediante retardos o por la modulación por ancho de pulsos.

- **Cables MM – HM**

Cables Dupont 20cm, ideales para prácticas y proyectos con Protoboard, PIC, Arduino, Raspberry, etc.

- **Reed Switch**

Este es pequeño dispositivo Reed Switch que tan solo mide 14 mm de largo, es conocido como "Reed Switch" (interruptor de lengüeta). Se caracteriza por ser un switch magnético, muy parecido a un relé, la diferencia que el campo magnético se aplica directamente al dispositivo y no mediante una bobina.

- **Cargador de 9 Voltios**

Alimentador Arduino para mantenerlo conectado a la corriente doméstica. Funciona de 110 a 220 V a 50 /60 Hz y proporciona hasta 9v y 1000 mA, ideal para alimentar tu Arduino de forma continua. Entrada: 220v - 50Hz, Salida: DC 9V - 1A (1000mA)

- **Cooler de PC**

El principio base de un 'cooler' es la utilización de agua en el proceso de enfriamiento del aire. El aire caliente de la habitación que entra al aparato pasa a través del medio de enfriamiento honeycomb.

- **Resistencias 220 Ω**

Una resistencia es un elemento pasivo que disipa energía en forma de calor según la ley de Joule. También establece una relación de proporcionalidad entre la intensidad de corriente que la atraviesa y la tensión medible entre sus extremos, relación conocida como ley de Ohm. En general, una resistencia podrá tener diferente comportamiento en función del tipo de corriente que circule por ella.

- **Modulo Bluetooth HC-05**

nos permite conectar nuestros proyectos con Arduino a un smartphone,

celular o PC de forma inalámbrica (**Bluetooth**), con la facilidad de operación de un puerto serial. ... La comunicación **Bluetooth** se da entre dos tipos de dispositivos: un maestro y un esclavo.

- **.Resistencias 10 K Ω .**

Las resistencias son uno de los tipos básicos de componentes electrónicos. Tienen dos terminales y un semiconductor, está formada por carbón y otros elementos resistivos.

- **Relay de cuatro salidas**

Este módulo de relevadores (relés) para conmutación de cargas de potencia. Los contactos de los relevadores están diseñados para conmutar cargas de hasta 10A y 250VAC (30VDC), aunque se recomienda usar niveles de tensión por debajo de estos límites.

1.10 LÍMITES Y ALCANCES

1.10.1 Límites

El presente sistema de control y seguridad al ser un prototipo de control interno y externo para una oficina no contempla un sistema completo de seguridad que incluya eventos como ser:

- Problemas en las instalaciones eléctricas dentro del hogar que pueden ser solucionados con otro tipo de sistema domótica.
- El presente sistema no reemplazara por completo a un sistema profesional de seguridad de alto costo, pero si brindara las funciones principales de control requeridas por el usuario.
- El límite de conectividad será de acuerdo a la recepción de datos por parte del Bluetooth.

1.10.2 Alcances

La domótica, así como sus aplicaciones dispone de una gran cantidad de dispositivos de entrada y salida. Por lo que el diseño y la implementación del sistema, propuestas por la presente Proyecto de grado se realizaran de acuerdo a las siguientes consideraciones:

- Se realizará la construcción de un sistema de componentes integrados de entrada y salida (analógico y digital).
- Se desarrollará una aplicación móvil para el control por comando de voz del sistema sobre la plataforma Arduino para el usuario.
- Se elaborará la construcción de un prototipo en una maqueta para las pruebas necesarias del sistema en funcionamiento.

1.11 APORTES

El aporte del presente trabajo de proyecto de grado está en la indagación, búsqueda, identificación, modelado y construcción de un sistema integrado, el cual está compuesto de elementos de hardware y software libre, enfocados a realizar la gestión de control de la Asociación De Fútbol El Alto a través de un dispositivo móvil que posea el usuario.

CAPÍTULO II

MARCO TEORICO

2. INTRODUCCIÓN

En este capítulo se describirá las características y definir los conceptos fundamentales sobre los que se basa, el sistema de control domótico basado en Arduino aplicación móvil.

2.1 DOMÓTICA

Para la (CIEC, 2012) la domótica es el conjunto de sistemas y tecnologías integradas que controlan y automatizan las diferentes instalaciones de un inmueble, contribuyendo gestión energética, confort, seguridad, comunicación y accesibilidad entre el usuario y el sistema.

El término domótica viene de la unión de las palabras domus (que significa casa en latín) y tica (de automática, palabra en griego, 'que funciona por sí sola') y por lo tanto se lo utiliza siempre que nos refiramos a viviendas unifamiliares. Cuando queremos hacer referencia a edificaciones compuestas por diferentes sectores habitacionales o comerciales el término que se utiliza es Inmótica. De manera más amplia, cuando llevamos el término al nivel de ciudades enteras, el término que se utiliza es Urbótica. (Balibrea, 2012)

Figura: 2.1 Domótica



Fuente: (domo. Educa)

2.2.1. Confort

Cuando su vivienda se adecúa por sí misma a sus necesidades mejora su calidad de

vida. La domótica juega un papel muy importante en este punto ya que posee el control de todo el inmueble, dicho control es usado fundamentalmente para, en primer lugar, hacer que las tareas repetitivas y rutinarias se realicen solas de forma automática, programar escenas para que el inmueble se adapte a las necesidades de cada persona.

El confort desde el punto de vista de la domótica es básicamente el control de dispositivos el cual se divide, por citar algunos ejemplos, en: control de iluminación, control de clima, control de aberturas, control de riego, control multimedia, generación de escenas, etc.

2.2.2 Seguridad

Consiste en una red de seguridad encargada de proteger las personas y los bienes, esta aplicación se apoya en dos pilares que son la prevención y la detección para la acción. Como la domótica tiene pleno conocimiento del estado de las puertas, ventanas y sensores dentro y fuera de la vivienda, puede, de una manera muy sencilla y eficiente, tomar control de esa información y poder, mediante la programación instalada, proteger todo el hogar. Esta es también una de las aplicaciones más importante de la domótica, ya que la figura de la persona que lleva adelante la responsabilidad global del hogar durante todo el día es cada vez menos frecuente en las familias actuales; en su reemplazo están los elementos que permiten saber lo que está pasando, sea de manera local o a distancia.

2.2.3 Comunicación

Esta aplicación puede que parezca poco importante, pero en realidad es la encargada o va de la mano con el resto de las aplicaciones ya que sin ella sería imposible conocer el estado y controlar los sistemas a distancia.

Lo que se consigue aquí es la posibilidad de conectarse con el hogar y dentro del mismo con la mayor cantidad de medios de comunicación disponibles, pudiendo de esta manera controlar la vivienda a distancia (tele gestión) y aumentar la interactividad entre las personas y el hogar.

2.2.4 Accesibilidad

En esta aplicación la domótica persigue posibilitar el acceso de cualquier persona a

cualquier entorno. La accesibilidad busca que en cualquier ambiente exista facilidad para la deambulaci3n, la aprehensi3n, la localizaci3n y la comunicaci3n. Como ejemplo podemos nombrar los sistemas de acci3n por voz, los cuales permiten ejecutar cualquier tipo de acci3n mediante un comando de voz espec3fico.

2.3 ARDUINO

Arduino es una plataforma electr3nica de hardware libre basada en una placa con un microcontrolador. Con software y hardware flexibles y f3ciles de utilizar, Arduino ha sido dise1ado para adaptarse a las necesidades de todo tipo de p3blico, desde aficionados, hasta expertos en rob3tica o equipos electr3nicos (Arduino, 2015).

Ante todo, y sobre todo es un microcontrolador, es decir un ordenador completo integrado en un chip, con su CPU, memoria de programa, memoria de datos y circuitos para el control de perif3ricos.

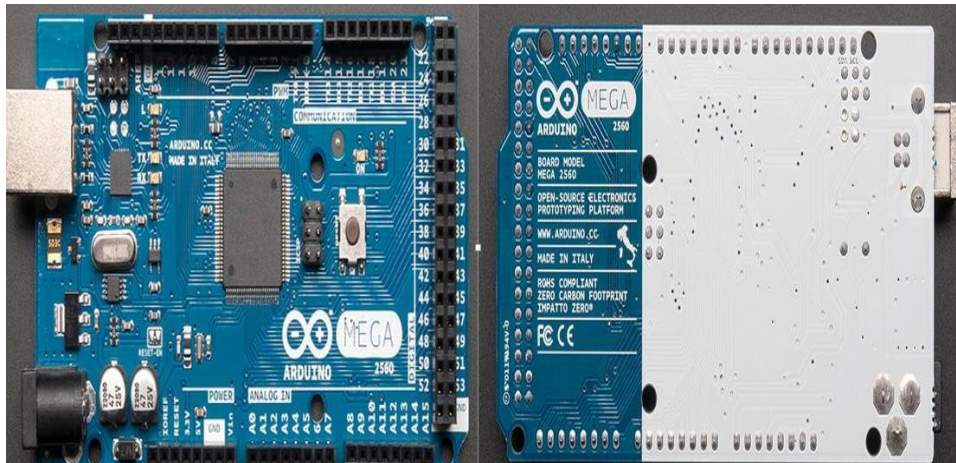
El microcontrolador necesita para su correcto funcionamiento, de algunos circuitos auxiliares y complementos tales como:

- La entrada de alimentaci3n
- El oscilador de trabajo
- Circuito de RESET
- La conexi3n USB
- Los accesos a las l3neas de entrada y salida, etc.

Tambi3n consta de un simple, pero completo, entorno de desarrollo, que nos permite interactuar con la plataforma de manera muy sencilla. Se puede definir por tanto como una sencilla herramienta de contribuci3n a la creaci3n de prototipos, entornos, u objetos interactivos destinados a proyectos multidisciplinarios y multitecnol3gia (Castro, 2013)

En la figura 2.1 se puede observar la placa Arduino, en este caso m3s espec3ficamente el Arduino Mega 2560. (Loureiro, 2017)

Figura 2.2. Arduino



Fuente: (UGE-ONE, 2016)

La placa Arduino está capacitada para incorporar hardware adicional, contiene una matriz de terminales en la que se puede añadir hardware de acuerdo al requerimiento del prototipo a desarrollar.

2.3.1 Características

El Arduino Mega está basado en el microcontrolador ATmega2560. Tiene 54 pines de entradas/salidas digitales (14 de las cuales pueden ser utilizadas como salidas PWM), 16 entradas análogas, 4 UARTs (puertos serial por hardware), cristal oscilador de 16 Mhz, conexión USB, jack de alimentación, conector ICSP y botón de reset. Incorpora todo lo necesario para que el microcontrolador trabaje; simplemente conéctalo a tu PC por medio de un cable USB o con una fuente de alimentación externa. El Arduino Mega es compatible con la mayoría de los shields diseñados para Arduino Duemilanove, diecimila o UNO (Arduino, 2016).

Esta nueva versión de Arduino Mega 2560 adicionalmente a todas las características de su sucesor, el Arduino Mega ahora utiliza un microcontrolador ATmega8U2 en vez del chip FTDI. Esto permite mayores velocidades de transmisión por su puerto USB y no requiere drivers para Linux o MAC (archivo inf es necesario para Windows) además ahora cuenta con la capacidad de ser reconocido por el PC como un teclado, mouse, joystick, etc.

En la Tabla 2.1 podemos ver un resumen de las características técnicas más relevantes del Arduino Mega 2560.

Tabla 2.1: Características Técnicas de Arduino Mega 2560

<i>Características Técnicas de Arduino Mega 2560</i>	
Elemento	Información
Microcontrolador	Atmega2560
Voltaje de operación	5 V
Voltaje de entrada (Recomendado)	7 – 12 V
Voltaje de entrada (Límite)	6 – 20 V
Pines para entrada- salida digital.	54 (de los cuales 15 proporcionan salida PWM)
Pines de entrada analógica.	16
Corriente continua por pin IO	20 mA
Corriente continua en el pin 3.3 V	50 mA
Memoria Flash	256 KB, 8 KB utilizado por el gestor de arranque
SRAM	8 KB
EEPROM	4 KB
Velocidad de reloj	16 MHz
LED_BUILTIN	8
Longitud	101.52 mm
Anchura	53,3 mm
Peso	37 g

Nota: Fuente: (Arduino, 2016)

2.3.2 Alimentación de un Arduino

El Arduino Mega puede ser alimentado vía la conexión USB o con una fuente de alimentación externa. El origen de la alimentación se selecciona automáticamente.

Las fuentes de alimentación externas (no USB) pueden ser tanto un transformador o una batería. El transformador se puede conectar usando un conector macho de 2.1mm con centro positivo en el conector hembra de la placa. Los cables de la batería pueden conectarse a los pines Gnd y Vin en los conectores de alimentación (POWER)

La placa puede trabajar con una alimentación externa de entre 6 a 20 voltios. Si el voltaje suministrado es inferior a 7V, el pin de 5V puede proporcionar menos de 5 Voltios y la placa puede volverse inestable; si se usan más de 12V los reguladores de voltaje se pueden sobrecalentar y dañar la placa. El rango recomendado es de 7 a 12 voltios.

Los pines de alimentación son los siguientes:

- VIN. La entrada de voltaje a la placa Arduino cuando se está usando una fuente externa de alimentación (en opuesto a los 5 voltios de la conexión USB). Se puede proporcionar voltaje a través de este pin, o, si se está alimentando a través de la conexión de 2.1 mm acceder a ella a través de este pin.
- 5V. La fuente de voltaje estabilizado usado para alimentar el microcontrolador y otros componentes de la placa. Esta puede provenir de VIN a través de un regulador integrado en la placa, o proporcionada directamente por el USB u otra fuente estabilizada de 5V.
- 3V3. Una fuente de voltaje de 3.3 voltios generada por un regulador integrado en la placa. La corriente máxima soportada 50mA.
- GND. Pines de toma de tierra.

2.3.3 Memoria

El ATmega2560 tiene 256KB de memoria flash para almacenar código (8KB son usados para el arranque del sistema). El ATmega2560 tiene 8 KB de memoria SRAM y 4KB de EEPROM, a la cual se puede acceder para leer o escribir con la librería EEPROM.

2.3.4 Entradas y Salidas

Cada uno de los 54 pines digitales en el Mega pueden utilizarse como entradas o como salidas usando las funciones `pinMode()`, `digitalWrite()`, y `digitalRead()`. Las E/S operan a 5 voltios. Cada pin puede proporcionar o recibir una intensidad máxima de 40mA y tiene una resistencia interna de pull-up (desconectada por defecto) de 20-50kOhms. Además, algunos pines tienen funciones especializadas:

Serie: 0 (RX) y 1 (TX), Serie 1: 19 (RX) y 18 (TX); Serie 2: 17 (RX) y 16 (TX); Serie 3: 15 (RX) y

14 (TX). Usados para recibir (RX) transmitir (TX) datos a través de puerto serie TTL. Los pines Serie: 0 (RX) y 1 (TX) están conectados a los pines correspondientes del chip FTDI USB-to-TTL.

Interrupciones Externas: 2 (interrupción 0), 3 (interrupción 1), 18 (interrupción 5), 19 (interrupción

, 20 (interrupción 3), y 21 (interrupción 2). Estos pines se pueden configurar para lanzar una interrupción en un valor LOW(0V), en flancos de subida o bajada (cambio de LOW a HIGH(5V) o viceversa), o en cambios de valor. Ver la función `attachInterrupt()` para más detalles.

PWM: de 0 a 13. Proporciona una salida PWM (Pulse Wave Modulation, modulación de onda por pulsos) de 8 bits de resolución (valores de 0 a 255) a través de la función `analogWrite ()`.

SPI: 50 (SS), 51 (MOSI), 52 (MISO), 53 (SCK). Estos pines proporcionan comunicación SPI, usando la librería SPI.

LED: 13. Hay un LED integrado en la placa conectado al pin digital 13, cuando este pin tiene un valor HIGH(5V) el LED se enciende y cuando este tiene un valor LOW(0V) este se apaga

El Mega tiene 16 entradas analógicas, y cada una de ellas proporciona una resolución de 10bits (1024 valores). Por defecto se mide desde 0V a 5V, aunque es posible cambiar la cota superior de este rango usando el pin AREF y la función `analogReference ()`.

I2C: 20 (SDA) y 21 (SCL). Soporte para el protocolo de comunicaciones I2C (TWI) usando la librería Wire.

AREF. Voltaje de referencia para la entrada analógicas. Usado por `analogReference ()`.

Reset. Suministrar un valor LOW (0V) para reiniciar el microcontrolador. Típicamente usado para añadir un botón de reset a los shields que no dejan acceso a este botón en la placa.

2.3.5 Comunicaciones

EL Arduino Mega 2560 facilita en varios aspectos la comunicación con la PC. El ATmega2560 proporciona cuatro puertos de comunicación vía serie UART TTL (5V).

Un ATmega16U2 integrado en la placa canaliza esta comunicación serie a través del puerto USB y los drivers (incluidos en el software de Arduino) proporcionan un puerto serie virtual en el ordenador. El software incluye un monitor de puerto serie que permite enviar y recibir información textual de la placa Arduino. Los LEDs RX y TX de la placa parpadearán cuando se detecte comunicación transmitida través de la conexión USB (no parpadearán si se usa la comunicación serie a través de los pines 0 y 1).

