

**UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA**



TESIS DE GRADO

**“CONTROL INTELIGENTE DE SEGURIDAD EN VIVIENDAS
MEDIANTE AGENTES MÓVILES”**

**PARA OPTAR AL TÍTULO DE LICENCIATURA EN INFORMÁTICA
MENCIÓN: INGENIERÍA DE SISTEMAS INFORMÁTICOS**

POSTULANTE: Blanca Alicia Machaca Azpe
TUTORA METODOLÓGICA: Lic. Menfy Morales Ríos
ASESOR: Dr. Guillermo Choque Aspiazu PhD.

LA PAZ – BOLIVIA
2015



**UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA**



LA CARRERA DE INFORMÁTICA DE LA FACULTAD DE CIENCIAS PURAS Y NATURALES PERTENECIENTE A LA UNIVERSIDAD MAYOR DE SAN ANDRÉS AUTORIZA EL USO DE LA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO SI LOS PROPÓSITOS SON ESTRICTAMENTE ACADÉMICOS.

LICENCIA DE USO

El usuario está autorizado a:

- a) visualizar el documento mediante el uso de un ordenador o dispositivo móvil.
- b) copiar, almacenar o imprimir si ha de ser de uso exclusivamente personal y privado.
- c) copiar textualmente parte(s) de su contenido mencionando la fuente y/o haciendo la referencia correspondiente respetando normas de redacción e investigación.

El usuario no puede publicar, distribuir o realizar emisión o exhibición alguna de este material, sin la autorización correspondiente.

TODOS LOS DERECHOS RESERVADOS. EL USO NO AUTORIZADO DE LOS CONTENIDOS PUBLICADOS EN ESTE SITIO DERIVARA EN EL INICIO DE ACCIONES LEGALES CONTEMPLADOS EN LA LEY DE DERECHOS DE AUTOR.

DEDICATORIA:

Quiero dedicar esta tesis a la memoria de mi madre, quien me dio la vida y me inculco el amor a los valores del bien, la verdad, la justicia, la responsabilidad y el respeto, a pesar de nuestra distancia física, siento que estás conmigo siempre y aunque nos faltaron muchas cosas por vivir juntas, sé que este momento hubiera sido tan especial para ti como lo es para mí.

Blanca Alicia Machaca Azpe

AGRADECIMIENTOS

Agradezco a Dios por protegerme durante todo mi camino y darme fuerzas para superar obstáculos y dificultades a lo largo de toda mi vida, pero sobre todo por darme las oportunidades que me llevaron hasta este momento tan importante de mi formación profesional.

Agradezco a mi asesor Dr. Guillermo Choque Aspiazu PhD., su experiencia profesional, las correcciones y consejos que ayudaron a formar este trabajo. También agradecer a mi tutora Lic. Menfy Morales Ríos, por el apoyo brindado en el proceso de conclusión de este trabajo. Gracias a los dos por sus orientaciones, ánimos y por haber estado dispuestos a ayudarme siempre que lo he necesitado.

Y por supuesto agradecer a mi familia, a mi padre por ser el pilar más importante y por demostrarme siempre su cariño y apoyo incondicional sin importar nuestras diferencias de opiniones, a mis hermanos Julio y Miriam por el apoyo brindado, por ser grandes amigos para mí, a mi tía Carmen, a quien quiero como a una madre, por compartir momentos significativos conmigo y por siempre estar dispuesta a escucharme y ayudarme en cualquier momento.

Gracias a todas las personas que ayudaron directa e indirectamente en la realización de esta tesis.

Blanca Alicia Machaca Azpe

RESUMEN

Seguridad es la ausencia del peligro o riesgo, sensación de confianza que se tiene en algo o alguien, mantener este concepto en viviendas es de vital importancia ya que es el lugar protegido para que la persona comparta su hábitat. Con respecto al avance tecnológico la propuesta se enfoca a los dispositivos móviles que actualmente está en disposición de gran parte de la población. De esta manera si se utiliza esta capacidad para brindar la seguridad que se solicita en viviendas sería de gran apoyo a la sociedad para satisfacer la necesidad de sentirse seguro y con más beneficios, ya que podría realizarse desde puntos alejados de la misma. Para poder realizar una gestión de seguridad de una vivienda mediante un dispositivo móvil con acceso a internet se trabajó con herramientas en primer lugar que realizan la recolección de datos del ambiente, como lo sensores, dichos datos son procesados reduciendo el grado de incertidumbre mediante lógica difusa, las cuales se guardan en una base de datos y son enviadas a través de agentes móviles al dispositivo móvil en la cual existe una aplicación desarrollada en Android para tratar este mensaje y mostrarlo al usuario de la manera más sencilla posible. De esta manera el trabajo de investigación realiza una integración de diferentes tecnologías como ser: circuitos con la placa Arduino UNO, modelo de sistemas de control con Simulink, agentes móviles y base de datos PostgreSQL. Gracias a la integración e interacción de las tecnologías mencionadas, se logra realizar un sistema de control inteligente de seguridad de una vivienda, la cual será de provecho para el usuario que desee realizar un seguimiento al estado actual en el que se encuentra su vivienda, a través de un dispositivo móvil que podría ser su celular.

Palabras clave: Vivienda, seguridad, dispositivos móviles.

SUMMARY

Security is the absence of danger or risk feeling of confidence you have in something or someone, keep this concept in housing is vital because it is the place that protected the person share their habitat. With respect to technological advance the proposal focuses on mobile devices is currently available to much of the population. Thus if this capacity is used to provide the security requested in housing would be of great support to the company to meet the need to feel safe and more benefits, since it could be performed from points far from it. To perform security management of a home through a mobile device with internet access worked with tools first conducting data collection environment, such as sensors, these data are processed by reducing the uncertainty by fuzzy logic , which are stored in a database and are sent through mobile agents to the mobile device in which there is an Android application developed to treat this message and show the user the easiest way possible. In this way the research performed an integration of different technologies such as: circuit with Arduino UNO model with Simulink control systems, mobile agents and PostgreSQL database. With the integration and interaction of the above technologies, it is possible to perform an intelligent control system security of a home, which will be of benefit to the user who want to track the current state in which it finds its housing, through a mobile device that could be your phone.

Keywords: housing, security, mobile devices.

ÍNDICE GENERAL

	Pág.
CAPITULO I	
MARCO INTRODUCTORIO DE FUNDAMENTOS	
1.1. Introducción.....	1
1.2. Antecedentes.....	5
1.3. Análisis del problema	7
1.3.1. Planteamiento del problema	7
1.3.2. Formulación del problema	9
1.4. Objetivos.....	9
1.4.1. Objetivo general	9
1.4.2. Objetivos específicos.....	10
1.5. Hipótesis	10
1.6. Operacionalización de variables.....	11
1.6.1. Variable dependiente	11
1.6.2. Variable independiente	12
1.7. Variables de investigación	14
1.7.1. Variable dependiente	14
1.7.2. Variable independiente	14
1.8. Justificación.....	14
1.8.1. Justificación social	14
1.8.2. Justificación económica.....	15
1.8.3.- Justificación técnica	16
1.9. Límites	16
1.10. Alcance	17
1.11. Aporte	17
1.12.- Metodología de investigación	17
1.12.1. Enfoque cuantitativo.....	17
1.12.2. Modalidades de la investigación.....	18
1.12.3. Tipo de investigación	18
1.12.4. Diseño de la investigación.....	18

1.12.5. Instrumentos de recolección de datos.....	19
1.13. Metodología de desarrollo de software	20
1.13.1. Herramientas.....	20

CAPITULO 2

MARCO TEÓRICO

2.1. Sistema integrado	22
2.2. Sistema de seguridad	22
2.3. Control inteligente	23
2.3.1. Técnicas básicas de control inteligente	24
2.4. Lógica difusa	25
2.4.1. Conjuntos difusos	26
2.4.2. Operaciones sobre conjuntos difusos	27
2.4.3. Sistema basado en lógica difusa	27
2.4.4. Métodos directos de lógica difusa	28
2.5. Arduino	29
2.5.1. Tipos y características	30
2.6. Sensores	31
2.7. Sensor de movimiento	35
2.7.1. Características	35
2.8. Sensor de color	36
2.8.1. Características	36
2.9. Sensor de temperatura	37
2.9.1 Características	37
2.10. Aplicación móvil	37
2.10.1. Tipos de aplicaciones móviles	38
2.11. Android.....	40
2.11.1 Arquitectura de Android.....	41
2.11.1.1 El núcleo de Linux.....	41
2.11.1.2 Runtime de Android	42
2.11.1.3 Bibliotecas Nativas.....	42

2.11.1.4 Entorno de aplicación	43
2.11.1.5 Aplicación.....	44
2.12. Agentes	44
2.12.1. Agentes móviles	46
2.13. Plataforma de agentes Java.....	46
2.13.1. Paquete de agentes Java.....	47
2.13.2. Componentes principales de la plataforma de agentes	48
2.13.3. Lenguaje de comunicación	49
2.13.4. Plataforma de agentes para Android.....	50
2.13.5. Entorno de ejecución	51
2.14. Matlab.....	51
2.14.1 Entorno de Matlab	52
2.15. Simulink	52
2.15.1 Elementos de Simulink.....	53
2.16 Base de datos	54
2.16.1 Características de PostgreSQL	55
2.17 Ingeniería del software	56
2.17.1 Proceso, métodos y herramientas	56
2.17.2 Una visión general de la ingeniería del software.....	57
2.17.3 Casos de uso	58
2.17.4 El modelo de clases	59
2.17.4.1 Generalización.....	59
2.17.5 Diagrama de secuencia.....	60
2.18 Metodología Scrum	60
2.18.1 Componentes de Scrum.....	62
2.18.1.1 Las reuniones.....	62
2.18.1.2 Los roles	63
2.18.2 Elementos de Scrum.....	63
2.18.2.1 Product Backlog	64
2.18.2.2 Sprint Backlog.....	64
2.18.3 Incremento.....	64

CAPITULO 3

MARCO APLICATIVO

3.1 Definición de requerimientos	65
3.2 Análisis.....	66
3.2.1 Casos de Uso	66
3.2.2 Diagrama de secuencia	69
3.2.3 Diagrama de clases	70
3.2.3 Identificación de agentes	70
3.2.4 Información de despliegue del agente.....	71
3.2.5 Identificación de elementos hardware.....	71
3.3 Diseño y arquitectura	72
3.3.1 Arquitectura del prototipo.....	72
3.3.1.1 Adquisición de datos.....	73
3.3.1.2 Control inteligente.....	73
3.3.1.3 Comunicación al usuario final.....	73
3.3.2 Arquitectura de componentes electrónicos de control se seguridad	74
3.3.3 Diseño de bloques de control se seguridad de vivienda en Simulink.....	74
3.3.4 Diseño de interfaz de usuario.....	76
3.4 Desarrollo	80
3.4.1 Plan de ejecución de desarrollo.....	81
3.4.1.1 Diagrama de actividades planificado.....	81
3.4.2 Asignación de roles	83
3.4.3 Pila de Productos	83
3.4.4 Historias de Usuario	84
3.4.4 Sprint 1- Sprint 3.....	89
3.4.5 Sprint 4.....	90
3.4.6 Sprint 5	91
3.4.7 Sprint 6.....	92
3.4.8 Sprint 7	100
3.4.9 Sprint 8.....	100

3.4.10 Sprint 9.....	101
3.5 Pruebas	102
3.6 Prueba de Hipótesis	104

CAPITULO 4

CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones.....	107
4.2 Estado de los objetivos.....	107
4.2.1 Estado del objetivo general	107
4.2.2 Estado de los objetivos específicos	107
4.3 Estado de la Hipótesis	108
4.4 Recomendaciones.....	108

Bibliografía

Anexos

Anexo A, Árbol de problemas

Anexo B, Árbol de objetivos

Anexo C, Cuestionario 1

Anexo D, Cuestionario 2

Documentación

ÍNDICE DE FIGURAS

	Pág.
Figura 1.1 Inseguridad ciudadana y criminalidad en Bolivia.....	2
Figura 1.2 Estratificación social a escala nacional por nivel socioeconómico.....	14
Figura 2.1 Lógica difusa Vs lógica clásica.....	26
Figura 2.2 Sistema basado en lógica difusa.....	28
Figura 2.3 Placa Arduino UNO.....	30
Figura 2.4 Sensor de movimiento PIR HC-SR501.....	36
Figura 2.5 Sensor de color TCS3200.....	36
Figura 2.6 Sensor de temperatura LM35.....	37
Figura 2.7 Arquitectura de Android.....	41
Figura 2.8 Diagrama de agente en interacción con el ambiente.....	44
Figura 2.9 Clasificación de agentes inteligentes.....	45
Figura 2.10 Interfaz principal de agentes Jade.....	47
Figura 2.11 Entorno de Matlab.....	52
Figura 2.12 Capas de la ingeniería del software.....	56
Figura 2.13 Clase descrita en UML.....	59
Figura 2.14 Diagrama de secuencia.....	60
Figura 2.15 Ciclo principal de Scrum.....	61
Figura 3.1 Diagrama de casos de usos del sistema integrado.....	66
Figura 3.2 Diagrama de secuencia del sistema integrado.....	69
Figura 3.3 Diagrama de clases del sistema integrado.....	70
Figura 3.4 Diagrama de implementación.....	71
Figura 3.5 Estructura funcional del sistema integrado.....	72
Figura 3.6 Estructura de implicados en el modelo.....	73
Figura 3.7 Arquitectura de componentes electrónicos del sistema integrado de seguridad.....	74
Figura 3.8 Modelo de bloques del sistema de control.....	75
Figura 3.9 Icono de la aplicación de usuario.....	76
Figura 3.10 Interfaz de informe.....	77
Figura 3.11 Interfaz de encendido de luces.....	78

Figura 3.12 Interfaz de historial	79
Figura 3.13 Interfaz de Usuarios	80
Figura 3.14 Función de pertenencia de la variable de estado: out.....	93
Figura 3.15 Función de pertenencia de la variable de estado: s2	93
Figura 3.16 Función de pertenencia de la variable de estado: s3	94
Figura 3.17 Función de pertenencia de la variable de estado: s1	94
Figura 3.18 Función de pertenencia de la variable de estado: s0	94
Figura 3.19 Función de pertenencia de la variable de control: Variación de luz.	95
Figura 3.20 Función de pertenencia de la variable de estado: Temperatura	97
Figura 3.21 Función de pertenencia de la variable de control: Variación de temperatura	97
Figura 3.22 Función de pertenencia de la variable de estado: Movimiento	99
Figura 3.23 Función de pertenencia de la variable de control: Variación de temperatura	99
Figura 3.24 Interfaz de trabajo Jade	101
Figura 3.25 Implementación del control, de seguridad en la presentación de una Vivienda	102