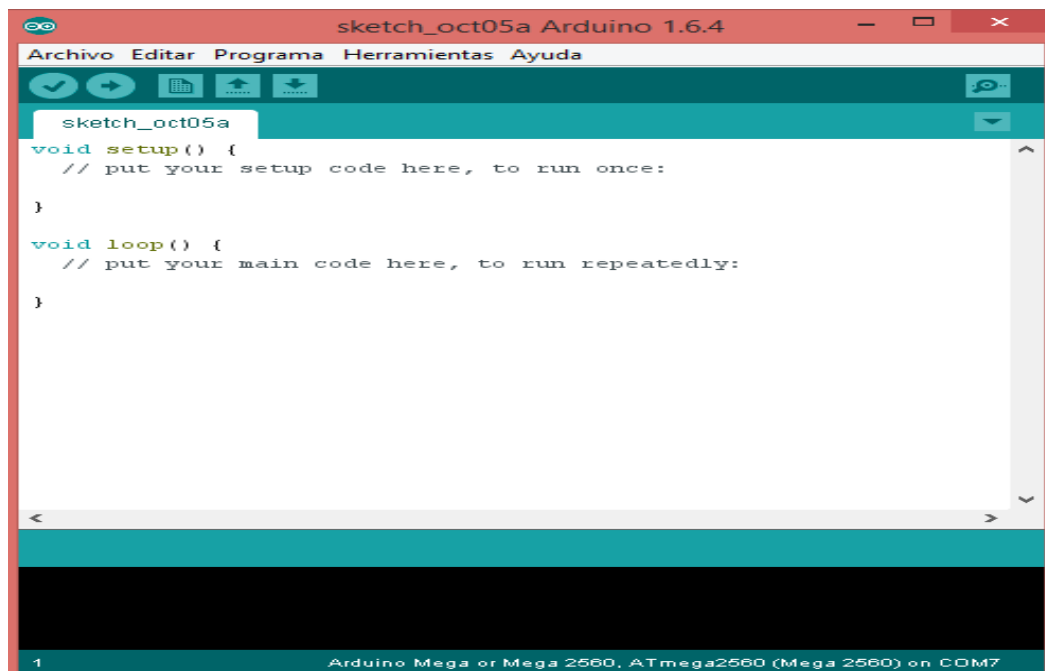
2.3.6 Programación.

El ATmega2560 en el Arduino Mega viene precargado con un gestor de arranque (bootloader) que permite cargar nuevo código sin necesidad de un programador por hardware externo. Se comunica utilizando el protocolo STK500 original (referencia, archivo de cabecera C).

También puede evitarse el gestor de arranque y programar directamente el microcontrolador a través del puerto ICSP (In Circuit Serial Programming)

En la figura 2.2 mostraremos el entorno de desarrollo de Arduino. (Gutiérrez, 2007)

Figura 2. 3: Entorno de desarrollo de Arduino



Fuente: (Elaboración Propia)

2.3.7 Reinicio automático por Software

En vez de necesitar reiniciar presionando físicamente el botón de reset antes de cargar, el Arduino Mega está diseñado de manera que es posible reiniciar por software desde el ordenador donde esté conectado. Una de las líneas de control de flujo (DTR) del ATmega16U2 está conectada a la línea de reinicio del ATmega2560 a través de un condensador de 100 nano faradios. Cuando la línea se pone a LOW(0V), la línea de reinicio también se pone a LOW el tiempo suficiente para reiniciar el chip. El software de Arduino utiliza esta característica para permitir cargar los sketches con solo apretar un botón del entorno. Dado que el gestor de arranque tiene un lapso de tiempo para ello, la activación del DTR y la carga del sketch se coordinan perfectamente.

Esta configuración tiene otras implicaciones. Cuando el Mega se conecta a un ordenador con Mac OS X o Linux, esto reinicia la placa cada vez que se realiza una conexión desde el software (vía USB). El medio segundo aproximadamente posterior, el gestor de arranque se está ejecutando. A pesar de estar programado para ignorar datos mal formateados (ej. cualquier cosa que la carga de un programa nuevo) intercepta los primeros bytes que se envían a la placa justo después de que se abra la conexión. Si un sketch ejecutándose en la placa recibe algún tipo de configuración inicial o otro tipo de información al inicio del programa, debe asegurarse de que el software con el cual se comunica espera un segundo después de abrir la conexión antes de enviar los datos.

El Mega contiene una pista que puede ser cortada para deshabilitar el auto reset. Las terminaciones a cada lado pueden ser soldadas entre ellas para rehabilitarlo. Están etiquetadas con "RESET-EN". También se puede deshabilitar el auto reset conectando una resistencia de 110 ohms desde el pin 5V al pin de reset.

2.3.7 Protección contra sobre corrientes en USB

El Arduino Mega tiene un multifusible reinicializable que protege la conexión USB del PC de cortocircuitos y sobretensiones. Aparte de que la mayoría de ordenadores proporcionan su propia protección interna, el fusible proporciona una capa extra de protección. Si más de 500mA son detectados en el puerto USB, el fusible automáticamente corta la conexión hasta que la sobretensión desaparezca.

2.3.8 Características físicas y compatibilidad de Shields

La longitud y amplitud máxima de la placa Mega 2560 son de 4 y 2.1 pulgadas respectivamente, con el conector USB y la conexión de alimentación sobresaliendo de estas dimensiones. Tres agujeros para fijación con tornillos permiten colocar la placa en superficies y cajas. Tener en cuenta que la distancia entre los pines digitales 7 y 8 es 160 mil (0,16"), no es múltiplo de la separación de 100 mil entre los otros pines.

2.4 Seguridad

La seguridad personal y de las propiedades particulares, preocupa desde siempre a todos los propietarios de una vivienda y en procurársela gastan elevadas sumas de dinero.

Un caso particular que últimamente está cobrando gran importancia es la protección del hogar, no solo con los elementos tradicionales (pasivos) como son rejas y puertas blindadas, sino con otros más sofisticados (activos), debido a que los sistemas profesionales que se venían empleando en la vigilancia de bancos, empresas, almacenes, etc., han reducido su tamaño, su precio y ya son accesibles para todos.

Son numerosos los anuncios de empresas de seguridad que ofrecen vigilancia y alarma las 24 horas del día, 7 días a la semana, instalando el equipo completo, por una módica cantidad y una cuota mensual de mantenimiento, que atraen a numerosos clientes. Ante la enorme proliferación de urbanizaciones, negocios, apartamentos en la playa, etc., los servicios públicos de policía no son suficientes para garantizar la vigilancia en todo momento y lugar, por lo que se hace necesario buscar medidas complementarias, como puede ser instalar sofisticados sistemas de seguridad o contratar un buen seguro que cubra los daños en caso de sufrir algún percance, ya que estamos expuestos a ellos.

2.4.1 Gestión de Seguridad en el hogar

La gestión de la seguridad debe contemplar tanto la seguridad personal como la seguridad del patrimonio; además, un sistema de seguridad debe considerar diferentes funciones que aseguren las tres áreas básicas de la seguridad: la prevención que se da antes que se produzca el ataque para evitarlo, la alarma en el

momento del ataque y la reacción una vez que se ha producido el ataque para considerar sus efectos. En un hogar automatizado como se observa en la Figura 2.3 existe una variedad de dispositivos que se puede implementar para mejorar la calidad de vida. (Pesantes, 2018)

Figura 2.4: Seguridad domótica en el Hogar.



Fuente: (Serconint, 2016)

Los sistemas domóticos de seguridad para el hogar suelen combinar varias funciones, así, además de las propias anti intrusión suelen tener otras para detectar humo, gas, fuego, inundaciones llamadas alarmas técnicas, las que son ligadas al confort como la temperatura, iluminación, comunicaciones o lo que se vienen a denominar alarmas médicas para la atención a distancia de personas enfermas o ancianas que monitorizan algunos de sus parámetros biométricos o permiten el aviso en caso de accidente, como puede ser una caída.

Para el diseño eficaz de un sistema de alarmas hay que tener muy claro qué es lo que se desea proteger, contra qué o quién se desea proteger y con qué grado de seguridad, además de lo que cuesta su instalación y mantenimiento, para que se dé un equilibrio entre unos y otros factores. Llega un momento en que no es interesante invertir más en seguridad (López, 2007).

2.4.2 Vigilancia Interna y Externa

Centrándonos en el hogar, tenemos dos zonas bien diferenciadas: una el interior,

donde el grado de seguridad ha de ser máximo ya que es la zona donde pasamos la mayor parte del tiempo y donde se duerme habitualmente, un momento especialmente peligroso; y otra, el exterior, en donde se permite un grado menor ya que al ser más difícil de controlar, no existen muros y techos como en la casa y los medios que requieren son más sofisticados por tanto son más caros. También habrá que distinguir entre lo que es una vivienda en un bloque de pisos, en donde casi la única posibilidad de entrada es por la puerta principal, por lo que una buena puerta acorazada viene a ser suficiente, o una vivienda individual, de varias plantas, en una urbanización o aislada, en la que las posibilidades de entrada son más amplias y la zona a cubrir mayor.

Por tanto, dentro del sistema de vigilancia se pueden definir diferentes niveles, en función del espacio a proteger, que son: perimetral, periférica, volumétrica, control de accesos y vigilancia de agresión (López, 2007).

2.4.3 Seguridad Perimetral

Los sensores para seguridad perimetral detectan a los intrusos al momento que entran a una zona protegida y antes de que tengan acceso al interior de la vivienda como se muestra en la Figura 2.4. Dicho sistema está preparado para condiciones ambientales adversas como temperaturas extremas, lluvia, nieve, animales, viento, efectos sísmicos, terreno y tráfico, entre otras. Por supuesto existe la posibilidad de ocultar las barreras perimetrales o sensores de exterior para obtener una prevención ideal. (Huidoboro, 2010).

Figura 2. 5: Seguridad perimetral en una vivienda

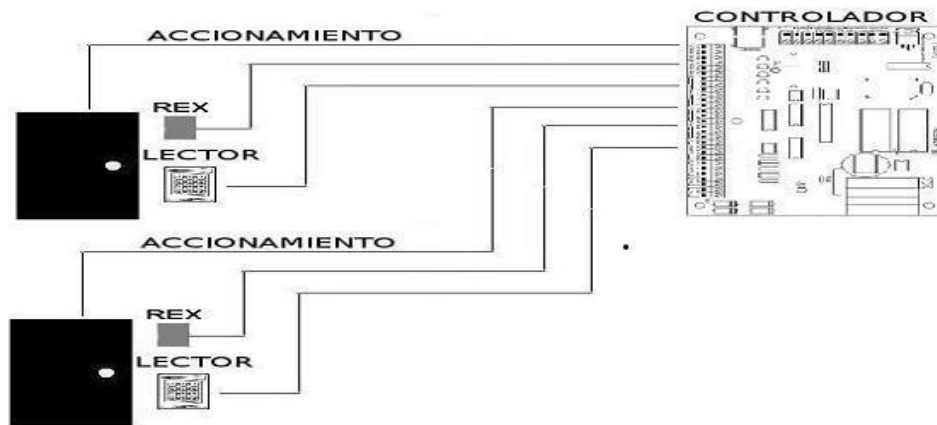


Fuente: (securemur, 2015)

2.4.5 Control de Acceso

Control sobre la apertura de los diferentes accesos a una instalación, pudiendo permitir la identificación de las personas que entran y salen de las zonas protegidas, por ejemplo, mediante la introducción de una clave de acceso como se ve en la Figura 2.5. Estos sistemas se basan en lectores de tarjetas magnéticas o teclados de acceso y permiten comprobar el horario de apertura y cierre de la empresa y por quién fueron realizados, así como controlar los movimientos del servicio doméstico del hogar, etc.

Figura 2. 6: Controlador de acceso



Fuente: [Scielo, 2013]

En una vivienda individual, el único control de acceso que suele hacerse es a la propia central de alarma, por lo que el propietario dispone de un código de activación y desactivación, además de otro falso (por ejemplo, el número anterior o el siguiente) por si es coaccionado por un intruso, en cuyo caso, aparentemente se desactiva la alarma, pero se genera una alarma silenciosa hacia la central de vigilancia (Huidoboro, 2010).

2.4.6 Alarma de Agresión

También denominados sistemas de detección de pánico, permiten el aviso de una agresión a personas o instalaciones. Para su utilización es necesaria la acción manual de la persona que se siente agredida, que suele hacerse mediante un pulsador que genera una alarma silenciosa o acústica. Sirve cuando nos bajamos del

coche y nos dirigimos a abrir la puerta de casa, cuando estamos por el jardín y en cualquier habitación (Huidoboro, 2010).

2.4.7 Centrales de Alarma

Todas las funciones que realiza un sistema de vigilancia se centralizan en una central de alarmas de las que existen numerosos modelos en el mercado. Esta central gestiona la salida de los numerosos detectores, permite el manejo a elección del usuario del funcionamiento del sistema.

(zonas a controlar, horarios, niveles de sensibilidad), y genera las acciones pertinentes de alarma óptica y/o acústica, aviso silencioso al usuario o aviso a una central receptora de alarmas remotas, ya que por ley no está permitido que se avise directamente a las fuerzas públicas de seguridad y es necesario un filtro intermedio que discrimine las alarmas, a cargo de empresas privadas de seguridad.

Por medio de Internet podemos tener acceso a la pasarela residencial, utilizando el protocolo normalizado IP, y su aplicación como red de comunicaciones la seguridad es obvia, como un medio muy sencillo y hasta cierto punto económico de acceso. La central de alarma puede ser un PC o un teléfono móvil con tecnología Android conectado permanentemente a la Red y el usuario puede acceder a él a través de una dirección Web, en cualquier momento y desde cualquier sitio (López, 2007).

2.5 SENSORES

Los sensores son los dispositivos encargados de recoger la información de los diferentes parámetros que controla el sistema de control centralizado como la temperatura ambiente, la existencia de un escape de agua o gas, la presencia de un intruso, etc., y enviársela a la central para que ejecute automáticamente las tareas programadas.

Los hay de diversos tipos: gas, temperatura, agua, humedad, luz, movimiento, rotura, etc., y están distribuidos por todo el domicilio, según la zona a vigilar/proteger son más adecuados unos sistemas que otros, y lo común suele ser utilizar una combinación de varios de ellos, cuantos más, mejor.

En general, los sensores que se utilizan, para la seguridad en el hogar, se clasifican en cinco tipos: de contacto, infrarrojos, ultrasonidos, vibración y microondas, aunque

existen otros para aplicaciones especiales.

Los sensores de contacto son los más simples y los primeros que se empezaron a utilizar. Se basan en la apertura o cierre de un circuito al actuar sobre el sensor, que puede ser un sencillo interruptor eléctrico que se activa al abrir una puerta o un sensor magnético que no necesita contacto entre las partes en movimiento.

Estos detectores son muy robustos y económicos, no requieren apenas mantenimiento y se suelen utilizar para la protección periférica. Permiten activar la carga directamente, como puede ser una sirena o una lámpara (Millan, 2004), existe una variedad de sensores que se pueden utilizar en un sistema domótico.

Figura 2. 7: Sensores Domóticos figura 2.7



Fuente: (Proyecta, 2015)

2.5.1 Sensores de Infrarrojos

Los sensores de infrarrojos, al contrario que los anteriores, son sistemas activos, que emiten radiación no visible y que basan su efectividad en la creación de una barrera invisible que al ser rota activa la alarma.

El sistema puede ser de barrera, con un emisor y un receptor separados unos cuantos metros, o “réflex”, en el que el emisor y el receptor se encuentran juntos y el haz de luz se refleja en un espejo enfrente al mismo, siendo en este caso el alcance menor. La ventaja de este segundo sistema es que toda la electrónica esté en el mismo dispositivo. Los sistemas réflex polarizados diferencian la luz directa de la reflejada, tienen un alcance en torno a unos 5 metros y son muy seguros, evitando las falsas alarmas (Huidoboro, 2010).

2.5.2 Sensores de Ultrasonidos

Los sensores de ultrasonidos como se ve en la Figura 2.8, permiten detectar movimiento, basándose en el efecto Doppler, el mismo que utiliza el radar de vigilancia en carretera, que hace que varíe la frecuencia de la onda al rebotar en el objeto en movimiento. Emiten ultrasonidos y tiene un alcance de muy pocos metros, utilizándose en la vigilancia volumétrica. Podemos comprobar su funcionamiento estando quietos y moviéndonos: veremos un LED apagarse y encenderse (Huidoboro, 2010).