ÍNDICE DE TABLAS

	Página
Tabla 1.1 Presupuesto de instalación de equipamiento domótico en euros.....	3
Tabla 1.2 Presupuesto de instalación de equipamiento domótico en bolivianos.....	4
Tabla 1.3 Operacionalización de variable dependiente.....	11
Tabla 1.4 Operacionalización de variable independiente.....	12
Tabla 2.1 Tipos y características de Arduino.....	30
Tabla 2.2 Ventaja e inconvenientes de una aplicación nativa.....	39
Tabla 2.3 Ventajas en inconvenientes de una aplicación Web.....	39
Tabla 2.4 Ventajas en inconvenientes de una aplicación híbrida.....	40
Tabla 3.1 Caso de uso: Iniciar Sesión.....	67
Tabla 3.2 Caso de uso: Solicitar información estado vivienda.....	67
Tabla 3.3 Caso de uso: Activar/desactivar actuadores.....	67
Tabla 3.4 Casos de uso: Historial de actividades.....	68
Tabla 3.5 Casos de uso: ver usuarios.....	68
Tabla 3.6 Descripción del entorno del agente.....	71
Tabla 3.7 Identificación de los elementos hardware.....	72
Tabla 3.8 Diagrama de Gantt: Análisis preliminar e identificación del problema.....	82
Tabla 3.9 Diagrama de Gantt: Desarrollo del proyecto.....	82
Tabla 3.10 Diagrama de Gantt: Evaluación y obtención de resultados.....	82
Tabla 3.11 Lista de comprometidos con el desarrollo del sistema.....	83
Tabla 3.12 Lista de involucrados con el desarrollo del sistema.....	83
Tabla 3.13 Pila de productos Scrum.....	83
Tabla 3.14 Historia de Usuario 1: Investigar elementos hardware para adquisición de datos.....	84
Tabla 3.15 Historia de Usuario 2: Investigar, testear placa Arduino UNO.....	84
Tabla 3.16 Historia de Usuario 3: Investigar sobre bibliotecas y controladores para conexión Matlab-Simulink y Arduino UNO.....	85
Tabla 3.17 Historia de Usuario 4: Investigar lógica difusa.....	85
Tabla 3.18 Historia de Usuario 5: Investigar bibliotecas sobre Matlab-Simulink y lógica difusa.....	85

Tabla 3.19 Historia de Usuario 6: Investigar sobre bibliotecas y controles para conexión Matlab-Simulink.	86
Tabla 3.20 Historia de Usuario 7: Investigar sobre bibliotecas y controles para conexión Matlab-Java.....	86
Tabla 3.21 Historia de Usuario 8: Investigar sobre bibliotecas y controles para conexión Java-PostgreSQL.	86
Tabla 3.22 Historia de Usuario 9: Investigar sobre agentes Java.....	87
Tabla 3.23 Historia de Usuario 10: Investigar sobre agente Java y aplicaciones Android.....	87
Tabla 3.24 Historia de Usuario 11: Diseñar circuito de control en Arduino uno.....	87
Tabla 3.25 Historia de Usuario 12: Diseñar modelo de control en Simulink.....	88
Tabla 3.26 Historia de Usuario 13: Control difuso del sensor de temperatura, sensor de luz, y sensor movimiento.....	88
Tabla 3.27 Historia de Usuario 14: Base de datos.....	88
Tabla 3.28 Historia de Usuario 15: Agente de control en la computadora	89
Tabla 3.29 Historia de Usuario 16: Agente de comunicación a usuario en aplicación Android.....	89
Tabla 3.30 Cumplimiento de requerimientos funcionales.....	103



CAPITULO I

MARCO INTRODUCTORIO DE

FUNDAMENTOS



1. FUNDAMENTOS

Resumen.

En el presente capítulo se describe la problemática que motivó el desarrollo de este trabajo de investigación, los objetivos a alcanzar, la hipótesis, justificación, metodología, herramientas, límites, alcance y aportes.

1.1 INTRODUCCIÓN

Se puede entender que una vivienda esta “Diseñada para proteger a los humanos de las inclemencias del tiempo y los depredadores, es considerada un área privada, a cuyo espacio tienen acceso ciertas personas y otros lo tienen prohibido” (Lestel, 2001: 69), una definición más extendida es la del Instituto Nacional de Estadística, Geografía e Informática de la república de México (INEGI) que señala: “Vivienda. Espacio delimitado normalmente por paredes y techos de cualquier material, con entrada independiente, que se utiliza para vivir, esto es, dormir, preparar los alimentos, comer y protegerse del ambiente” (INEGI, 2001).

Según las definiciones nombradas en párrafos anteriores, para la mayoría de la gente, el hogar significa un lugar de serenidad y tranquilidad. Desafortunadamente, esto no se da en todas partes. Esto se debe a que “la delincuencia es un problema que afecta a la sociedad en general, es por esto que para salvaguardar bienes y personas se ha tratado de implementar distintos mecanismos de seguridad tales como las alarmas que emiten mensajes acústicos, los cuales a lo largo del tiempo se volvieron obsoletas puesto que no realizan una actividad inteligente de seguridad en beneficio de los propietarios cuando ellos no se encuentran en la vivienda.

La Encuesta de Victimización, del Observatorio de Seguridad Ciudadana del Ministerio de Gobierno de Bolivia, indica que las denuncias de robo en viviendas o negocios se incrementan diariamente, los lugares de mayor ocurrencia de robos a personas son los barrios (45,3%), el centro de la ciudad (21%), en ferias y mercados (11,1%) y vehículos de transporte público (10,5%).”, tal como se puede ver en la Figura 1.1. (Oporto, 2012)

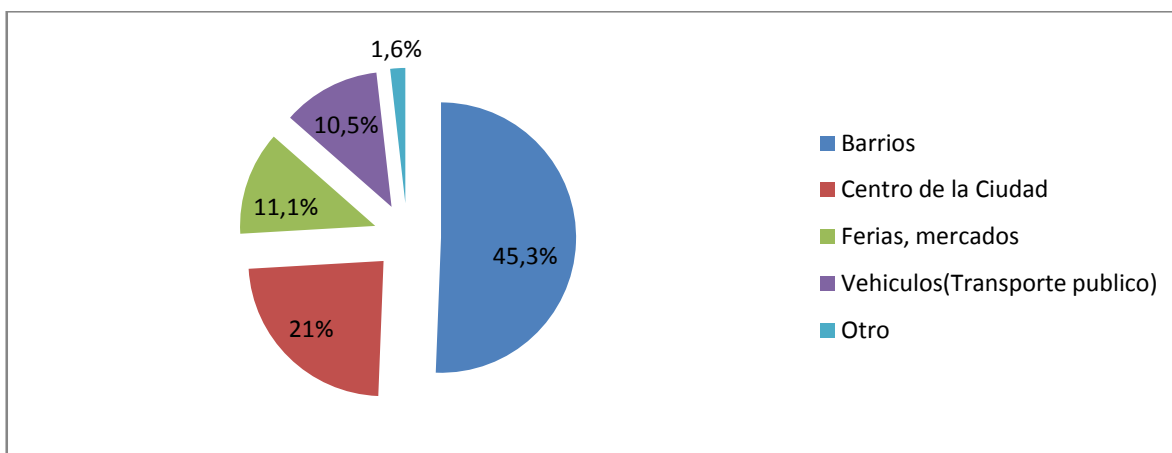


Figura 1.1: Inseguridad Ciudadana y Criminalidad en Bolivia.

Fuente: Elaborado con base en (Oporto, 2012).

Según las normas técnicas de vivienda de Bolivia, se puede entender que una vivienda “Es toda construcción destinada al uso residencial, que otorga a sus habitantes protección contra el intemperismo, seguridad y privacidad” (MOPSV, 2013).

En países desarrollados el deseo de cumplir con estas condiciones lleva al sector de la vivienda a la incorporación de las nuevas tecnologías relacionadas con la domótica las cuales proporcionan seguridad y control a edificios y viviendas inteligentes de manera automatizada logrando que el público acabe acostumbrándose a las comodidades que le brindan estas innovaciones.

Pasada la primera década del siglo XXI en los hogares y oficinas se está adoptando nuevas tecnologías que proporcionarán inteligencia ambiental, siempre tomando al ser humano en el centro de atención, las cuales se encuentran “rodeadas de un entorno digital, que responde de manera adaptativa a necesidades y hábitos que facilitan la vida diaria en el hogar” (Advisory, 2003), ejerciendo el control sobre tareas como: Abrir puertas, encender luces, indicar alternativas de programas de televisión, pagar el autobús, entre otras.