Figura 2. 8: Sensor de Ultrasonido



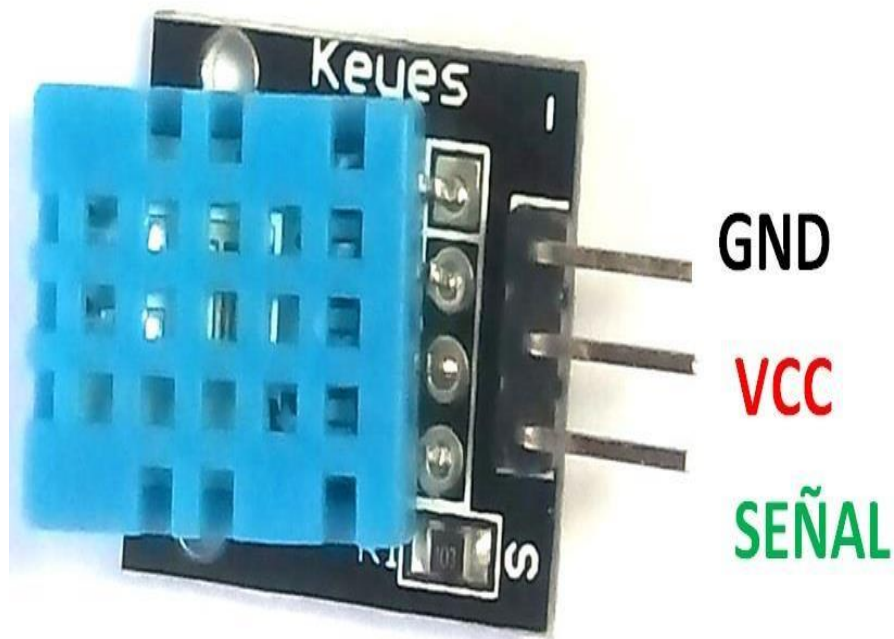
Fuente: (Kurtsik, 2020)

2.5.3 Sensores de Temperatura

Los sensores de Temperatura como se muestra en la Figura 2.9, El DHT11 consigue medir temperatura y humedad, recibiendo la información únicamente a través del pin digital 2. Es capaz de leer un porcentaje de humedad relativa de entre el 20 y el 90% con un error de $\pm 5\%$. Además, lee temperaturas entre los 0 y los 50 °C, con un error de 2°C. Necesita una alimentación de entre 3,3 y 5,5V de corriente continua para su funcionamiento.

En cuanto a sus conexiones, de izquierda a derecha se conectan el pin digital, 5V y GND. Para establecer la conexión de datos, se adopta un protocolo para comunicar ambos dispositivos. (Philco, 2018)

Figura 2. 9: Sensor de Temperatura

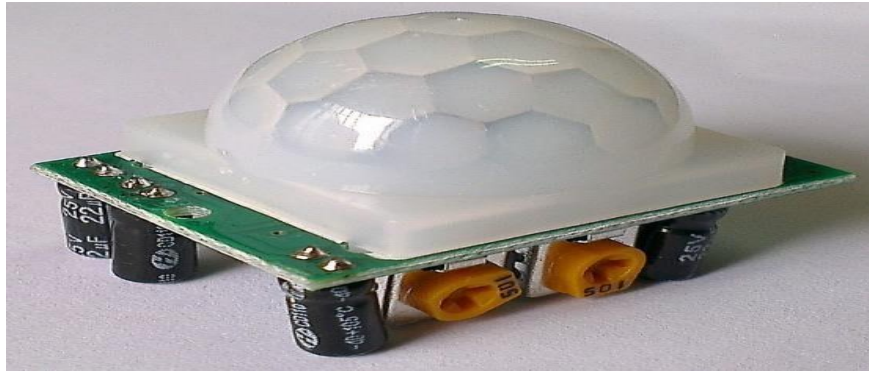


Fuente: (K-electrónica, 2014)

2.5.4 Sensor de Movimiento

Los sensores de movimiento son aparatos basados en la tecnología de los rayos infrarrojos o las ondas ultrasónicas para poder “mapear” o captar en tiempo real los movimientos que se generan en un espacio determinado. Estos sensores de movimiento como se puede ver en la Figura 2.10, relacionado sobre todo a cámaras de seguridad, puertas en almacenes y centros comerciales, etc., son uno de los dispositivos más reconocidos e importantes dentro de la seguridad electrónica, que tanto ha apostado por, sobre todo, dos aspectos fundamentales: el tamaño y la funcionalidad de cada uno de los equipos que usan durante el proceso.

Figura 2. 10: Sensor de Movimiento PIR



Fuente: (Ardobot, 2015)

Pero los sensores también están siendo adaptados a todo tipo de electrodomésticos, haciendo mucho más eficaz los niveles de protección o de vigilancia a los que un recinto puede llegar. Se ven sensores de movimiento ya instalados en algunas lámparas corrientes, por ejemplo, o hasta en relojes despertadores, siendo esta la última generación de sensores de movimiento que funcionan por intermedio de ondas ultrasónicas. Porque aquellos que operan mediante rayos infrarrojos resultan ser mucho más sofisticados, y se usan sobre todo en lugares que necesitan de un alto nivel de protección como por ejemplo la reserva federal de un banco.

Esta clase de sensores tienen la capacidad, así mismo, de poder dibujar a escala una representación del movimiento que puede darse por distintos puntos de unión, como si se tratara del mapa de una constelación. Por eso, los sensores de rayos infrarrojos dependiendo del caso, también vienen programados con algún auxiliar gráfico con los que complementan, gráficamente, sus acciones principales. Este es el tipo de dispositivo que es utilizado, para citar un caso, en la realización de películas de animación digital, donde se analizan los movimientos característicos de los distintos seres haciendo una imagen computarizada de ellos (Sensores, 2011).

2.6 SERVOMOTORES

Un servomotor o también llamado servo, es un dispositivo similar a un motor de corriente continua, que tiene la capacidad de ubicarse en cualquier posición dentro de su rango de operación, y mantenerse estable en dicha posición. Está conformado por un motor, una caja reductora y un circuito de control. Los servos se utilizan

frecuentemente en sistemas de radio control y en robótica, pero su uso no está limitado a estos.

Es posible modificar un servomotor para obtener un motor de corriente continua que, si bien ya no tiene la capacidad de control del servo, conserva la fuerza, velocidad y baja inercia que caracteriza a estos dispositivos.

Un servo normal como se puede apreciar en la Figura 2.11, tiene 3kg por cm. de torque que es bastante fuerte para su tamaño.

También potencia proporcional para cargas mecánicas. Un servo, por consiguiente, no consume mucha energía.

La corriente que requiere depende del tamaño del servo. Normalmente el fabricante indica cual es la corriente que consume. Eso no significa mucho si todos los servos van a estar moviéndose

todo el tiempo. La corriente depende principalmente del par, y puede exceder un amperio si el servo está enclavado (Ramires, 2019)

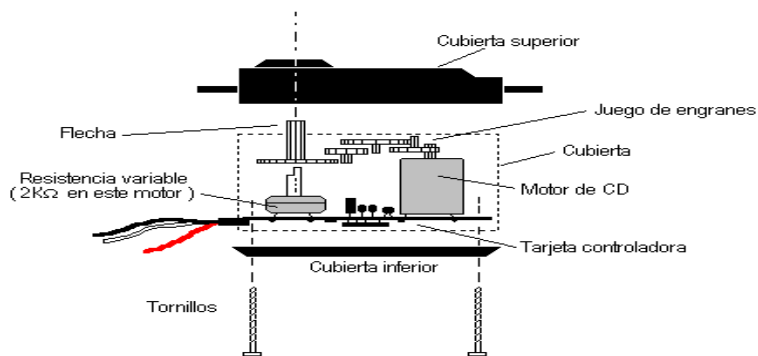
Figura 2. 11: Servomotor de giro de 0° - 180°



Fuente: (Electrónicos, 2013)

2.6.1 Partes de un servomotor

Figura 2. 12: Estructura interna de un Servomotor



Fuente: (Infolaser, 2002)

Un servomotor está compuesto internamente por un conjunto de elementos que se pueden apreciar en la Figura 2.12.

- **Motor de corriente continua**

Es el elemento que le brinda movilidad al servo. Cuando se aplica un potencial a sus dos terminales, este motor gira en un sentido a su velocidad máxima. Si el voltaje aplicado sus dos terminales es inverso, el sentido de giro también se invierte.

Código básico de un servomotor.

2.7 TECNOLOGÍAS DE SOFTWARE

- **IDE Arduino:** Lenguaje de programación propio de Arduino Ver. 1.6.4 de 64 bits para programar las instrucciones en el microcontrolador.
- **Proteus:** Es software Proteus Design Suite 8.5, este nos permite hacer la simulación de todo el esquema del circuito a realizar haciendo pruebas de corrida.
- **Android Kotlin:** Kotlin es un nuevo idioma (a veces conocido como Swift para Android), desarrollado por el equipo de JetBrains y ahora está en su versión 1.0.2. Lo que lo hace útil en el desarrollo de Android es que compila a bytecode JVM, y también se puede compilar con JavaScript. Es totalmente compatible con Java y el código de Kotlin puede ser simplemente convertido a

código Java y viceversa (hay un plugin de JetBrains). Esto significa que Kotlin puede usar cualquier marco, biblioteca, etc., escrito en Java. En Android, se integra por Gradle. Si tienes una aplicación Android existente y quieres implementar una nueva función en Kotlin sin volver a escribir la aplicación completa, solo empieza a escribir en Kotlin y funcionará.

2.8 METODOLOGÍA

2.8.1 Modelo en V

El modelo en V fue utilizado por primera vez en Alemania. Y apareció con su primera versión en agosto de 1992, el modelo en V fue proyectado en junio de 1997 y es prescrito en febrero del 2006 bajo la asignación de la V XT se dice que es el desarrollo para los sistemas de información y el diseño de ejecución de proyectos. El modelo en V es una variación del modelo en cascada que muestra cómo se relacionan las actividades de prueba con el análisis y el diseño. Como se muestra en la Figura 2.3, la codificación forma el vértice de la V, con el análisis y el diseño a la izquierda y las pruebas y el mantenimiento a la derecha. La unión mediante líneas discontinuas entre las fases de la parte izquierda y las pruebas de la derecha representa una doble información. Por un lado, sirve para indicar en qué fase de desarrollo se deben definir las pruebas correspondientes. Por otro sirve para saber a qué fase de desarrollo hay que volver si se encuentran fallos en las pruebas correspondientes. Por lo tanto, el modelo en V hace más explícita parte de las iteraciones y repeticiones de trabajo que están ocultas en el modelo en cascada. Mientras el foco del modelo en cascada se sitúa en los documentos y productos desarrollados, el modelo en V se centra en las actividades y la corrección. Es un procedimiento el cual sirve para el desarrollo de productos el cual es utilizado para los proyectos de Administración también se dice que es una representación básica para el desarrollo de sistemas. Es una secuencia de pasos para el desarrollo del ciclo de vida de un proyecto el cual se describe todas las actividades que se debe seguirse para el desarrollo de dicho proyecto, una breve descripción lo que significa la V. 16 El lado izquierdo de la V significa la representación las necesidades, y la creación de las especificaciones del sistema. El lado derecho de la V significa la

integración de todos los pasos y su verificación. El método en V significa “verificación” y “validación”, es muy similar al método de cascada ya que es muy rígido y contiene una gran cantidad de iteraciones. La verificación forma parte de la comparación entre los requisitos básicos y los refinamientos sucesivos que descienden de él, el diseño del producto, diseño detallado, el código, la base de datos y documentación a fin de mantener estas mejoras con los requisitos básicos aplicables. Por lo tanto, las actividades de verificación se inician en la fase de diseño de producto y concluir con la prueba de aceptación. No dan lugar los cambios en los requisitos básicos, y solo a los cambios en mejoras que descienden de él. El desarrollo de software de software de ciclo de vida se refiere al proceso de desarrollo de software completo desde el inicio de un proyecto hasta su conclusión. Es un modelo de ciclo de vida desde el punto de vista de aseguramiento de calidad. El modelo V es un modelo definido después del modelo de cascada, lo cual implica que posee características que el modelo cascada carece, por ejemplo, en el modelo en V se tiene la facilidad de que en el momento en el cual se está realizando una fase es posible realizar la documentación para las pruebas que se realizan más adentro. Las fases de modelo V son similares a la fase de cascada hay una gran diferencia, en el modelo en V en vez de ir debajo de forma lineal los pasos del proceso son doblados hacia arriba en la fase de codificación, para formar la típica forma de V. (Chavez J. G., 2015)

- **Fase 1.** Definición de Especificaciones: Se deben definir y documentar los diferentes requisitos del sistema a desarrollar, identificando los valores numéricos más concretos posibles. Entre ellos debe estar la especificación del nivel de integridad, o SIL, en caso de ser requerido. Especificación estructurada utilizando diferentes técnicas de diagramas para modelar el sistema nuevo.
- **Fase 2.** Diseño Global: También llamado diseño de alto nivel. Su objetivo es obtener un diseño y visión general del sistema. Establecer un conjunto de módulos entre ellos, desplegando la especificación obtenida en la fase de análisis facilitando la tarea de codificación y los modelos lógicos de los datos físicos.

- **Fase 3.** Diseño en Detalle: Consiste en detallar cada bloque de la fase anterior. Cada módulo como resultado de la fase anterior es traducido a la herramienta o lenguaje apropiado.
- **Fase 4.** Implementación: Es la fase en la que se materializa el diseño en detalle.
- **Fase 5.** Test Unitario: En esta fase se verifica cada módulo HW y SW de forma unitaria, comprobando su funcionamiento adecuado. Es la verificación del correcto funcionamiento de cada módulo y de todo el sistema una vez que ha sido integrado, diseñar errores en la codificación.
- **Fase 6.** Integración: En esta fase se integran los distintos módulos que forman el sistema. Como en el caso anterior, ha de generarse un documento de pruebas. 20 por una parte, se debe comprobar en todo el sistema el funcionamiento correcto, y por otra, en caso de tratarse con un sistema tolerante a fallos, debe verificarse que ante la presencia de un fallo persiste el funcionamiento correcto. Se comprueba el cumplimiento de los requisitos establecidos.
- **Fase 7.** Test Operacional Del Sistema: Se realizan las últimas pruebas, pero sobre un escenario real, en su ubicación final, anotando una vez más las pruebas realizadas y los resultados obtenidos. Garantizar el mantenimiento del sistema, corrección de errores detectados en la fase, aplicación de los sistemas en nuevos entornos.

2.8.1 Mobile-D

Consta de cinco fases: exploración, iniciación, producción, estabilización y prueba del sistema. Cada una de estas fases tiene un número de etapas, tareas y prácticas asociadas.

2.8.2 Fases de la Metodología

La metodología cuenta con 5 fases por las cuales pasa el producto a realizarse, la línea de producción empieza con la fase de exploración, después pasa a la fase de Iniciación, luego pasa a la fase de producto posteriormente a la fase de estabilización y la fase de pruebas. (Agile, 2008)

- **Exploración**

Se dedica a la planificación y a los conceptos básicos del proyecto. Es diferente del resto de fases (Mobile-D, 2015).

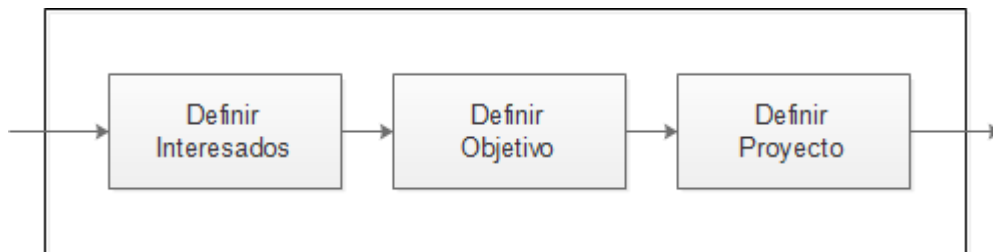
La fase de exploración es una fase importante para establecer las bases para la implementación controlada de productos de software, por ejemplo, las cuestiones relacionadas con la arquitectura del producto, proceso de desarrollo y la selección del entorno.

Los objetivos de esta fase son:

Establecer los grupos de interesados.

- Definir y aceptar los objetivos y metas para el desarrollo del proyecto.
- Realizar la planificación estimada acerca del entorno del proyecto, personal necesario, y procesos a realizar.

Figura 2. 13 fase de Exploración



Proceso Fuente: (Mobile-D, 2015)

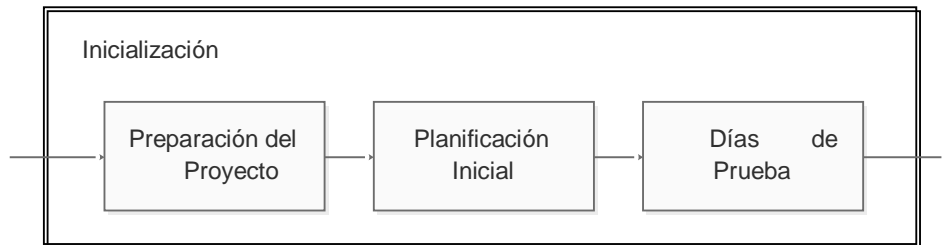
- **Inicialización.**

Se preparan e identifican todos los recursos necesarios. Se establece el entorno técnico.