El sector de la vivienda es sin duda uno de los sectores que más recientemente se ha incorporado a las nuevas tecnologías. Sin embargo en Bolivia aún no se cuenta con esta tecnología avanzada y una de las principales causas del limitado desarrollo tecnológico es la falta de presupuesto, ya que si se planifica automatizar una vivienda es necesario realizar una

inversión fuerte como se observa en la Tabla 1.1. “En esta tabla se encuentra información sobre el costo medio de los componentes Domintell¹ necesarios para realizar la instalación domótica organizados en tres niveles de equipamiento, los precios son orientativos y obtenidos con base en la media de proyectos Domintell realizados en los años 2002-2012 en Europa, la tabla no incluye costos de instalación ni el impuesto al valor agregado (IVA), diferentes en cada país; para realizar este tipo de trabajo, se requiere siempre de un estudio personalizado que no está incluido en el costo de la tabla” (Domintell, 2012). De retorno a la realidad boliviana, según Oporto (2012), Figura 1.1, “cerca del 58% de las víctimas de robo en la vivienda o negocio son personas de bajos ingresos, mientras que el 14% pertenecen a estratos de ingresos altos”, esto causa que no todos puedan gozar de esta tecnología debido a que es costosa y debe cumplir ciertos requisitos.

Tabla 1.1: Presupuesto de instalación de equipamiento domótico calculado en euros

Fuente: Elaborado con base en (Domintell, 2012).

Tipo de vivienda	Equipamiento estándar [Euros]	Equipamiento lujoso [Euros]	Equipamiento Premium [Euros]
Apartamento 2 habitaciones	2.200	5.900	8.800
Unifamiliar 3 habitaciones	4.200	8.900	12.700
Chalet 4 habitaciones	9.000	12.600	18.700

Tomando en cuenta la Tabla 1.1 se puede hacer una estimación en pesos bolivianos con un tipo de cambio de Euro=Bs 7.47945, que se observa en la Tabla 1.2²

¹ División de la empresa belga Trump S.A. especializada en electrónica avanzada, realiza actividad de creación de prototipos, test de componentes, ensamblaje, embalaje, almacenamiento y expedición.

² Información extraída del Banco Central de Bolivia en fecha 28 de mayo de 2015.

Tabla 1.2: Presupuesto de instalación de equipamiento domótico calculado en bolivianos.

Fuente: Elaborado con base en (Domintell, 2012).

Tipo de vivienda	Equipamiento estándar [Bolivianos]	Equipamiento lujoso [Bolivianos]	Equipamiento Premium [Bolivianos]
Apartamento 2 habitaciones	16.454,79	44.128,76	65.819,16
Unifamiliar 3 habitaciones	31.413,69	66.567,11	94.989,02
Chalet 4 habitaciones	67.315,05	94.241,07	139.865,72

Analizando la Tabla 1.2 se observa que la instalación de un equipo domótico es costoso sin dejar al lado la probabilidad de tener que agregar costos extras por el servicio de instalación, impuestos por importación y otros más que fueran necesarios, observando la Tabla 1.2 se infiere que el equipamiento estándar es el más económico pero está fuera del alcance de una familia de nivel económico medio.

Analizando la Figura 1.1 “cerca del 58% de las víctimas de robo en la vivienda o negocio son personas de bajos ingresos, mientras que el 14% pertenecen a estratos de ingresos altos” y teniendo en cuenta el detalle de la Tabla 1.2, se llega a la conclusión de que en nuestro medio el costo incurrido para optar a este tipo de tecnología es demasiado caro.

Es por ello que la presente propuesta de tesis tiene como propósito la indagación de los elementos que conforman un modelo de seguridad para viviendas así como también el diseño y construcción de un sistema integrado compuesto de los elementos como producto de la indagación, estos elementos se clasificarán en elementos de hardware y elementos de software, los cuales permitirán a un usuario tener acceso y control de la gestión de la seguridad de su vivienda, de modo tal que estas acciones puedan ser controladas de manera dinámica y remota, esta propuesta cumplirá con funcionalidades similares a las tecnologías domóticas de la inteligencia ambiental, cuidando al mismo tiempo que su costo sea accesible a los usuarios del medio local.

Es por ello que la propuesta de la tesis es investigar, modelar, diseñar y construir un sistema integrado de seguridad compuesto de elementos computacionales, que permita a un usuario tener acceso a la información y control de la gestión remota de la seguridad de la vivienda, lo cual la hace interesante en un entorno donde existen pocas aplicaciones que utilizan esta tecnología.

1.2 ANTECEDENTES

Para la presente investigación se ha realizado un análisis de proyectos desarrollados en diferentes países sobre este tema, obteniendo información que se relaciona con el tema y con el objeto de investigación.

La historia del hogar digital se empezó a considerar en la década de los años 1980, entonces se trataba principalmente de edificios terciarios, que luego fueron denominados edificios inteligentes, en el sector doméstico la integración de sistema a escala comercial se ha desarrollado más tarde coincidiendo con la evolución y despliegue de Internet, el cual empezó en 1990 en Japón, Estados Unidos y algunos países del norte de Europa, los distintos sistemas autónomos como la domótica, la seguridad y las comunicaciones, tienen una historia bastante frondosa (CEDOM, 2008).

Con la plena conexión de Internet en el hogar y en general las denominadas Tecnologías de Información y Comunicación (TIC), se ha forjado una nueva forma de entender la aplicación de las tecnologías en la vivienda, mucho más positiva, donde lo importante es el usuario y los problemas, necesidades y deseos que puede tener este con relación a su vivienda.

Por ello la tecnología adquiere un papel de soporte importante para brindar servicios y resolver problemas, necesidades y deseos que el usuario tiene con su vivienda, las cuales poseen un gran número de equipos y sistemas de seguridad automatizados.

Algunos proyectos analizados que llaman la atención respecto a la propuesta planteada son:

1. “Diseño de un sistema domótico aplicado a una clínica de hemodiálisis” en la que se señala que la domótica es un área que prácticamente está empezando abrirse mercado en la República del Ecuador específicamente en el mundo de las tecnologías de la información, sin embargo, en contra de lo que pueda parecer, al estudiarlas puede proporcionar un muy buen soporte de diseño y de soluciones a las necesidades del usuario (Miranda, 2009).
2. “Instalación domótica de una vivienda unifamiliar con el sistema de bus de Instalación Europeo EIB” en la que concluye que, el número de empresas implicadas en el sector de la automatización de viviendas es limitado, solo unas pocas aportan un catálogo amplio de productos de donde se puede escoger los precios más convenientes (García, 2009).
3. “Diseño del sistema de automatización para un edificio inteligente” en la que se señala que pese a necesitarse de una inversión económica representativa para la implementación de un sistema de este tipo, se puede concluir que el ahorro provisto por el mismo, logrará que la inversión inicial se recupere a mediano plazo (Morrillo, 2009).
4. “Sistema domótico con tecnología Eibkonnex para la automatización de servicios, confort y seguridad en la empresa Sisteldata S.A “ en la que se concluye que, dotar a la vivienda de un sistema domótico le confiere un valor añadido que hace que se convierta en una inversión de cara a la futura venta de la misma, amortizando con demasía el valor invertido, al mismo tiempo ayuda a la calidad de vida, el confort, la seguridad y la tecnología representa un campo que busca el bienestar personal y material dentro de una infraestructura inteligente (Valle, 2012).

Las investigaciones mencionadas señalan que la domótica y las casas inteligentes son nuevas tecnologías que constituyen un apoyo al cumplimiento de requerimientos y necesidades que

tienen los usuarios en una vivienda, si bien señalan que requieren una inversión cara, las mismas cumplen el trabajo de ofrecer un mejor confort y seguridad. Uno de los aspectos más sobresalientes que se debe destacar es la poca existencia de empresas que realizan este tipo de trabajos, especialmente en Bolivia existen pocas empresas que brinden este tipo de tecnología, las cuales demandan un presupuesto bastante elevado, no acorde a la economía de un habitante boliviano de ingresos económicos medio, que constituye el estrato poblacional con un gran número de habitantes

1.3 ANÁLISIS DEL PROBLEMA

1.3.1 Planteamiento del problema

La delincuencia es un problema que afecta a toda la sociedad en general, “La Encuesta de Victimización, del Observatorio de Seguridad Ciudadana del Ministerio de Gobierno, respecto a delitos contra la propiedad, indica que las denuncias de robo en viviendas o negocios llegaron al 26% de los afectados, los lugares de mayor ocurrencia de robos a personas son los barrios 45,3%, el centro de la ciudad 21%, en ferias y mercados 11,1% y vehículos de transporte público 10,5%” (Oporto, 2012), estos datos se visualizan en la Figura 1.1.

Según las normas técnicas de vivienda de Bolivia, una de las condiciones principales que debe cumplir una vivienda es la “seguridad ante agentes externos, incursión de terceros, fallas de funcionamiento, uso, manipulación, materiales inapropiados o accidentes comunes, desastres naturales previsibles y riesgos a causa de fallas en las instalaciones”(MOPSV, 2013).

En países desarrollados el deseo de cumplir con estas condiciones lleva al sector de la vivienda a la incorporación de las nuevas tecnologías relacionadas con la domótica que logren que el público pueda acostumbrarse a comodidades que le brindan estas innovaciones. La domótica que es “la incorporación al equipamiento de las viviendas y edificios de una sencilla tecnología que permita gestionar de forma energéticamente eficiente, segura y confortable para el usuario los distintos aparatos e instalaciones domésticas tradicionales que conforman una vivienda”(CEDOM, 2008), es un área de la tecnología que se ocupa de proveer elementos que acerquen a la vivienda a adquirir el calificativo de inteligente, puesto que con ello se logra

gestionar de manera adecuada la toma de decisiones para encender y apagar la iluminación, la climatización de la temperatura dentro de la vivienda, el funcionamiento de electrodomésticos, la detección de incendios, el riego automático y la seguridad de la vivienda a través de sensores de movimiento, en especial cuando no se encuentra el propietario.

Sin embargo el medio local no cuenta aún con esta tecnología avanzada y una de las principales causas del limitado desarrollo tecnológico es la falta de presupuesto, ya que al automatizar una vivienda se realiza una inversión fuerte, y “cerca del 58% de las víctimas de robo en la vivienda o negocio son personas de bajos ingresos, mientras que el 14% pertenecen a estratos de ingresos altos” (Oporto, 2012), esto provoca que no todos puedan gozar de esta tecnología debido a que son costosas y deben cumplir ciertos requisitos que no están al alcance de la mayoría de la población.

Lo mencionado se analiza en un árbol de problemas, el cual se describe en el anexo A, Figura A1, en el mismo se puede detallar que el problema central se encuentra en la escasa existencia aplicaciones de control de gestión de seguridad y acceso a la información del estado de viviendas, la cual produce los siguientes efectos:

1. Existe una limitación de la gestión de seguridad de viviendas, esto debido a que no existe un sistema que efectúe la gestión de seguridad de la vivienda y el usuario, en este caso el propietario o residente de la vivienda.
2. Desconocimiento del estado de seguridad de la vivienda, el residente de la vivienda sale del hogar con la incertidumbre de los eventos que pueden ocurrir durante el tiempo de ausencia de su hogar.
3. Viviendas vulnerables a peligros de abuso de extraños, incendios producidos en la vivienda o eventos inusuales; una vivienda sin protección es vulnerable a robos u otro tipo de eventos.

En párrafos anteriores se describieron los efectos de no contar con un sistema de gestión de seguridad de viviendas, a continuación se describe las causas que llevan a estos efectos.

1. Los recursos económicos escasos juegan un papel importante debido a que el costo de adquirir sistemas que realicen este tipo de trabajos son bastante elevados y en un país en el cual un gran porcentaje de la población no cuenta con una gran cantidad de recursos económicos es difícil realizar la instalación de un sistema de seguridad de vivienda.
2. Pocos son los hogares que cuentan con sistemas de seguridad en sus viviendas aunque no siempre son de alta tecnología.
3. Los hogares que cuentan con sistemas de alarmas que no pueden ser gestionados de manera remota; si bien existen hogares que por temor a robos implementan sistemas básicos de seguridad como alarmas no son muy útiles cuando el residente no se encuentra en el hogar debido a que el monitoreo no se realiza de manera remota, es decir, no puede ser controlada desde otro punto que no sea el propio hogar.

En el análisis realizado al árbol de problemas, Figura A1 encontrado en Anexos A, se indica que una vivienda puede contar con un control de seguridad mediante un sistema integrado que utilice elementos de hardware y software, los elementos de hardware tendrán la capacidad de recolectar información del entorno y enviar datos a dispositivos móviles a través de elementos de software que se encargarán de la comunicación, de esa manera se tendrá la gestión, monitoreo y control de seguridad de la vivienda, este proceso puede ser de gran beneficio para los usuarios con respecto a la gestión de seguridad en viviendas.

1.3.2 Formulación del problema

¿Cuáles son los elementos computacionales que conforman un modelo informático integrado para el control inteligente de seguridad de una vivienda a través de un dispositivo móvil?

1.4 OBJETIVOS

1.4.1 Objetivo general

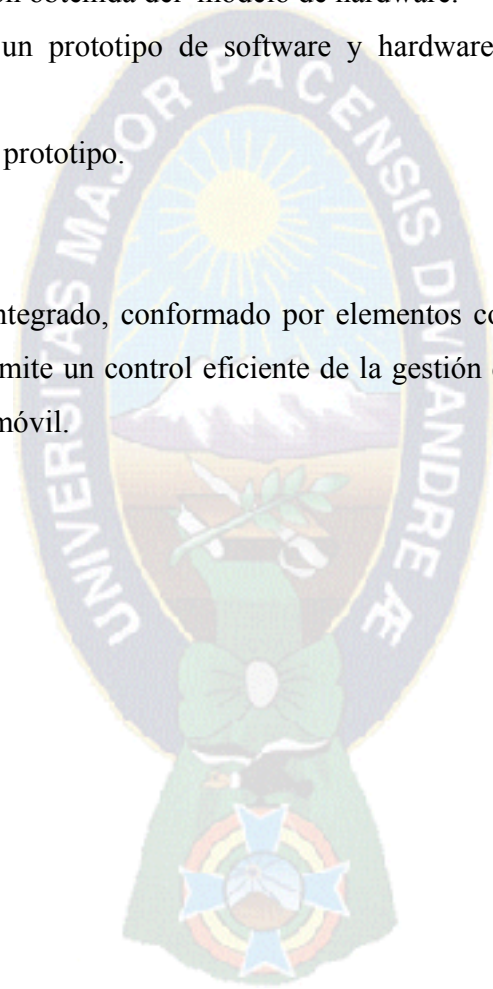
Indagar los elementos computacionales que permitan contar con un modelo informático integrado, que sea capaz de realizar el control inteligente de la seguridad en viviendas controlada a través de un dispositivo móvil.

1.4.2 Objetivos específicos

1. Buscar y seleccionar elementos computacionales de hardware y software para el diseño del prototipo de un sistema de control de vivienda.
2. Diseñar un modelo de adquisición de datos de una vivienda utilizando los elementos hardware seleccionados.
3. Diseñar un modelo de software mediado por un agente móvil que procese la información obtenida del modelo de hardware.
4. Construir un prototipo de software y hardware que implemente el modelo planteado.
5. Evaluar el prototipo.

1.5 HIPÓTESIS

El modelo informático integrado, conformado por elementos computacionales adecuados de hardware y software, permite un control eficiente de la gestión de seguridad de la vivienda a través de una aplicación móvil.