El objetivo de esta fase es:

- Obtener una comprensión global acerca del producto, destinado al equipo de trabajo, basado en los requerimientos iniciales (Ramirez, 2012).
- Preparar los recursos físicos, técnicos y humanos, así también la comunicación con el cliente, proyectar los planes y todos los asuntos críticos de desarrollo a fin de que todos ellos estén listos para la implementación de los requisitos seleccionados por el cliente durante las próximas fases del proyecto (Ramirez, 2012).

Figura 2. 14 fase de Inicialización



Proceso Fuente: (Mobile-D, 2015)

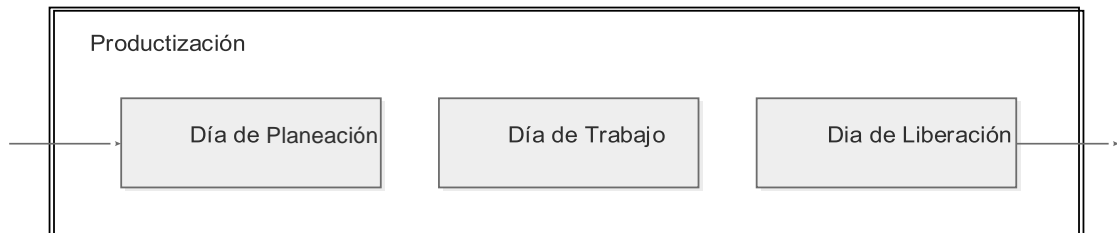
- **Productización o fase de producto.**

Se repiten iterativamente las subfases, con un día de planificación, uno de trabajo y uno de entrega. Aquí se intentan utilizar técnicas como la del test driven development para conseguir la mayor calidad (Mobile-D, 2015).

- Implementar la funcionalidad priorizada por el cliente para el producto.
- Enfocar la atención en la funcionalidad crucial de fondo, implementándola en los inicios del incremento para permitir múltiples ciclos de mejora.

Los estados de esta fase se pueden ver a continuación (Ver 15)

Figura 2.15 fase de Productización Estados



Fuente: (Mobile-D, 2015)

- **Fase de estabilización.**

Se llevan a cabo las acciones de integración para asegurar que el sistema completo funciona correctamente (Mobile-D, 2015).

Los objetivos de esta fase son:

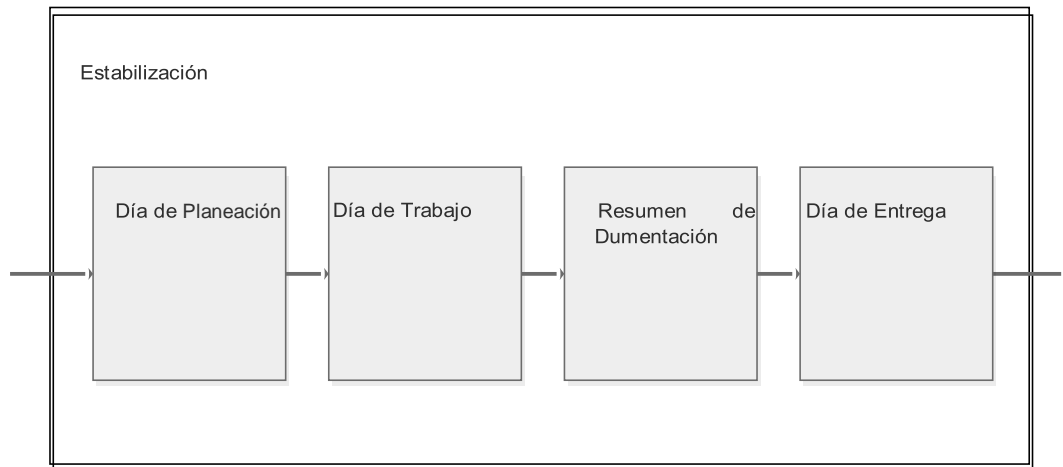
Se llevan a cabo las acciones de integración para asegurar que el sistema completo funciona correctamente (Mobile-D, 2015).

Los objetivos de esta fase son:

- Finalizar la implementación de los productos.
- Mejorar y asegura la calidad de los productos.
- Finalizar la documentación de los productos.

Los estados de esta fase se pueden ver a continuación (Ver 16)

Figura 2.16 fase de Estabilización Estados



Fuente: (Mobile-D, 2015)

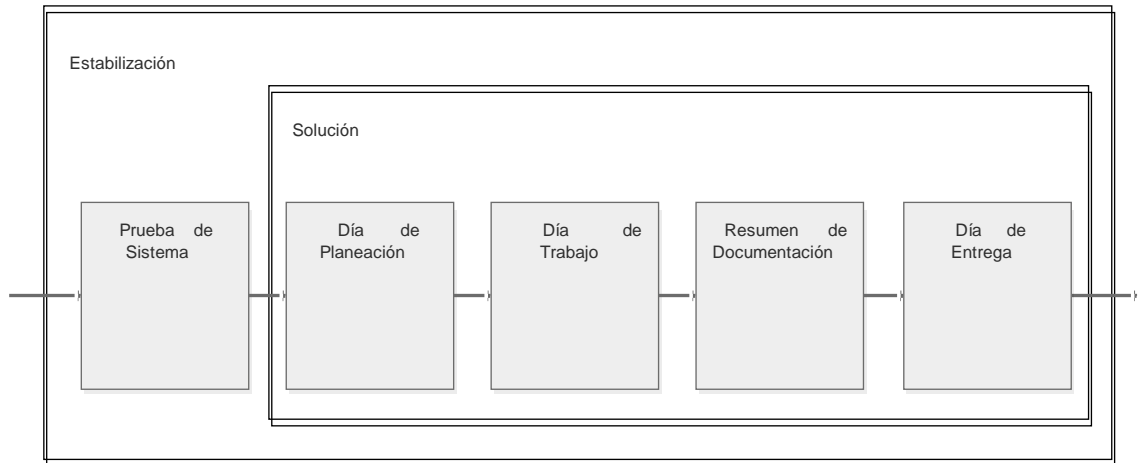
- **Fase de pruebas y reparación.**

Tiene como meta la disponibilidad de una versión estable y plenamente funcional del sistema según los requisitos del cliente (Mobile-D, 2015).

Los objetivos de esta fase son:

- Probar el sistema con base en la documentación producida en el proyecto
- Proveer información de defectos encontrados
- Permitir al equipo de proyecto preparar un plan para solucionar los defectos encontrados
- Arreglar los defectos
- Entregar un producto tan libre de errores como sea posible.
- El proceso de esta fase puede

Figura 2.17 etapa de Estabilización Estados



Fuente: (Mobile-D, 2005)

2.8.3 METODO DE INGENIERIA

1. identificación del problema
2. recolección de la información necesaria
3. búsqueda de soluciones creativas
4. pasar de la idea principal al diseño preliminar
5. evaluación y selección de la solución
6. preparación de reportes, planos y especificaciones
7. implementación del diseño.

- **IDENTIFICACIÓN DEL PROBLEMA:**

Un problema definido de manera adecuada, es un problema parcialmente resuelto. Plantear correctamente el problema es un paso importante hacia su solución.

- **RECOPIACIÓN DE LA INFORMACIÓN NECESARIA:**

Una vez que el problema está identificado y que las necesidades se han definido de manera adecuada, el ingeniero debe comenzar a reunir la información y los datos que precisa para resolverlo, el tipo de información que se requiere dependerá, por supuesto, de la naturaleza del problema a resolver.

- **BÚSQUEDA DE SOLUCIONES CREATIVAS:**

Después de completar los pasos preparatorios del proceso de diseño, el ingeniero está listo para comenzar a identificar las soluciones creativas. En realidad, el desarrollo de nuevas ideas, productos y dispositivos puede ser consecuencia de la creatividad, un esfuerzo subconsciente, o de la innovación, un esfuerzo consciente.

- **PASAR DE LA IDEA PRINCIPAL AL DISEÑO PRELIMINAR:**

El ingeniero está ahora listo para pasar de la idea al diseño preliminar. Este es el núcleo del proceso de diseño y es la fase que más depende de la experiencia y del buen juicio del ingeniero. Aquí es donde se descartan las ideas que no funcionan y las ideas que prometen se moldean para formar planos y diseños funcionales.

- **EVALUACIÓN Y SELECCIÓN DE LA SOLUCIÓN:**

En este punto el ingeniero evalúa sus posibles soluciones prometedoras y selecciona una de ellas, la más adecuada a su solución dando paso al siguiente punto de empezar a diseñar.

- **PREPARACIÓN DE REPORTE, PLANOS Y ESPECIFICACIONES**

- **IMPLEMENTACIÓN DEL DISEÑO**

Para recordar los pasos del método simplificado de la ingeniería se recomienda tener presentes las siglas DAMES (DAMAS en inglés) que significan:

D = Definir el problema

A = Analizar

M = Meditar

E = Evaluar alternativas

S = Señalar la solución

Desglosando este método de una manera más amplia, un ingeniero usa el siguiente

esquema de trabajo:

- a. Parte de una necesidad e identifica el problema
- b. Determina especificaciones
- c. Hace un estudio de factibilidad

d. Realiza una búsqueda de información

2.9 Metodología de Costos (COCOMO II).

Los objetivos principales que se tienen en cuenta para construir el modelo COCOMO II son:

Desarrollar un modelo de estimación de costo y cronograma de proyectos de software que se adaptara tanto a las prácticas de desarrollo de la década del 90 como a las futuras.

Construir una base de datos de proyectos de software que permitiera la calibración continua del modelo, y así incrementar la precisión en la estimación.

Implementar una herramienta de software que soportara el modelo.

Proveer un marco analítico cuantitativo y un conjunto de herramientas y técnicas que evaluaran el impacto de las mejoras tecnológicas de software sobre los costos y tiempos en las diferentes etapas del ciclo de vida de desarrollo. (González, 2017)

2.10 Seguridad ISO/IEC 27000.

La norma ISO/IEC 27000 son un conjunto de estándares que fueron y están siendo desarrolladas por la International Organization for Standardization (ISO) y la International Electrotechnical Commission (IEC), con el fin de proporcionar un marco de trabajo y administración de la seguridad de la información, para que esta pueda ser utilizada por cualquier organización sea esta pública, privada micro, mediana o grande. Siendo de todas ellas la norma ISO/IEC 27001 como la principal al ser la única certificable pudiendo ser aplicada a cualquier organización sin importar su giro, tamaño, privada o pública aportando una metodología orientada a la puesta en funcionamiento del SGSI (Sistema de Gestión de la Seguridad de la Información) al implementar controles para ello. Esta puede ser aplicada para resolver problemas dentro de las empresas que no tenga claro quien toma decisiones sobre ciertos activos de información, quien tiene acceso a ellos o quien autoriza el acceso a los sistemas informáticos de la empresa. (Coronado, 2019).

CAPÍTULO III

DISEÑO METODOLÓGICO

3.1. INTRODUCCIÓN

En el presente capítulo se llevará a cabo la construcción del controlador domótico y el desarrollo de la arquitectura que se va a implementar, utilizando las herramientas y la metodología descrita en el capítulo anterior. Asimismo, se realizará la construcción del controlador domótico y el funcionamiento, a través de los sensores, actuadores y medios de comunicación que se encarga de tomar datos, procesos, control y programación.

3.2 Metodología de desarrollo en “V “

3.1.2 Fase 1 Definición de Especificaciones y Requerimientos

Los requerimientos y especificaciones necesarias para el sistema domótico son los siguientes:

- Ausencia de instalación de software y hardware para automatismo en la institución no existe regulación de la iluminación y temperatura.
- Carencia de sistema de seguridad en el hogar, no existe instalación de alarmas y no existe un sistema de antirrobo.
- Los sistemas domóticos tradicionales usan el puerto serial o paralelo para efectuar la comunicación de hardware en el computador.
- Diversidad de sistemas de: seguridad, monitoreo, comunicación y confort para el hogar no están incorporados en un solo sistema y esto nos lleva a que el sistema este descentralizado y no podamos realizar un control de seguimiento del sistema domótico.

Tabla 3.1 Procesos del prototipo

Proceso	
1	Desplegar una pantalla de autenticación por el ingreso al sistema por medio de un navegador.
2	Desplegar una pantalla
3	Desplegar una pantalla del estado de los actuadores encendido o

	apagado de forma manual.
4	Desplegar una pantalla del contexto del modo automático
5	Desplegar una pantalla el apagado o encendido de los actuadores de forma automática en función de los sensores
6	Desplegar una pantalla para la administración de nuevos usuarios

Fuente: (Elaboración Propia)

- **Historias de Usuarios**

El acceso al sistema de control es exclusivamente para usuarios en la tabla 3.2 se especifica el proceso, el ingreso al sistema se realiza a partir de un administrador de sistema.

Tabla 3.2 Historia de usuario acceso al sistema de control

Historia de Usuario

Numero 1	Usuario: Habitante
Nombre historia	Usuario: propietario
Programador responsable	
Descripción	
Acceso al sistema de control domótico mediante su correspondiente usuario y contraseña.	
Observaciones:	
El ingreso se habilita a los usuarios.	

Fuente: (Elaboración Propia)

3.1.2 Fase 2 Diseño Global

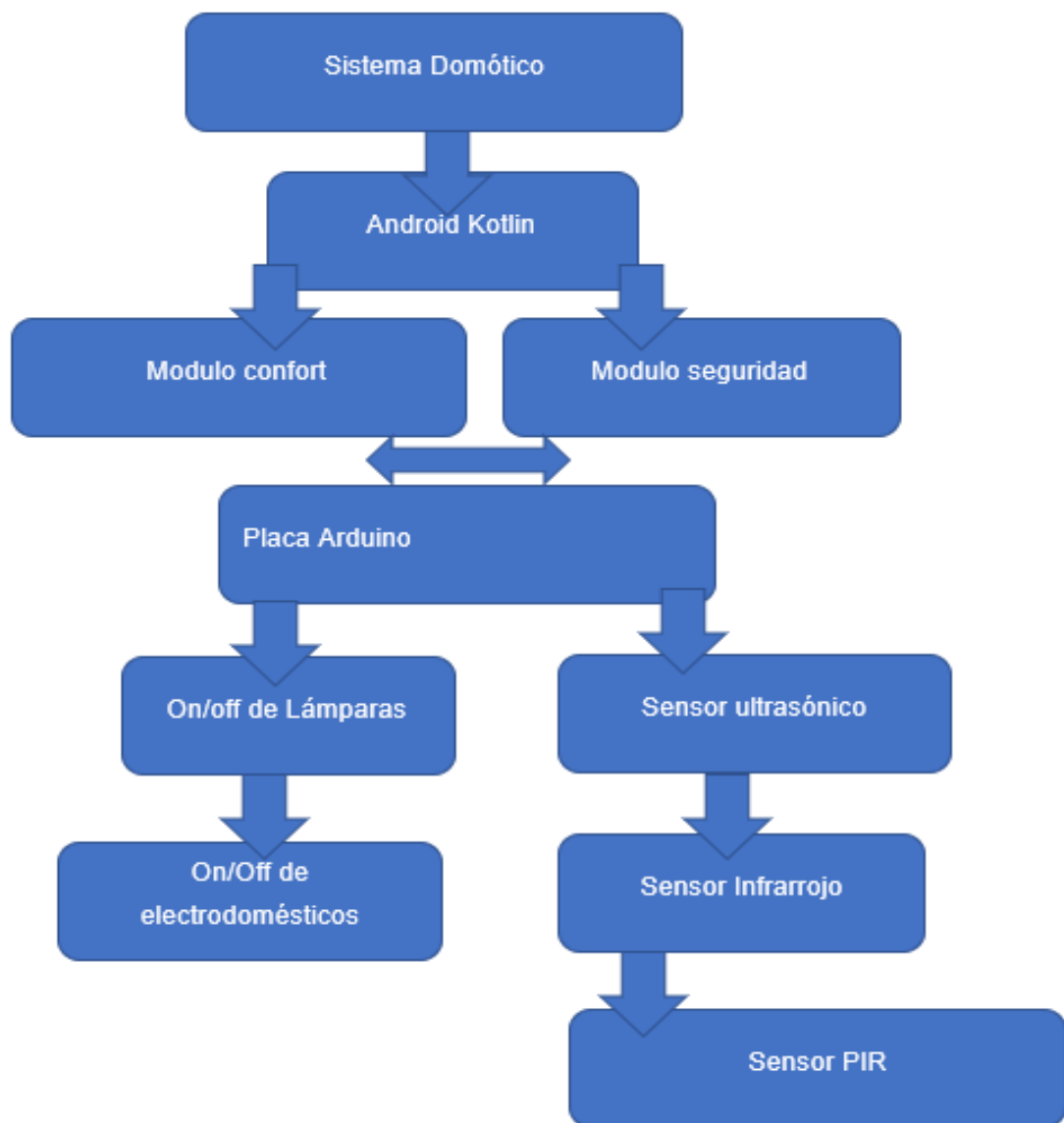
El diseño para el sistema domótico para la automatización de aparatos eléctricos y electrodomésticos para el confort y seguridad basado en software y hardware libre, van integrados a un microcontrolador Arduino que realiza una interfaz mediante el puerto USB con el ordenador mediante un el software de sistema domótico, porque de este elemento dependen los demás ya sean sensores o actuadores, tal como se puede observar en la figura.