1.6 OPERACIONALIZACIÓN DE VARIABLES

1.6.1. Variable dependiente

Sistema integrado de seguridad

Tabla 1.3: Operacionalización de variable dependiente.

Fuente: Elaborado con base en (Tintaya, 2000)

Dimensión Conceptual	Dimensión Operacional	Indicador	Medidores	Escala	Instrumentos
Supervisión de la seguridad en la vivienda mediante administración remota.	Sistema Computacional	Control inteligente de la seguridad en la vivienda	Informe alarma	Reglas de producción determinada para cada sensor. Ver Figuras; 3.14, 3.15, 3.16, 3.17, 3.18, 3.22, 3.26	Lógica Difusa método Mamdani
		Comunicación y envío de información de agente de control y aplicación móvil.	Si- No	Si:100 No: 0	Historial de base de datos Aplicación móvil en funcionamiento.
		Recolección de datos de la vivienda	Sensor de color Sensor de movimiento Sensor de temperatura	Si:100 No: 0	Modelo de bloques Simulink Base de datos
		Emisión de informes de estado de la vivienda.	Si-No	Si:100 No: 0	Historial de base de datos

1.6.2. Variable independiente

Elementos computacionales

Tabla 1.4: Operacionalización de variable independiente.

Fuente: Elaborado con base en (Tintaya, 2000)

Dimensión Conceptual	Dimensión Operacional	Sub-dimensión Operacional	Indicador	Medidores	Escala	Instrumentos	
Programas y dispositivos que realizan tareas específicas.	Agente de software móvil	Propiedades	Autónomo	Si - No	Si:100 No: 0	Redes de comunicación internet, Protocolos TCP/IP	
			Envío de información	Si - No	Si:100 No: 0	Base de datos, registro de actividades	
		Rendimiento	Trabajo en tiempo real	Eficiente Deficiente	Eficiente:100 Deficiente: 0	Control de tiempo de respuesta mediante reloj externo.	
	Aplicación móvil	Usabilidad	Eficiencia	Buen funcionamiento de la aplicación	Regula de la aplicación	Bueno:100 Regular: 50 Malo: 0	Cuestionario de funcionamiento de la aplicación.
				Mal funcionamiento de la aplicación.			
		Validación	Acceso de usuario	Ingreso de usuario a sistema mediante logueo de usuario	Con acceso:100 Sin acceso: 0	Dispositivo móvil (celular, tableta) con aplicación móvil en funcionamiento.	

	Sensores	Comunicación de señales	Detección de magnitudes físicas o químicas	Si - No	Determinadas para cada sensor según la lógica difusa. Ver Figuras; 3.14, 3.15, 3.16, 3.17, 3.18, 3.22, 3.26	Tester digital, termómetro ambiental, linterna industrial. Modelo de bloques de Simulink, lógica difusa.
			Precisión	Si - No	Si:100 No: 0	Tester digital, termómetro ambiental, linterna industrial. Experimentación.



1.7 VARIABLES DE INVESTIGACIÓN

1.7.1 Variable dependiente

Modelo integrado de seguridad

1.7.2. Variable independiente

Elementos computacionales

1.8 JUSTIFICACIÓN

1.8.1 Justificación Social

La seguridad es la sensación de sentirse bien en un entorno seguro y controlado, debido a crecientes actos de delincuencia en Bolivia y pocos hábitos de seguridad en viviendas no se tiene esa sensación de tranquilidad, a su vez el nivel socioeconómico de la población no permite contar con alta tecnología para la seguridad de una vivienda, debido a su elevado costo, ver Figura 1.2.

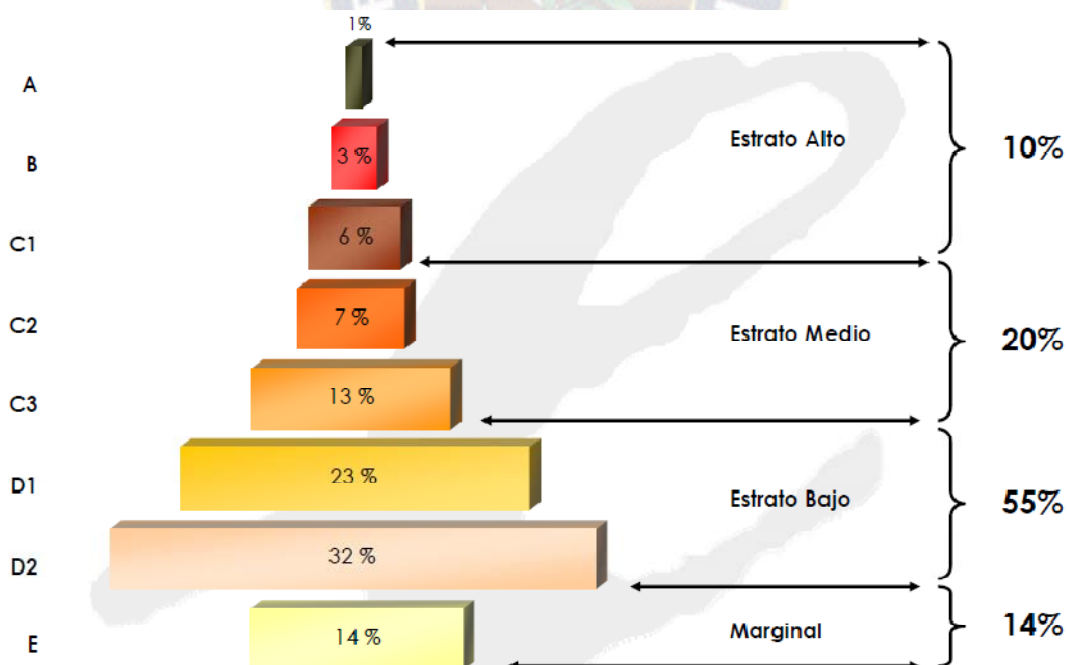


Figura 1.2: Estratificación social a escala nacional por nivel socioeconómico 2006-2007.
Fuente: (MORI, 2007)

El presente trabajo tiene un impacto social asociado, puesto que propone una herramienta con la capacidad de solucionar necesidades de los usuarios para la gestión remota de la seguridad de viviendas a través de los elementos computacionales del sistema informático integrado de seguridad de la vivienda, el cual emite alertas que son enviadas a un dispositivo móvil y son visualizadas gracias a la aplicación móvil Android desarrollada.

La aplicación móvil será capaz de realizar un control de gestión de la seguridad de las viviendas contando también con informes del estado de la vivienda desde cualquier punto, brindando facilidad en la organización de la vida cotidiana, lo cual repercutirá en la mejora de la calidad de vida, la que se basa en factores como; salud, familia, entre otros, salir de paso o viaje además de salud son factores a los cuales se colabora debido a que podrá realizar viajes o paseos familiares sin preocupaciones acerca del estado de seguridad de su vivienda teniendo como resultado menor preocupación o estrés.

1.8.2 Justificación económica

La domótica logra gestionar la iluminación, la climatización de la temperatura dentro de la vivienda, el funcionamiento de electrodomésticos, la detección de incendios, el riego automático y la seguridad de la vivienda a través de sensores de movimiento cuando no se encuentra el propietario.

Sin embargo automatizar una vivienda es una inversión fuerte debido a que los equipos y sistemas son bastante costosos y deben cumplir ciertos requisitos que no están al alcance de todos. Bolivia es un país que en su mayoría está conformada por una población “de estrato medio en un 20% y por un estrato bajo en un 55%” (MORI, 2007), lo que imposibilita el uso de tecnologías de este tipo debido a su elevado costo. Ver Figura 1.2.

La presente propuesta intenta conseguir un sistema domótico de seguridad de bajo costo para que los usuarios puedan utilizarlo. Con el modelo del proyecto se contara con un prototipo barato y fácil de usar.