La figura muestra la integración del sistema domótico con todos los módulos o elementos correspondientes respecto al aspecto de hardware centralizado a la placa Arduino.

3.1.3 Fase 3 Diseño en Detalle

La figura representa a detalle los más relevantes estados a cumplir y realizar por todos los elementos son:

Figura 3.1 Diseño en detalle



Fuente: (Elaboración Propia)

3.1.4. Fase 4 Implementación

Para esta fase consideramos dos aspectos importantes dado a que el prototipo comprende de hardware y software, para ello se tienen los siguientes.

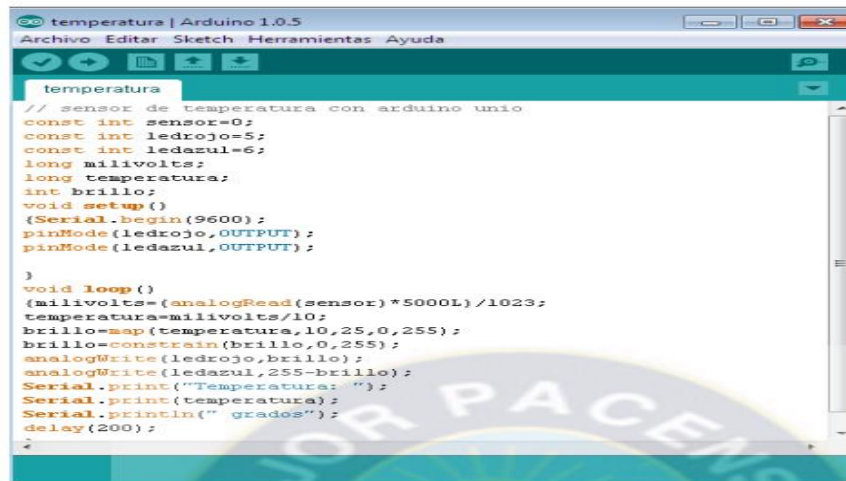
- **Conexiones**

Los sensores tienen tres cables: energía, la tierra, y de señal. El cable de alimentación es normalmente de color rojo, y debe ser conectado al pin 5V de la placa Arduino. El cable de tierra es típicamente de color negro o marrón y debe ser conectado a un conector de tierra de la placa Arduino. El pasador de señal es normalmente de color amarillo, naranja o blanco y debe estar conectado a un pin digital en la placa Arduino. El módulo electroimán puede ser conectado consta de dos salidas que puede ser alimentado por 5 VCC y cuando se unen da la información de censado.

- **Codificación**

El uso de los módulos que componen este prototipo viene a ser controlados por el elemento central Arduino. La figura 3.11 corresponde al uso de las librerías de cada módulo que Arduino controla y además alguna de los procedimientos y variables locales con las cuales trabaja. La codificación fue desarrollada con el propio software del controlador Arduino v1.5.5.

Figura 3.2 Diseño en detalle



```
temperatura | Arduino 1.0.5
Archivo Editar Sketch Herramientas Ayuda

temperatura
// sensor de temperatura con arduino unio
const int sensor=0;
const int ledrojo=5;
const int ledazul=6;
long milivolts;
long temperatura;
int brillo;
void setup()
{Serial.begin(9600);
pinMode(ledrojo,OUTPUT);
pinMode(ledazul,OUTPUT);
}
void loop()
{milivolts=(analogRead(sensor)*5000L)/1023;
temperatura=milivolts/10;
brillo=map(temperatura,10,25,0,255);
brillo=constrain(brillo,0,255);
analogWrite(ledrojo,brillo);
analogWrite(ledazul,255-brillo);
Serial.print("Temperatura: ");
Serial.print(temperatura);
Serial.println(" grados");
delay(200);
}
```

Fuente: (Elaboración Propia)

3.1.5 Fase 5 test Unitario

Para verificar el correcto funcionamiento de los elementos que componen este prototipo se debe tomar en cuenta que implica dos aspectos, hardware y software.

- **Hardware**

La siguiente figura 3.14 expone los elementos de hardware que involucra la construcción del afinador automático. Su funcionalidad es esencial para correcta función de todo como uno.

Figura 3.3. Interfaz Arduino y PC



Fuente: (Elaboración Propia)

Tabla 3.3 Documentación de Pruebas 1

Documentos de prueba 1

Modulo	Arduino
Actividad	Control de módulos
Observación	Función Correcta
Detalle de observación	Comunicate las placas Arduino sea el modelo que fueron, funcionaran a la perfección

Fuente: (Elaboración Propia)

- **Software**

El módulo más importante en lo que llega a ser software, es la aplicación de interfaz entre el pc y la placa Arduino programa con el lenguaje de programación en Android Kotlin, esta aplicación interactúa directamente con el usuario, su interfaz muestra información del trabajo que realiza el dispositivo la automatización de confort y seguridad de un hogar.

Tabla 3.4 Documentación de Pruebas

<i>Documento de prueba</i>	
Modulo	Aplicación de interfaz Kotlin.
Actividad	Determinar la interfaz entre Arduino y el pc mediante la automatización en confort y seguridad.
Observación	Su función es como se esperaba
Detalle de Observación	Cada elemento que se puede observar en la interfaz de la aplicación funciona como se esperaba.

Fuente: (Elaboración Propia)

3.1.6 Fase 6 Integración

Todo el dispositivo desde el punto de vista sistémico va conformado por los módulos esenciales que se aprecia en la figura, totalmente integrado y en correcto funcionamiento.

Tabla 3.13 Documentación de Pruebas 5

<i>Documento de Pruebas</i>	
Modulo	Aplicación de interfaz en Android, Arduino, Actuadores, Sensores
Actividad	Las pruebas realizadas son el funcionamiento correcto del sistema domótico. En el aspecto de hardware que va conformado por los módulos que van integrado a la placa

Arduino funcionan correctamente. Con la integración de ambas partes de lo que son el hardware y software el dispositivo funciona correctamente.

Soluciones o correcciones	Con respecto a la conexión del puerto USB demora un poco en reconocer la interfaz entre el programa y la placa Arduino, pero funciona correctamente.
----------------------------------	--

Fuente: (Elaboración Propia)

3.1.7. Fase 7 Test operacional del Sistema

En esta última fase se realiza las pruebas del sistema como producto final, a diferencia de la anterior fase, esta vez involucrando la maqueta de un hogar que es el principal objetivo de estudio la tabla muestra la documentación de funcionamiento de la automatización del hogar sobre un escenario real.

Tabla 3.6 Documentación de Pruebas

<i>Documento de Pruebas</i>	
Módulo	Aplicación de interfaz en
Actividad	Con las diferentes pruebas realizadas en un escenario real se tienen las siguientes observaciones: La trasmisión de datos desde la interfaz hasta la placa Arduino viene siendo no muy fluido esto debido a que existe un retardo de tiempo en el código. los sensores se acondicionaron adecuadamente para que puedan realizar el senseo correcto.
Soluciones o correcciones	Para corregir el punto uno de la parte detalles del presenta documento es

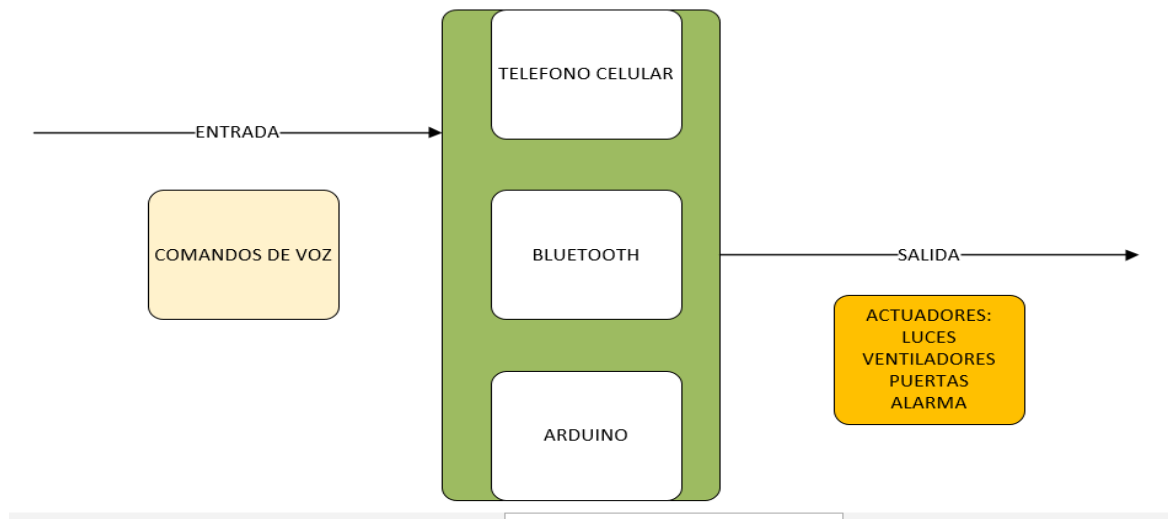
necesario quitar el código de retardo para que de esa forma la transmisión de datos sea más fluido, función en tiempo real. El punto de las dos observaciones es más relevante, pero tanto los sensores como actuadores se tendrá cuidado en utilizar tensión alterna y continua.

Fuente: (Elaboración Propia)

3.3 MODELO DE SISTEMA

El presente modelo tiene la característica básica de mostrar el proceso de la información como se muestra en la Figura 3.4:

Figura 3. 4: Proceso del sistema elemental



Fuente: Elaboración Propia

- **Entrada**

Eventos físicos en un ambiente y acciones realizadas por los usuarios de la instalación.

Los eventos físicos que se tomarán en cuenta como datos de entrada son:

movimientos físicos dentro del área del sensor de movimiento, el accionar de un ventilador, apertura y cierre de la puerta principal.

Estas entradas son capturadas de los siguientes dispositivos: Sensor de movimiento y del sensor de temperatura.

- **Proceso**

Está conformada por la captura de información desde los sensores y la programación de los mismos. La captura de información constituye un proceso y se consigue de la lectura de datos que los sensores entregan después de haber sido instalados en un ambiente.

El proceso que se sigue a continuación es realizar la programación integrando las salidas de los sensores en un solo programa estructurado controlado desde un dispositivo móvil.

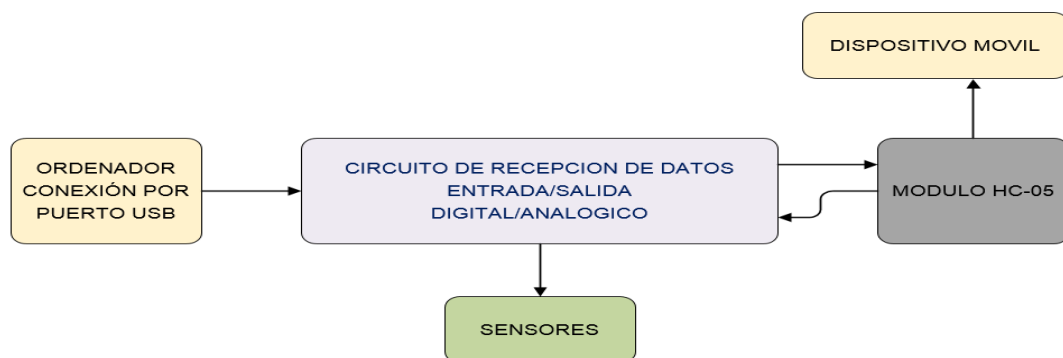
- **Salida**

De acuerdo a lo descrito anteriormente los resultados son: activación y desactivación de sistemas de alarma, apagado y encendido automático de iluminación, automatización de apertura y cierre de puertas realizadas por el usuario desde un dispositivo móvil.

3.4 HARDWARE DEL SISTEMA

El hardware principal de Arduino Mega2560 que será utilizado como unidad de control y proceso de datos de entradas y salidas de los dispositivos programados, en la Figura 3.3 se puede ver las unidades fundamentales del sistema y la relación entre ellos.

Figura 3. 5: Relación de unidades del sistema

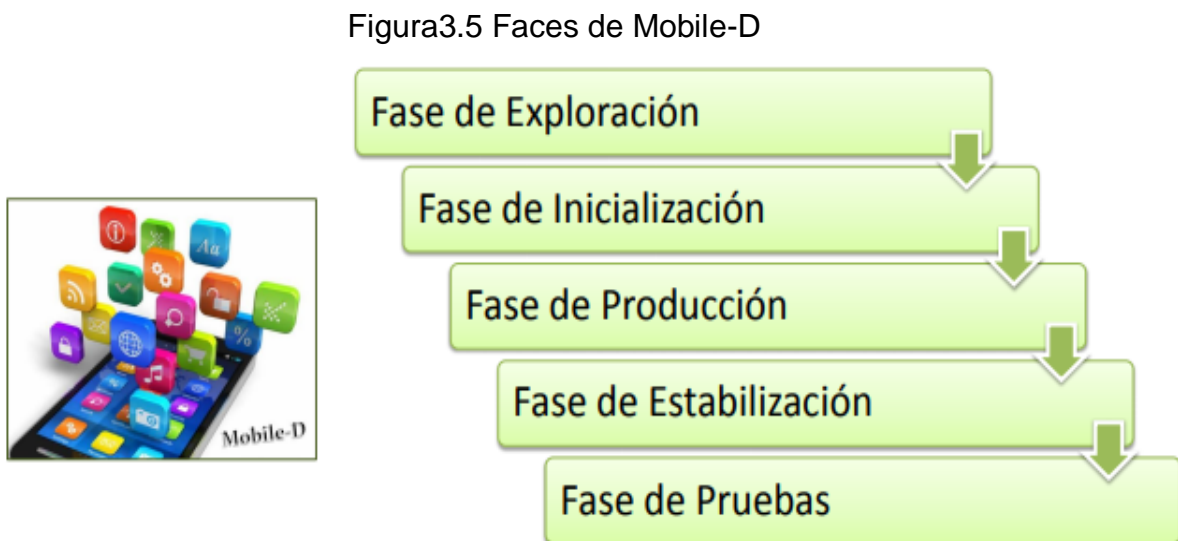


Fuente: Elaboración Propia

- Intensivas de Red. Porque reside en la red y proporciona servicios que satisfacen las necesidades de los clientes.
- Evolución Continua. A diferencia de las aplicaciones convencionales, las aplicaciones web están en constante evolución por esta razón se debe crear una estructura inicial que permita crecer.

3.5 DESARROLLO DEL SISTEMA EN BASE A LA METODOLOGÍA MOBILE D.

La metodología utiliza para el desarrollo de la aplicación móvil es Mobile-D el cual tiene las siguientes fases presentadas en la figura 3.1.



Fuente: (Elaboración Propia)

3.5.1 Fase I - Exploración

3.5.2 Definir Alcance

La definición del alcance tiene como objetivo generar un plan para el proyecto, el cual será el que guíe el proceso de desarrollo a lo largo de las siguientes fases.

Pretende también incluir todos los requerimientos iniciales para que puedan ser organizados y clasificados posteriormente.

La Tabla 3. 1 muestra los requerimientos iniciales obtenidos a través de reuniones con el Cliente. Estos requerimientos serán posteriormente clasificados de acuerdo a

determinados parámetros para su correcta asimilación y posterior realización.

Tabla 3.7 Planeación Inicial Del Proyecto

<i>Planeación Inicial Del Proyecto</i>	
Nombre	Planeación inicial del Proyecto
Proyecto	Sistema de Control Domótico Basado en Arduino y Aplicación móvil
Fase	Exploración
Requerimientos	
Id	Descripción
R1	Desarrollar una aplicación móvil capaz de ejecutarse en la mayoría de los dispositivos con sistema Android de mercado
R2	Desarrollar un interfaz de usuario que sea sencilla e intuitiva para el usuario
R3	Crear una adaptable en lo que se refiere a la legibilidad.

Nota: fuente (Elaboración propia)

Posteriormente en la Tabla 3. 2 se muestra también la clasificación de los requerimientos previamente obtenidos para ser puestos dentro del plan del proyecto.

Tabla 3.8 Requerimientos Iniciales

Requerimientos Iniciales del Proyecto

<i>Requerimientos Iniciales del Proyecto</i>		
Nombre	Requerimientos Iniciales del Proyecto	
Proyecto	Sistema de Control Domótico Basado en Arduino y Aplicación móvil	
Fase	Exploración	
Requerimientos Iniciales		
IDE	Descripción	Prioridad

Área del Requerimiento; interfaz

R2	Desarrollar un interfaz de usuario que sea sencilla e intuitiva para el usuario	Media
R1	Desarrollar una aplicación móvil capaz de ejecutarse en la mayoría de los dispositivos con sistema Android de mercado	Alta

Nota: Fuente: (Elaboración propia)

- **Definir Proyecto**

Una vez definidos los aspectos teóricos se pasa a la última subfase de esta etapa que contempla la creación de un plan de proyecto actualizado que contenga toda la información

3.5.2 Fase II – Inicialización

- **Establecimiento del proyecto**

En esta etapa se establecen los recursos físicos y técnicos para el proyecto, así también el ambiente para monitorearlo, y se establecen los medios de comunicación con los grupos o usuarios de interés.

- **Configuración del ambiente de desarrollo**

Esta tarea consiste en configurar los ambientes tanto físicos como técnicos para el proyecto. Esta tarea involucra al desarrollador de software en el ambiente técnico de desarrollo, además se realizan prueba de concepto sin necesariamente implementar algún requerimiento.

Configuración de App móvil

Tipo de proyecto: Android Studio versión 1.2 preview 2015

Configuraciones: Ninguna específico

Versión de android: Android 4.1 jelly bean

- **Entrenamiento**

Es una tarea en la cual el equipo de desarrollo se entrena de acuerdo a las

necesidades específicas. Incluye procesos de desarrollo, metodología, colección de datos, problemas técnicos relacionados como nuevas herramientas y métodos.

Planificación de fases

Tabla 3.9 Plan de entrenamiento

Plan de Proyecto

Nombre	Plan de Proyecto	
Proyecto	Sistema de Control Domótico Basado en Arduino y Aplicación móvil	
Fase	exploracion	
	Tópicos a estudiar	Tiempo(semanas)
	Descripción	
	Interfaz de Android Studio	1
	Diseño de interfaz de android	
	Interfaces reponsivas para android	

Nota: Fuente: (Elaboración propia)

Una vez definidos el plan de entrenamiento, es necesarios generar la Lista de Verificación del Proyecto, para esto se ha utilizado la plantilla provista en la documentación oficial de Mobile-d, puede observarse en la Tabla 3. 6.

Este documento certifica que la información necesaria ha sido obtenida, clasificada y ordenada.

Una vez realizada esta comprobación se tendrán todos los requisitos necesarios para

comenzar el desarrollo del proyecto.

Tabla 3.10 Lista de verificación del proyecto.

<i>Lista de verificación del proyecto</i>			
Nombre	Lista de Verificación		
Proyecto	Sistema de Control Domótico Basado en Arduino y Aplicación móvil		
Fase	Exploración		
Lista de verificación del proyecto			
Requerimientos		SI	NO
Todos los requerimientos funcionales han sido incluidos en el plan de proyecto.		SI	
Todos los requerimientos no funcionales han sido incluidos en el plan de proyecto		SI	.
Recursos			
El plan de proyecto ha sido actualizado de tal forma que identifique los requerimientos planteados			SI
	por el cliente.		
Entrenamiento			
Las necesidades de entrenamiento han sido incluidas en el plan de proyecto.			SI
Documentación			
Los documentos que son producidos en el proyecto han sido incluidos en el plan del proyecto.			SI

Nota: Fuente: (Elaboración propia)

3.5.3 Fase III - Fase del Producto

Dentro de esta fase es donde se comienza a ejecutar el plan de proyecto realizado

en la fase previa. En base a las Task Cards e Story Cards es que se desenvuelve esta fase a cada iteración se busca obtener un módulo funcional y así pasar al siguiente.

- **Iteración**

Esta primera Iteración se centrará en aquellos requerimientos clasificados con prioridad “alta” y posteriormente se irán realizando uno a uno el resto de los requerimientos siguiendo el mismo parámetro.

- **Planeación**

muestra el proceso seguido en esta subfase. Este proceso se repite iteración por iteración tomando como referencia las Tarjetas de Historia realizadas previamente.

3.5.4 Fase IV - Fase de Estabilización y Fase de Pruebas – Corrección

- **Tarjeta CRC**

Las tarjetas CRC (clase-responsabilidad-colaboración) son una herramienta de BrainStorming, usado como metodología para el diseño de software orientado a objetos creada por Kent Beck y Ward Cunningham.

- **Casos de Prueba**

Un caso de prueba o test case es, en ingeniería del software, un conjunto de condiciones o variables bajo las cuáles un analista determinará si una aplicación, un sistema software (software system), o una característica de éstos es parcial o completamente satisfactoria.

Se pueden realizar muchos casos de prueba para determinar que un requisito es completamente satisfactorio. Con el propósito de comprobar que todos los requisitos de una aplicación son revisados, debe haber al menos un caso de prueba para cada requisito a menos que un requisito tenga requisitos secundarios. En ese caso, cada requisito secundario deberá tener por lo menos un caso de prueba. Algunas metodologías como RUP recomiendan el crear por lo menos dos casos de prueba para cada requisito. Uno de ellos debe realizar la prueba positiva de los requisitos y el otro debe realizar la prueba negativa.

- **Interfaz de usuario**

Login del sistema

Tabla 3.12: Login de la aplicación.

Figura3.6 menú principal



Fuente: (Elaboración propia)

Menú principal

Tabla 3.11: Menú principal

Menú principal

Permite ingresar al sistema ingresando el usuario y contraseña.

Módulo de seguridad para que solo el usuario registrado pueda ingresar al sistema. Interfaz de usuario para que pueda ingresar su user y password.

Nota: Fuente: (Elaboración propia)

Tabla 3.12: Menú principal

Casos de Prueba

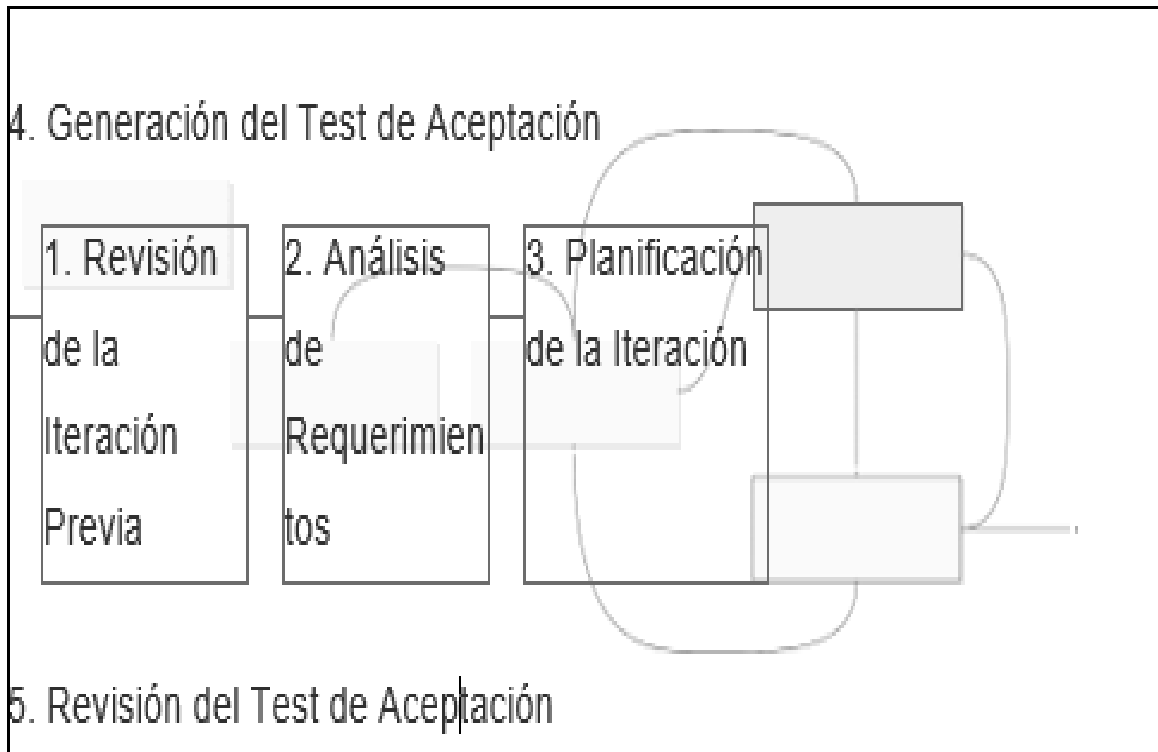
Pruebas De Acceptation

Nombre de Teléfono	Xiomí	
Versión SW	Android 6.1	Android 4.4

Nota: Fuente: (Elaboración propia)

Día de Planeación

Figura 3.3 menú principal



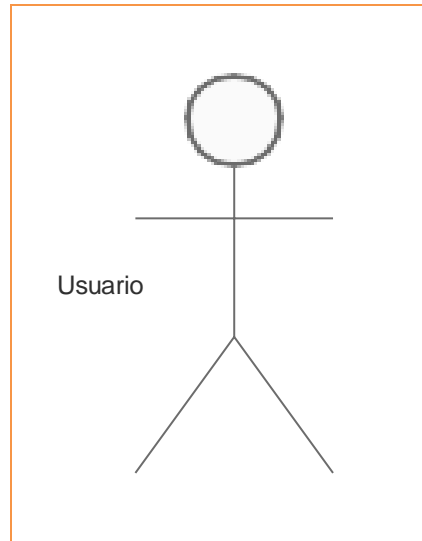
Fuente: (Mobile - D)

La siguiente tabla muestra la definición para la primera Tarjeta de Historia de acuerdo a las prioridades establecidas en las fases previas.

Actores del sistema

Para el desarrollo del sistema se identificaron dos actores principales como se observa en la figura 3.4:

Figura 3. 6 actor del Sistema



Fuente: (Elaboración propia)

Tabla 3.13.: Usuario

Usuario:	Es la persona que utiliza la aplicación e interactúa con ella para obtener
----------	--

Nota: Fuente: (Elaboración propia)

información.

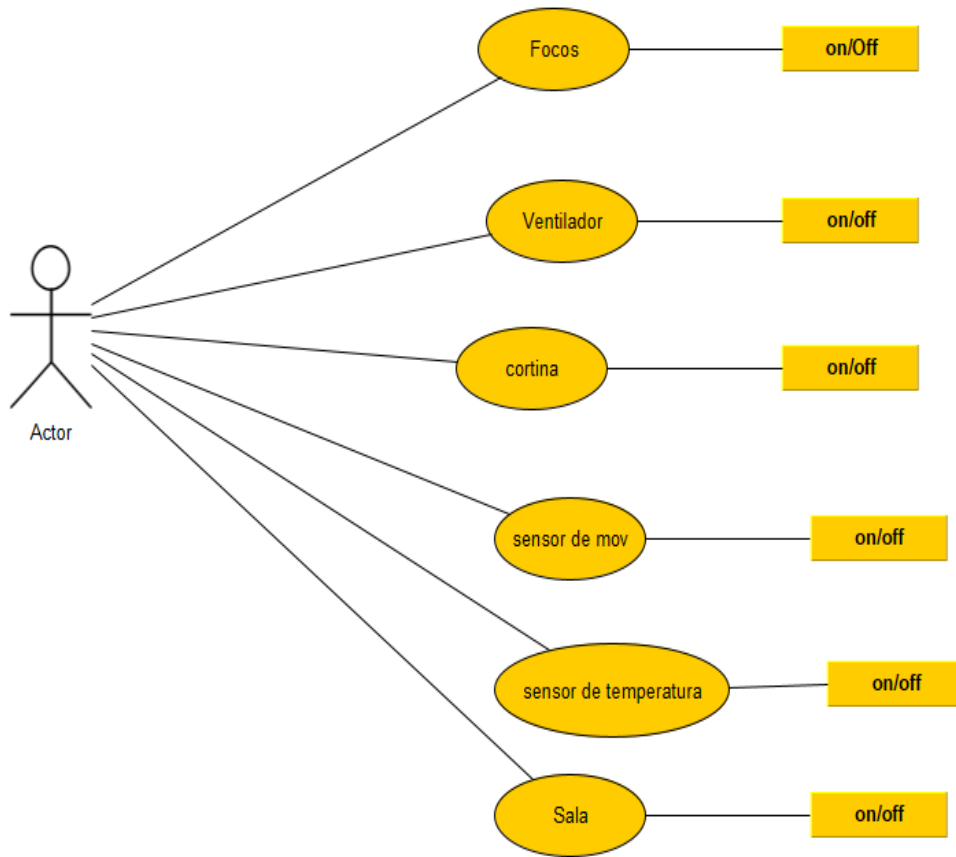
Diagrama Global del Sistema

A continuación, se muestra el diagrama global de funcionamiento del sistema, el cual se ira desglosando en los capítulos posteriores.

Este gráfico muestra los procesos previstos en base a los requerimientos, los cuales serán desarrollados mediante iteraciones.

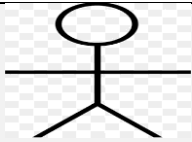
En el gráfico también se muestran las relaciones que existen entre cada proceso, para así hacer más claro el desarrollo posterior.

Figura 3.7 diagrama Global del Sistema.



Fuente: (Elaboración propia)

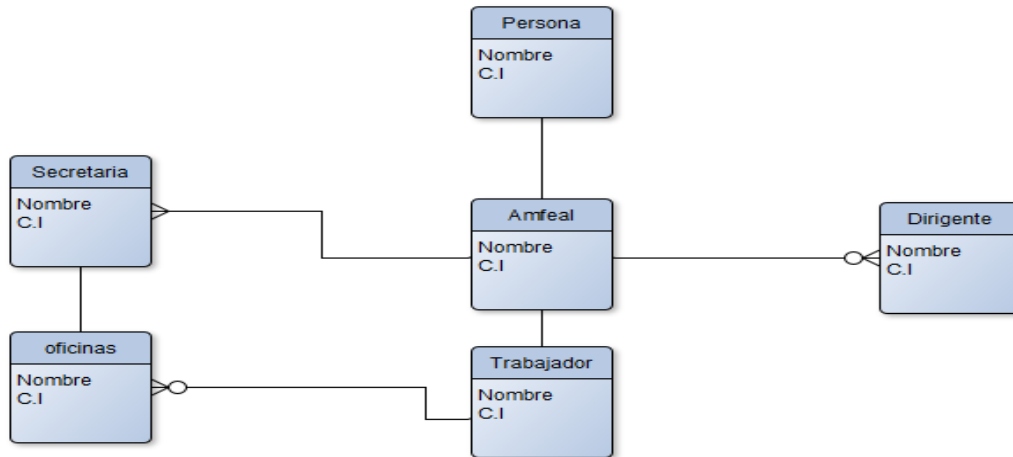
Tabla 3.14: Menú principal

<i>Actor</i>	<i>Descripción</i>
	<p>Es la persona que toma decisiones fundamentales sobre el funcionamiento del sistema</p> <p>Es la persona que puede manipular desde la aplicación móvil: como ser prender y apagar los diferentes aparatos electrodomésticos</p>

Fuente: (Elaboración propia)

Diagrama de Clases

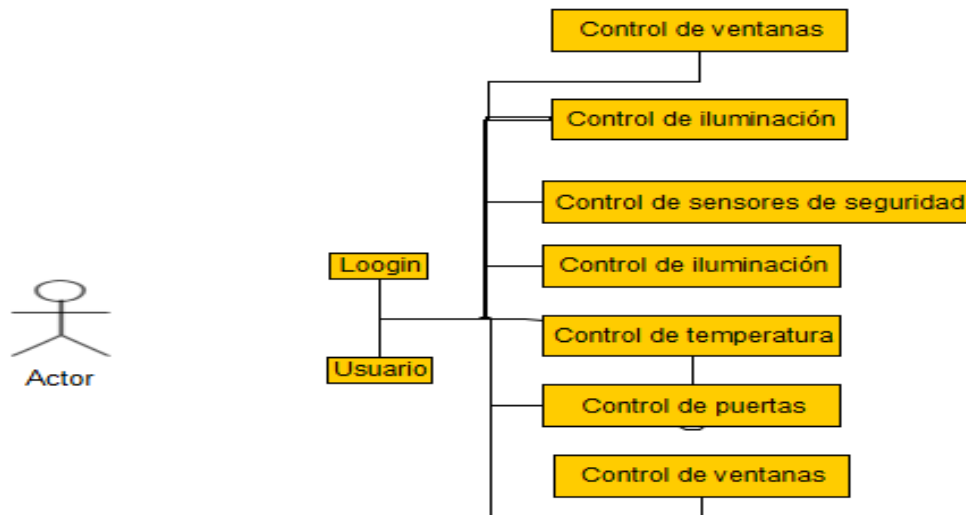
Figura 3.8 diagrama Global del Sistema



Fuente: (Elaboración propia)

Diagrama Navegacional

Figura 3.8 diagrama Global del Sistema



Fuente: (Elaboración propia)

Tabla 3.15: Menú principal

<i>Descripción del diseño navegacional global</i>	
Modulo	Descripción
Lampara	Tiene las opciones de prender y apagar.
Sensor de movimiento	Tiene las opciones de prender y apagar.
Ventiladora	Tiene las opciones de prender y apagar, también se prende automáticamente en una determinad a temperatura.
Puerta	Tiene las opciones de abrir y cerrar.
Sensor de temperatura	Tiene función de medir la temperatura.

Fuente: (Elaboración propia)

Figura 3. 6 empezando a programar

```

sketch_jun20a Arduino 1.6.3
Archivo  Editar  Programa  Herramientas  Ayuda
✓ → 📄 ⬆ ⬇
sketch_jun20a
1 void setup() {
2   // put your setup code here, to run
3
4 }
5
6 void loop() {
7   // put your main code here, to run r
8
9 }
    
```

Fuente: (Elaboración propia)

3.5.5 Fase 5

Tiene como meta la disponibilidad de una versión estable y plenamente funcional del sistema. El producto terminado e integrado se prueba con los requisitos de cliente y se eliminan todos los defectos encontrados.