1.8.3 Justificación técnica

El avance en las tecnologías ha permitido desarrollar una nueva tecnología en sistemas distribuidos denominada computación móvil, la tecnología de agentes móviles cobra especial importancia en entornos inalámbricos, donde la computación y los recursos de comunicación son necesarios.

Por otro lado está la creación de ambientes saturados de computación y comunicación inalámbrica, integrados en forma natural a la actividad humana llamada computación ubicua, la cual integra dispositivos inteligentes que tienen influencia sobre el ambiente permitiendo un entorno inteligente. Mark Weiser (1988), enfatiza la computación ubicua como: “realza el uso de la computadora, ya que hace posible que muchas computadoras estén disponibles en un ambiente físico, y a la vez, de forma invisible al usuario.”

Con este concepto se tiene en cuenta que la tecnología forma parte importante en el diario vivir de una persona debido a que se recurre a la tecnología en diferentes aspectos como la comunicación, seguridad, lo cual lleva a investigaciones que mejoren estos servicios integrando diferentes tecnologías, lo propio se realiza para mejorar la seguridad del refugio llamado vivienda.

La integración de dos tecnologías como la computación ubicua y los agentes inteligentes permitirá abrir camino para el desarrollo de un sistema informático integral que será responsable de la gestión efectiva de seguridad de una vivienda, utilizando tecnologías que facilitan el diario vivir de las personas.

1.9 LIMITES

La presente propuesta plantea el desarrollo un sistema integrado capaz de realizar un control de seguridad de los elementos de hardware instalados en la vivienda, con el propósito de tener información y control de la gestión de seguridad en la vivienda, a través de elementos de software, la cual cuenta con las siguientes limitaciones:

1. Se limitara a tres elementos en la adquisición de datos del ambiente: Sensor de movimiento, sensor de color, sensor de temperatura.

2. La aplicación está limitada a ejecutarse solamente en dispositivos móviles que cuenten con el sistema operativo Android.

1.10 ALCANCES

El sistema integrado propuesto automatiza la seguridad de una vivienda utilizando tres sensores; Sensor de movimiento, sensor de color y sensor de temperatura, con los cuales se abstraen información del ambiente de la vivienda, así también utiliza puntos de luz que representan la iluminación de la vivienda. La información obtenida de los sensores, luego de pasar por un proceso de reducción de incertidumbre con lógica difusa, es enviada al usuario a través del agente móvil que utiliza una aplicación móvil con el sistema operativo Android.

1.11 APORTE

El aporte del presente trabajo de tesis es la indagación, búsqueda, identificación, modelado y construcción de un sistema informático integrado, el cual está compuesto de elementos computacionales, enfocados a realizar la gestión de seguridad inteligente de una vivienda a través de un dispositivo móvil.

1.12 METODOLOGÍA DE LA INVESTIGACIÓN

1.12.1 Enfoque cuantitativo

Usa la recolección de datos con base en la medición numérica y el análisis estadístico, para establecer patrones de comportamiento y probar teorías. Asimismo, brinda una gran posibilidad de réplica y un enfoque sobre puntos específicos de tales fenómenos, además de que facilita la comparación entre estudios similares³.

El tipo de enfoque que se dará a la presente investigación es de tipo cuantitativo (Tintaya, 2000), es decir se realizara de manera experimental con técnicas de diseño de experimentos,

³ Metodología de la investigación, Roberto Hernández Sampieri - Dr. Carlos Fernández Collado - Dra. Pilar Baptista Lucio

observación y cuestionarios, porque la solución del problema involucra investigación en aspectos tecnológicos y la aplicación de los mismos a la seguridad de las viviendas.

1.12.2 Modalidades de la investigación

La investigación bibliográfica-documental tiene el propósito de conocer, comprar, ampliar profundizar y deducir diferentes enfoques, teorías, conceptualizaciones y criterios de diversos autores sobre una cuestión determinada, basándose en documentos de investigación (fuentes primarias), o en libros, revistas, periódicos y otras publicaciones (fuentes secundarias). Para el entendimiento de los factores estudiados y el análisis de los datos obtenidos se tuvo que examinar obras referentes al tema, de entre las cuales se destacan: Libros, páginas Web, textos especializados en el tema de investigación.

1.12.3 Tipo de investigación

Por el grado de profundidad, el presente estudio corresponde al tipo de investigación descriptivo-analítico, debido a que se realiza según el análisis y alcance de los resultados obtenidos del experimento que se efectúa a la vivienda con el sistema integrado, “buscar, estudiar tantas manifestaciones del fenómeno como sea posible” (Hernández, Fernández y Baptista, 1995)

1.12.4 Diseño de la investigación

La investigación tendrá el tipo de diseño experimental, es decir, se organiza la muestra en dos grupos; Grupo de control y grupo de estudio experimental.

- ✓ **Grupo de control**, no se aplica el sistema integrado de seguridad en la vivienda.
- ✓ **Grupo de estudio experimental**, se aplica los elementos de hardware y software en la vivienda y se observa los efectos del sistema integrado de seguridad.

Luego con base en los resultados se detalla los efectos existentes en cada grupo, analizando los detalles que ayudan a comprobar la hipótesis planteada por presente tesis.

1.12.5 Instrumentos de recolección de datos

“Las técnicas de recolección de datos permiten construir y seleccionar instrumentos que permiten obtener datos del objeto de investigación”⁴. Las técnicas utilizadas serán:

- Observación: Permite obtener datos sobre el funcionamiento de la propuesta de sistemas integrado de seguridad.
- Cuestionario: Conjunto de preguntas que obtienen información para la investigación, a diferencia de la entrevista se realizan preguntas dicotómicas, de las cuales se obtiene información de cada usuario.

El cuestionario está compuesto por seis preguntas perfectamente claras y dirigidas a obtener información verídica, oportuna y relevante para la resolución del problema originalmente planteado.

El cuestionario está compuesto por preguntas estructuradas “son las que especifican en forma previa el grupo de alternativas de respuesta y el formato de las respuestas. Una pregunta estructurada puede ser de opción múltiple dicotómica o de escala” (Malhotra, 1997).

Para este cuestionario se utiliza la escala de Likert, que lleva el nombre de su creador Rensis Likert quien explica que “es una escala de clasificación que pide a los entrevistados que indiquen un grado de acuerdo o desacuerdo” (Malhotra, 1997), por lo regular una escala de Likert tiene cinco categorías de respuesta. Existen dentro del cuestionario preguntas cerradas, con opciones de respuesta sí y no. El cuestionario aplicado, para esta investigación consta de seis partes:

1. Establecer el perfil del usuario o afectado.
2. Evaluar el problema.
3. Evaluar y entender el ambiente de usuario.
4. Evaluar las necesidades de la aplicación en el ambiente.
5. Evaluar las soluciones.
6. Evaluar la oportunidad.

⁴ Metodología de Investigación Tecnológica

1.13 METODOLOGÍA DE DESARROLLO DEL SOFTWARE

El desarrollo del prototipo se realiza con el procedimiento de la ingeniería de software el mismo consta de los siguientes elementos:

- I. Requerimientos, se identifican los requerimientos necesarios para la elaboración del prototipo.
- II. Análisis requerimientos. Se diseñan los modelos que son de apoyo para el análisis para la elaboración del prototipo.
- III. Diseño, se realiza la estructura del prototipo, se realiza la estructura del circuito, el modelo de bloque de Simulink.
- IV. Desarrollo y programación, se realiza el desarrollo con la metodología ágil Scrum, debido a la cualidad de ser evolutivo incremental.
- V. Pruebas. Se detalla el formato de elaboración de pruebas para el prototipo.
- VI. Mantenimiento, se centra en el cambio que va asociado a la corrección de errores, a las adaptaciones requeridas a medida que evoluciona el entorno del software.