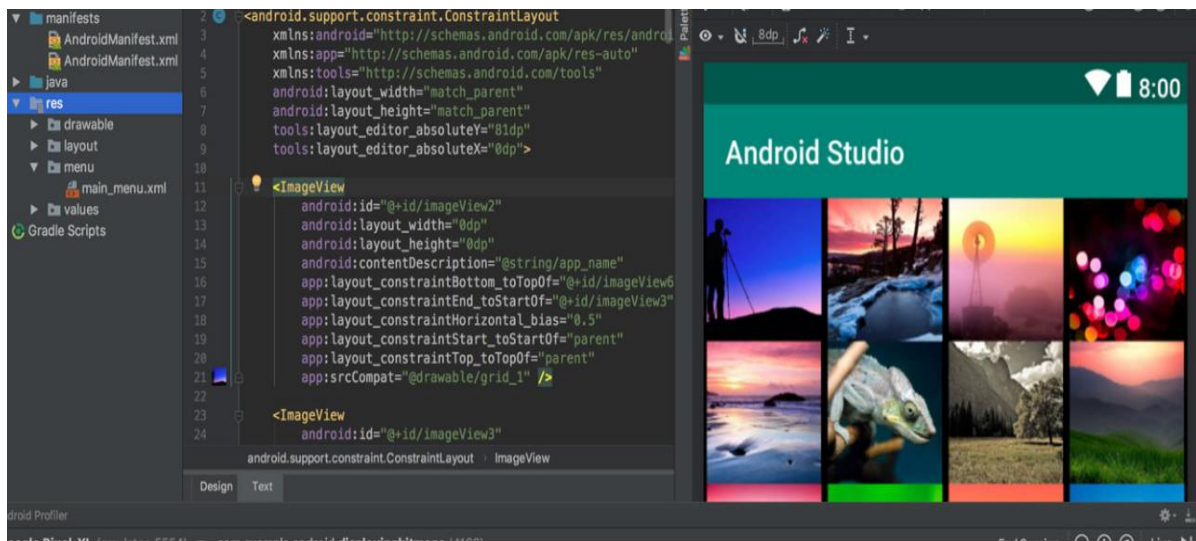
Las entradas de esta fase son las siguientes:

- La funcionalidad implementada.
- Documentación de aceptación de pruebas.
- Funcionalidad del usuario definida completamente.
- Descripción de la interfaz de usuario que se utiliza para crear casos de pruebas

Las salidas de la fase de pruebas son

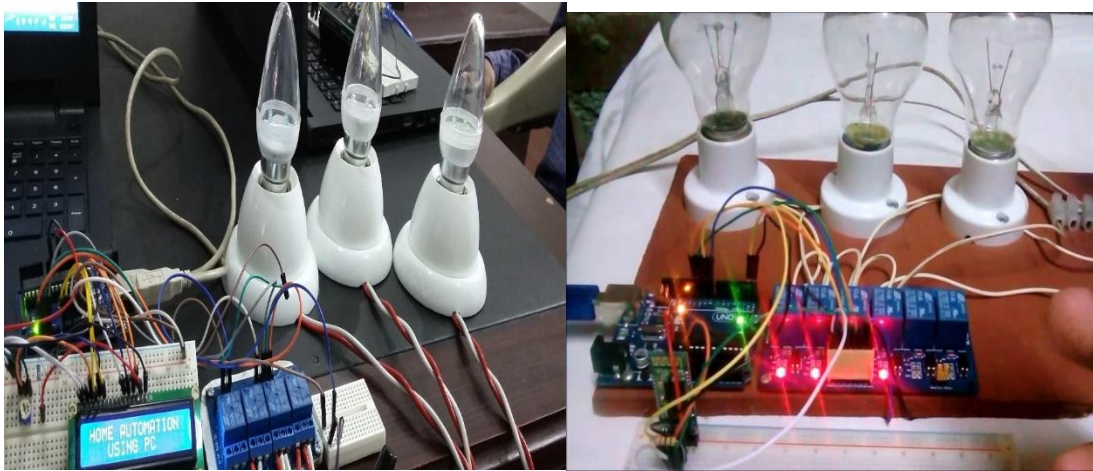
- Un sistema testeado y corregido (versión final)
- Documentación de errores encontrados.
- Informe de pruebas del sistema descripción del proceso de pruebas y los errores y defectos encontrados en el software.
- Registro de pruebas realizados en el sistema y los resultados obtenidos al momento de ejecutar el testeo.

- Figura 3. 7 editor de texto inteligente



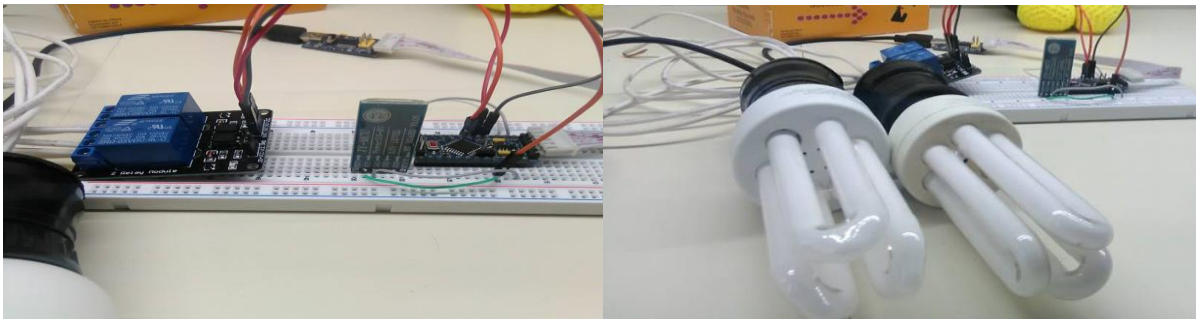
Fuente: (Elaboración propia)

Figura 3. 8 Editor de texto inteligente



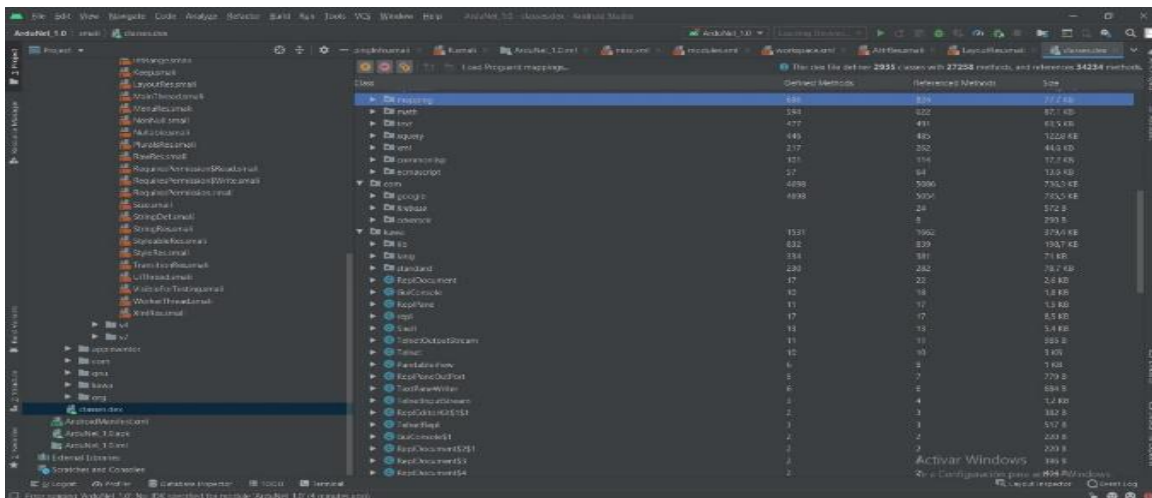
Fuente: (Elaboración propia)

Figura 3. 9 (Pruebas en el prendido de focos)



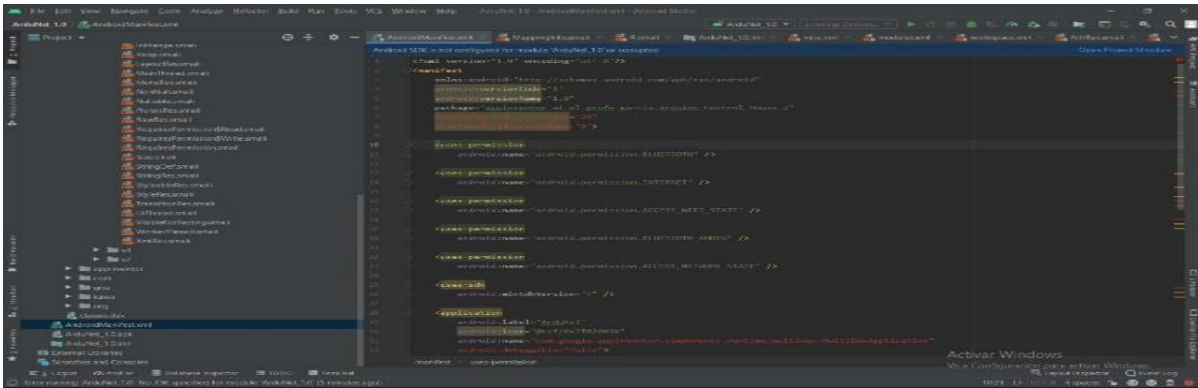
Fuente: (Elaboración propia)

Figura 3. 10 (Editor de texto inteligente y código fuente en Android)



Fuente: (Elaboración propia).

Figura 3. 6 (Editor de texto inteligente y código fuente en Android)

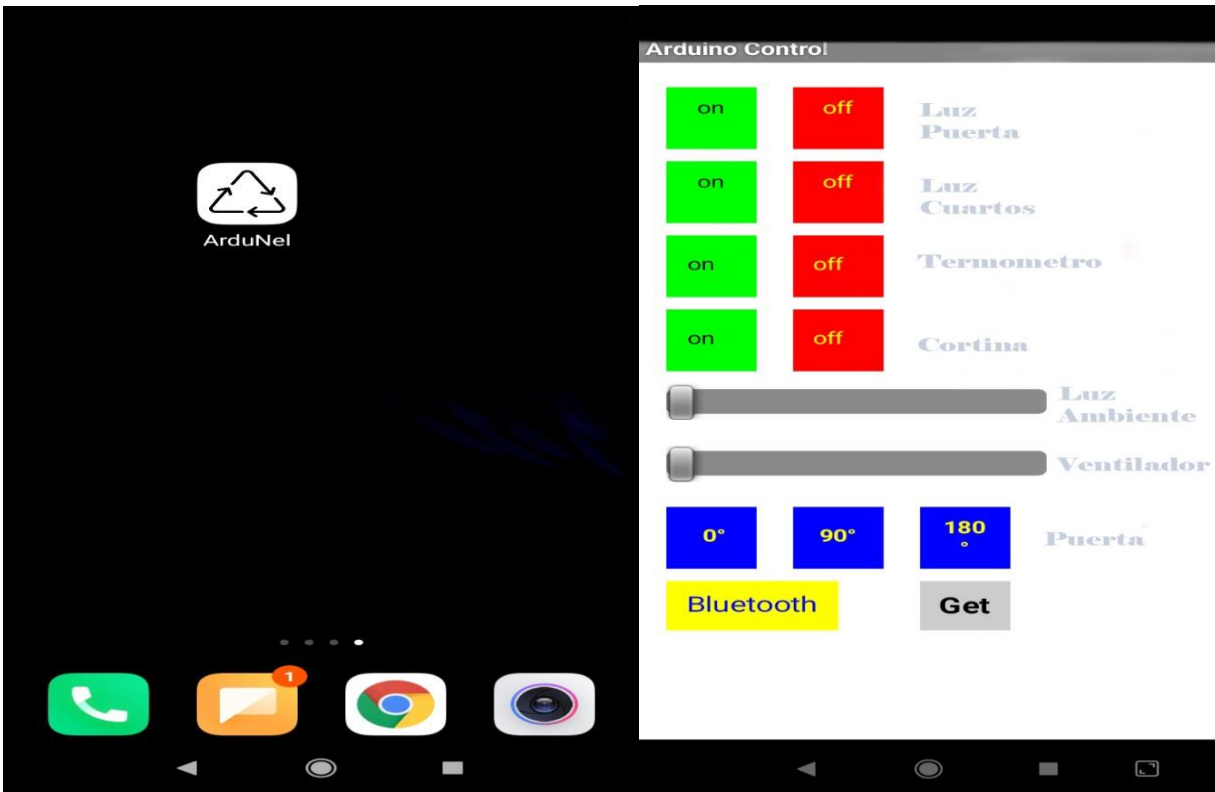


Fuente: (Elaboración propia).

Descripción de la interfaz de usuario que se utiliza para crear casos de pruebas

Figura 3. 11 (Aplicación móvil).

Figura 3. 12 (Interfaz de la aplicación).



Fuente: (Elaboración propia).

Fuente: (Elaboración propia).

CAPÍTULO IV

**CALIDAD, PRUEBAS,
SEGURIDAD
Y ANÁLISIS DE COSTOS**

4. Seguridad, Control

4.1 Seguridad

Las medidas prácticas de seguridad para las aplicaciones móviles, tales como una buena gestión de dispositivos móviles y el cifrado de datos almacenados, pueden mitigar los riesgos de la entrega de aplicaciones móviles. Así que es imperativo entender los riesgos inherentes asociados con los dispositivos móviles, las medidas de seguridad nativas integradas en los sistemas operativos móviles aportan a este objetivo.

La seguridad parte desde el ingreso y acceso al equipo móvil con un patrón de seguridad, un PIN o una contraseña. A partir de ahí para el ingreso a la aplicación se parte desde la Identificación y Autenticación.

4.2 Identificación y Autenticación

En este caso la autenticación de usuario para la aplicación móvil es el proceso de determinar si una persona está autorizada para llevar a cabo una acción dada. La identificación y autenticaciones es considerada una de las primeras líneas de defensa para la mayoría de las aplicaciones, esto con el objetivo de impedir el acceso no autorizados a la información en el sistema móvil. El paso normal de autenticación del sistema consta de los siguientes pasos:

- El usuario solicita al sistema.
- El sistema le solicita al usuario que inserte su nombre de usuario y contraseña.
- El usuario realiza su identificación.
- El sistema valida según sus reglas los datos aportados por el usuario y verificar si son los válidos para dar acceso al sistema o no

4.5 Análisis Costo-Beneficio del Sistema

La técnica de análisis de Costo Beneficio tiene como objetivo fundamental proporcionar una medida de la rentabilidad del proyecto, mediante la comparación de los costos previstos con los beneficios esperados en la realización del proyecto.

4.6 Modelo Base de Estimación COCOMO

- **Punto Objeto**

El procedimiento para determinar Puntos Objeto en un proyecto software se resume en:

- **Determinar Cantidad de Objetos:** Estimar la cantidad de pantallas, reportes, componentes que contendrá la aplicación.
- Clasificar cada instancia de un objeto según sus niveles de complejidad (simple, media o difícil).
- Dar el peso a cada objeto según el nivel de complejidad. Los pesos reflejan el esfuerzo relativo requerido para implementar una instancia de ese nivel de complejidad.
- Determinar la cantidad de Puntos Objeto, sumando todos los pesos de las instancias de los tipos de objetos especificados.

El proyecto de acuerdo a sus características se clasifica en el siguiente modo de desarrollo:

- **Orgánico:** proyectos relativamente sencillos, menores de 50 KDLC líneas de código, en los cuales se tiene experiencia de proyectos similares y se encuentran en entornos estables.
- **Semi-acoplado Modelo intermedio:** proyectos intermedios en complejidad y tamaño (menores de 300 KDLC), donde la experiencia en este tipo de proyectos es variable, y las restricciones intermedias. Para este caso el modelo intermedio será el que se ajusta para una estimación con bastante estimación.
- **Empotrado:** proyectos bastante complejos, en los que apenas se tiene experiencia y se engloban en un entorno de gran innovación técnica. Además, se trabaja con unos requisitos muy restrictivos y de gran volatilidad.

$$E = \text{Esfuerzo} = a \text{ KLDC}^e * \text{FAE} \text{ (persona x mes)}$$

$$T = \text{Tamaño de duración del desarrollo} = c \text{ Esfuerzo}^d \text{ (meses)}$$

P= Personal = E/T (personas)

En nuestro caso las líneas por cada PF equivalen a 53 líneas lógicas de acuerdo a los factores de diferentes lenguajes en este caso JAVA es el lenguaje que se utiliza en ANDROID STUDIO.

$KLDC = (PF * \text{Líneas de código por cada PF}) / 1000$

$KLDC = (261,36 * 53) / 1000 = 13,852$ $KDLC < 50$ $KLDC = \text{PROYECTO NO MUY COMPLEJO}$

PROYECTO DE SOFTWARE	a	b	c	d
Orgánico	3,2	1,05	2,5	0,38
Semi – acoplado	3,0	1,12	2,5	0,35
Empotrado	2,8	1,20	2,5	0,32

Tabla4.1 Tabla de ponderación

Fuente: (Elaboración propia)

- Cálculo del esfuerzo del desarrollo:

$$E = a \text{ KLDC } e * \text{FAE} = 3,2 * (13,852) ^{1,05} * 0,53508480 = 27,04 \text{ personas /mes}$$

- Cálculo tiempo de desarrollo:

$$T = c \text{ Esfuerzo } d = 2,5 * (27,04) ^{0,38} = 4,75 \text{ meses}$$

- **Productividad:**

$$PR = LDC/\text{Esfuerzo} = 5200/27,04 = 192,30 \text{ LDC/personas mes}$$

- **Personal promedio:**

$$P = E/T = 27,04/8,75 = 3,09 \text{ personas}$$

Según estas cifras será necesario un equipo de 3 personas trabajando alrededor de 8 meses, pero puesto que el desarrollo del proyecto debe realizarse en un plazo 4 meses, incrementaremos a 7 personas el número de personas del equipo de proyecto (ya que $27,04/4$ nos da alrededor de este resultado).

Con estos resultados tendremos un equipo formado por 1 jefe de proyecto, 2 Analistas, 3 programadores y 1 responsable de calidad.

- **Costo de Facilidades** El costo de facilidades es de 0 USD porque se cuenta con un cuarto de computadoras, oficina para el personal de operación, la gente de mantenimiento y el personal usuario.

- **Estudio del Sistema**

Se toma en cuenta que un ingeniero de software percibe un salario de 500 a 700 USD El costo de estimación de proyecto es:

- Tiempo equivalente a 4 meses

Costo del estudio del sistema por los 4 meses es 2000 USD.

- **Costos de Hardware**

La institución deberá invertir en la compra de dispositivos móviles de mediano costo para el personal operativo.

- Costo estimado por dispositivos = 170 USD
- Cantidad que se requiere es = 3 unidades total 1,020 USD
- Material de apoyo y de escritorio es de 120 USD

- **Costo del Software**

LDC son líneas de código estimadas de código

Función	LDC
Apagar	20
Prender	30
Medir temperatura	40
Sensor de seguridad	15

Tabla4.2 Tabla de estimación de código

Fuente: (Elaboración propia)

Tabla:4.3 LDC son líneas de código estimadas de código

Costo Personal	
Descripción	Costo
Software	50

Programador	70
Transcripción	50

Fuente: (Elaboración propia)

- El costo asignado al ingeniero y programador es = 0 USd por tratarse de un proyecto de grado. 83 - Costo de mano de obra = 70 USD

- **Costo Total operacional del Proyecto**

Tabla4.4: costo total

Descripción	Monto en USD
Costo de Hardware	200
Costo Software	100
Costo de Apoyo	50
Costo de Facilidades	0
TOTAL	350 USD

Fuente: (Elaboración propia)

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5. Conclusiones

Se desarrolló el software controlador del prototipo permitiendo el control de dispositivos como sensores y actuadores.

- Se concluye con el desarrollo del sistema de control domótico basado en Arduino, cumpliendo con el objetivo general.
- Se realizó el análisis de la información y requerimientos de la institución, para definir el alcance y la arquitectura del sistema domótico.
- Se procesó la información de los datos del sistema por medio de bluetooth y Trávez del Arduino.
- Se integró los elementos software, hardware y la aplicación móvil, para realizar la prueba y evaluar el funcionamiento del prototipo del sistema de control domótica.
- Se realizó la estimación del costo del software utilizando la metodología COCOMO II, indicando el requerimiento del personal, tiempo y el costo del desarrollo del software.
- Se desarrolló el diseño lógico y físico utilizando las metodologías MOBILE-D y MODELO EN V, basado en los requerimientos que permite generar un análisis amplio.
- Se desarrolló una interfaz de control sencillo en Android para el usuario.
- Se realizaron pruebas de instalación del prototipo, presentados sobre ello una maqueta que implementa los dispositivos domóticos.

5.2 Recomendaciones

Se recomienda Integrar nuevas tecnologías, ya que además de lograr una reducción significativa en el tiempo de utilización del usuario, serán de gran utilidad para reducir aún más el tiempo de solución a los problemas presentados y aumentando la calidad en el servicio de la aplicación móvil.

Es importante este tipo de proyectos considerar que se trabaja con tensiones de 5 voltios esto a nivel prototipo de maqueta para la implementación de sistema en una vivienda por lo tanto no es recomendable para sistemas domóticos con tensiones de 220 voltios se recomienda el uso de relés y otros dispositivos de amplitud en función

al voltaje para este tipo de conexiones importante realizar una lectura previa a la ficha técnica de cada dispositivo.

También se recomienda en cuanto a trabajos posteriores la aplicación en teléfonos móviles para el control manual y automático bajo plataformas Android integrara la domótica con arduino y Android, pero controlada con este dispositivo móvil.

La automatización de más dispositivos conectados en una vivienda caso multimedia que ya se realizó este proyecto, pero no en el campo de Arduino ni mucho menos con el Arduino uno control de persianas, puertas, cámaras web para el acceso en cuanto a seguridad, monitorizar el consumo de energía eléctrica realizando una aplicación móvil.

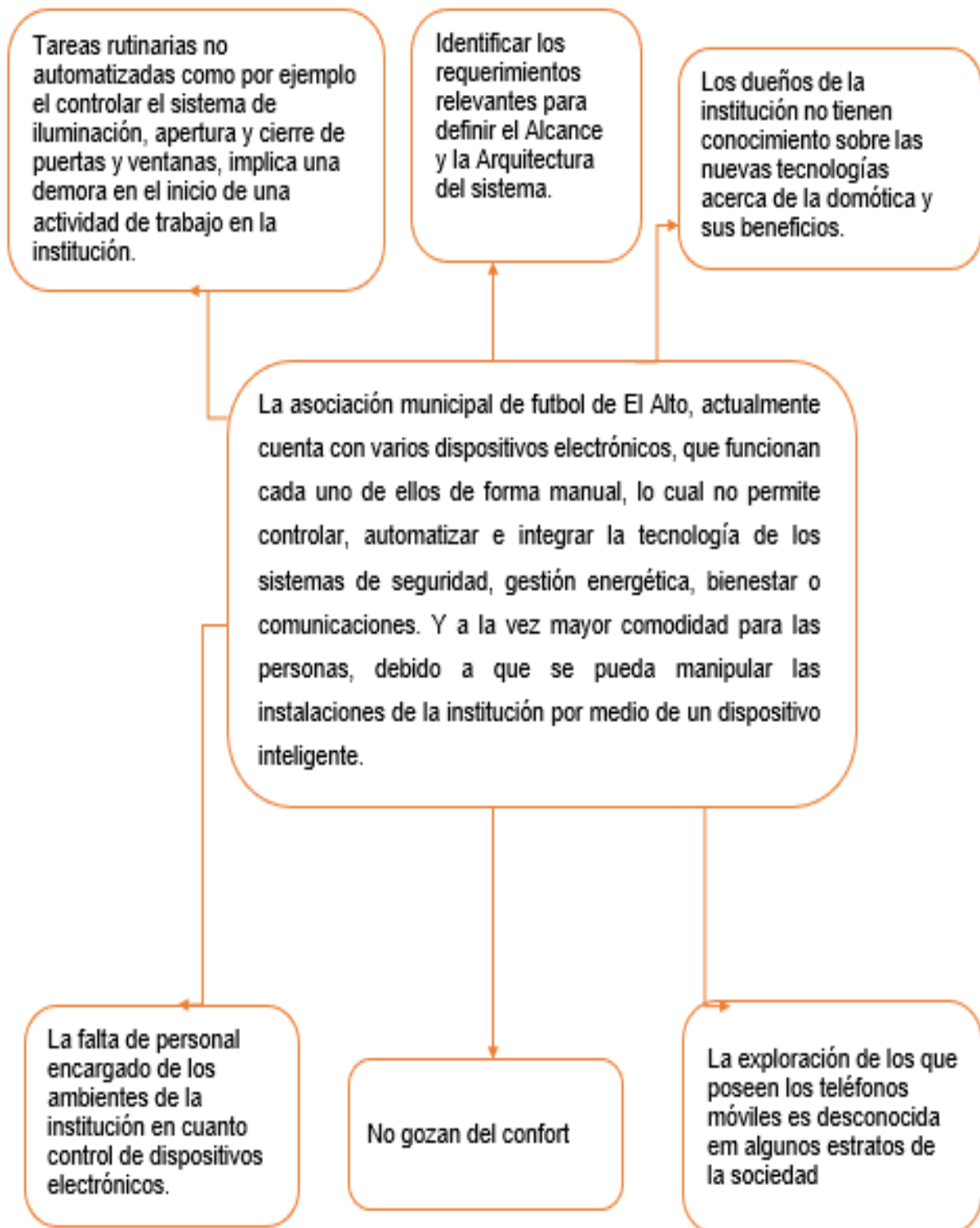
Bibliografía

- Álvarez, g. M. (2013). Automatizacion de un sistema de banco de agua helada en la empresa pil andina s.a. La paz – bolivia.
- Balibrea, r. H. (2012). Domotica. Cartajena.
- Barragán., j. L. (2011). Cocomo. Venezuela: escuela informatica educativa.
- Blanco, j. G. (2015). Sistema domotico para un hogar basado en software y hardware libre. La paz bolivia.
- Blanco, j. G. (2015). Sistema domotico para un hogar basado en software y hardware libre”. La paz.
- Chavez, j. G. (2015). Sistema domotico para un hogar basado en software y hardware libre. La paz bolivia.
- Chavez, j. G. (2015). Sistema domotico para un horgar basado en software y hardware libre.
- Coronado, e. D. (2019). Impotancia de la norma iso/eic 27000 en la implementacion de un sistema de gestion de la seguridad de la informacion . Mexico.
- Fernandez, a. (2019). Metodologia movile d. Colombia.
- González, g. G. (2017). Estimación de costo de software: una propuesta de aplicación pedagógica de cocomo. San jose, costa rica: universidad nacional a distancia uned.
- Gutierrz, j. M. (2007). Manual de programación. San francisco: arduino notebook: a beginner’s reference written .
- Loureiro, j. P. (2017). Arduino y el open source. Madrid españa: cc by-sa.
- Morales, f. S. (2017). Diseño e implementación de un sistema de automatización domótico para un salón prototipo en la facultad de ingeniería de la universidad distrital francisco José de caldas. Bogotá colombia.
- Pesantes, x. (2018). La domótica en la seguridad de la vivienda. Lima.
- Philco, m. O. (2018). Sensores delaware temperatura. Guayaquil.
- Ramires, j. (2019).
- Ramirez, j. (2002). Servo moto.
- Redondo, c. M. (2016). Proyecto de hogar digital para una vivienda basado en la tecnología knx. Cuenca ecuador: campus universitario.

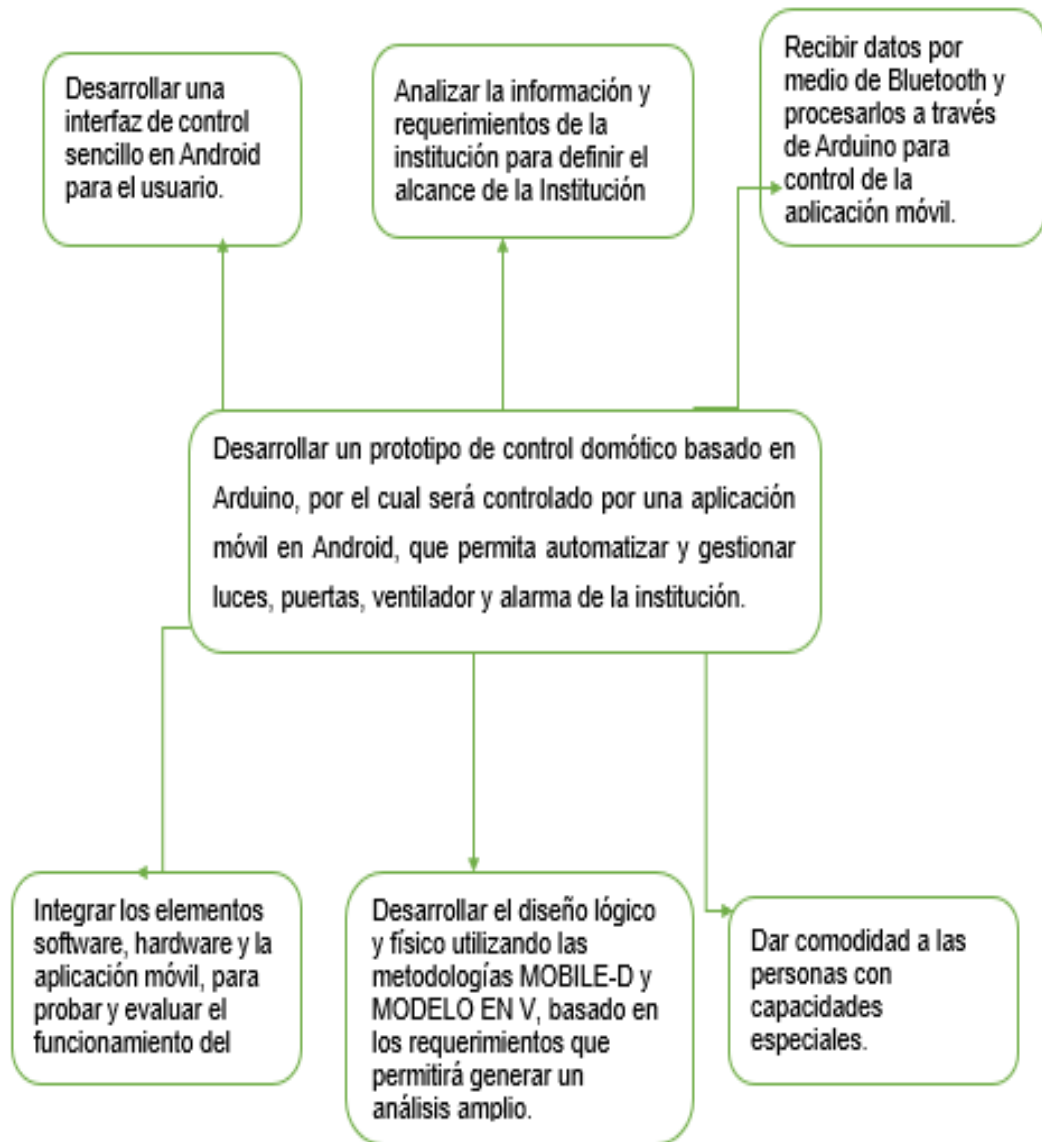
Sistema domotico para un hogar basado en software y hardware libre. (2015). La paz bolivia.

ANEXOS

ARBOL DE PROBLEMAS



ARBOL DE OBJETIVOS



MANUAL DE USUARIO



**SISTEMA DE CONTROL DOMÓTICO BASADO EN ARDUINO Y
APLICACIÓN MÓVIL**

**Sala de Reuniones de la Asociación Municipal de
Futbol “AMFEAL”**

Nombre: Nelson Guarachi Mita

Descripción

La aplicación móvil se encarga de prender y apagar los diferentes aparatos electrodomésticos dentro la Institución.

1. Requerimientos

Los requisitos para acceder a la aplicación web son:

- Tener un dispositivo móvil con sistema operativo Android



2. Instalación

El proyecto utiliza principalmente como entorno de desarrollo Java Kotlin.

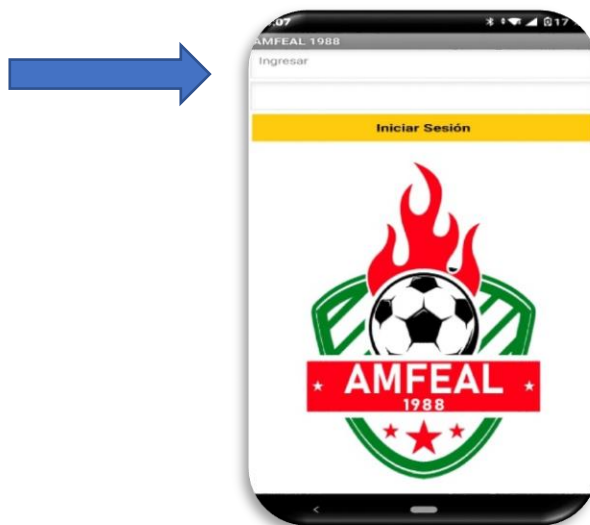
3. Ingreso a la aplicación móvil

Paso 1: Ingresar en el navegador la siguiente dirección web:

Usuario: Nelson

Contraseña:12345678

Al ingresar a la aplicación nos aparecerá la siguiente ventana principal.



4. Iniciar Sesión

Para iniciar sesión en el sistema el usuario previamente debe tener una cuenta creada en el sistema.

Paso 1: Presionar el botón de Iniciar Sesión.

Paso 2: Una vez presionado el botón de Iniciar Sesión se mostrará un formulario de inicio de sesión en donde se debe ingresar su correo electrónico y la contraseña correspondiente.

5. Menú Principal de la Aplicación