1.13.1 Herramientas

- Herramientas de software
 - ✓ Eclipse Juno, se utiliza para la programación del agente de control.
 - ✓ Adt eclipse. Se utiliza para la elaboración de la aplicación móvil en Android.
 - ✓ Plataforma Jade, Para la comunicación entre agentes es necesario contar con una plataforma, en este caso Jade
 - ✓ Matlab - Simulink. Para realizar el procesamiento de datos extraído del mundo natural y realiza un proceso de reducción de incertidumbre mediante lógica difusa.
 - ✓ PostgreSQL, Se diseñó un modelo de datos y para guardar datos manipulados en el prototipo.

- Componentes hardware

- ✓ Placa Arduino UNO
- ✓ Sensor de Color
- ✓ Sensor de temperatura
- ✓ Sensor de movimiento
- ✓ Protoboard
- ✓ Cables de conexión
- ✓ Luces





CAPITULO II

MARCO TEÓRICO



2. MARCO TEÓRICO

Resumen

En este capítulo se presenta el marco conceptual y las definiciones relacionadas a la investigación, entre ellas se encuentran; Sistema integrado, lógica difusa, Android, Arduino UNO, Simulink, sensor de movimiento, sensor de temperatura, sensor de color, plataforma de agentes java.

2.1 SISTEMA INTEGRADO

Según Puleo un sistema se define como " un conjunto de entidades caracterizadas por ciertos atributos, que tienen relaciones entre sí y están localizadas en un cierto ambiente, de acuerdo con un cierto objetivo", así mismo "un sistema integrado lleva a cabo una o unas pocas tareas predefinidas, generalmente con requerimientos muy específicos, y a menudo incluye hardware específico para cada tarea" (Alegsa, 2013), según la Guía Nacional para la Simplificación, el Mejoramiento y la Racionalización de Procesos, Procedimientos y Trámites, expedida por el DAFP y la Alcaldía Mayor de Bogotá un sistema integrado de gestión "es un conjunto de actividades mutuamente relacionadas que tienen por objetivo orientar y fortalecer la gestión, dar dirección, articular y alinear conjuntamente los requisitos de los Subsistemas que lo componen Gestión de Calidad, Modelo Estándar de Control Interno, Gestión Ambiental, Gestión de Desarrollo Administrativo, entre otros".

2.2 SISTEMA DE SEGURIDAD

Se puede indicar que un sistema de seguridad "es un conjunto de dispositivos colocados estratégicamente en el perímetro de un sitio específico para detectar la presencia, irrupción, o invasión de un desconocido o de un individuo que no posea un acceso permitido" (Maquinariapro, 2014) Estos equipos avisarán cuando el sistema fue violado mediante un ruido característico o señal aguda, a su vez, el sistema de seguridad puede estar conectado a una central de vigilancia privada para que al cabo de pocos minutos personal policial se haga presente en nuestra ayuda.

Actualmente el mercado nos ofrece infinidad de alternativas a la hora de proteger la casa de las personas están: Las alarmas simples con sistema acústico cuyo objetivo es espantar a los posibles ladrones, este tipo de alarmas no cuenta con conexiones externas. Los detectores volumétricos captan el movimiento de cualquier cuerpo u objeto en el campo de actuación determinado por la longitud de onda; los detectores por ultrasonido detectan los ruidos extraños como roturas de cristales y es por eso que suelen colocarse en las ventanas. Se Tiene también los detectores infrarrojos que analizan las variaciones térmicas que se producen en un radio específico, al intruso lo delatarán sus 37°C de su cuerpo; los más modernos incorporan detectores volumétricos. Por último, los sistemas de seguridad actuales poseen contactos magnéticos para puertas y ventanas, estas son placas imantadas que generas un campo magnético que hace saltar la alarma cuando alguien intenta abrir la ventana.

2.3 CONTROL INTELIGENTE

La línea de investigación de Control Inteligente nace con la intención de aplicar a control las técnicas de Inteligencia Artificial, las cuales comenzaban a tener una serie de resultados y se consolidaban como disciplina.

La "inteligencia artificial" en sí es un campo amplio, que abarca lógica, optimización, probabilidad, percepción, razonamiento, toma de decisiones y aprendizaje. El objetivo es integrar agentes inteligentes en los sistemas de control.

De forma un tanto recíproca, la teoría de control ha influenciado a la investigación en inteligencia artificial (el concepto de "agente" es muy similar al de "regulador"). Así, desde unos comienzos basados en sistemas expertos con reglas intuitivas y métodos de aprendizaje de redes neuronales, en la actualidad el control inteligente bebe de las teorías de sistemas adaptativos, no lineales y de modelos locales y planificación de ganancia (desigualdades matriciales lineales). Asimismo, como se ha dicho, los investigadores en inteligencia artificial utilizan resultados de control y "cibernética", en particular optimización de índices de coste y programación dinámica y el concepto de estado de sistema dinámico y filtrado de Kalman. (CEA, 2010)

“El Control Inteligente es el resultado de la intersección del Control Automático, la Inteligencia Artificial y los sistemas informáticos distribuidos de tiempo real.”(Galán, 2000).

2.3.1 Técnicas básicas de control inteligente

Entre las técnicas de Inteligencia Artificial usadas en Control Inteligente destacan:

- a. **Sistemas expertos:** Basados en el uso de las técnicas y herramientas de diseño de sistemas expertos de Inteligencia Artificial. (Alamán, 1992)

En este tipo de sistemas la base de conocimientos de control se obtiene de un experto humano: El operador del proceso a controlar. En ella se recoge esencialmente dos tipos de información: Reglas referentes a la interpretación del estado del proceso y reglas para la determinación de las actuaciones. Para aumentar la eficiencia las reglas se clasifican en grupos o meta reglas.

- b. **Lógica difusa:** Se trata de un control basado en reglas que utiliza técnicas para manejar la imprecisión. Cabría separar el estudio de los controladores borrosos como alternativa al control adaptativo, predictivo u otros del control experto que utiliza incertidumbre.

Los reguladores borrosos tratan de implantar estrategias de control expresadas en términos lingüísticos por los operadores de proceso, para ello se basan en técnicas de lógica borrosa (Mamdani, 1981)

La lógica borrosa ha alcanzado un notable desarrollo tanto en el estudio formal (Matía, 92) como en el de aplicaciones y herramientas para diseño disponibles.

- c. **Redes neuronales:** Una red neuronal es, siguiendo a Hecht-Nielsen (Hecht-Nielsen 1988), "una estructura de procesamiento de información paralela y distribuida, formada por elementos de procesamiento interconectados mediante canales unidireccionales de información. Cada elemento de procesamiento tiene una conexión de salida con diferentes ramas portadoras de la misma señal. Esta señal de salida será de un tipo matemático cualquiera. Todo el procesamiento que se hace en un elemento debe ser

completamente local, por ejemplo: Dependerá solo de los valores actuales de las entradas al elemento y de posibles valores almacenados en memoria local.

Las redes neuronales en control se utilizan por su capacidad de aprender el comportamiento no lineal de las variables de un proceso. Esta capacidad se puede utilizar para el diseño de sistemas que funcionen como simulador, identificador o controlador.

- d. **Algoritmos genéticos:** Los algoritmos genéticos se están utilizando en control, entre otras aplicaciones, para depurar de forma automática las reglas que forman la base de conocimiento. Ésta se equipara a un conjunto de organismos vivos, capaces de evolucionar para adaptarse mejor al entorno. Esta adaptación se medirá a partir de la tasa de fallos y aciertos de los individuos (Holland, 1989), (Goldberg, 1989).

Al igual que en la evolución de las especies, cuando se produce una variación del entorno, sólo los que se adaptan a esa variación sobreviven, mientras que los que no pueden adaptarse son eliminados. A lo largo del tiempo van naciendo nuevos individuos que pasan a formar parte de la comunidad, con características genéticas que les hacen parecerse a sus padres, y permiten que la especie se mantenga. En algunas ocasiones se producen mutaciones que dan lugar a individuos mejor o peor adaptados.

2.4 LÓGICA DIFUSA

Este tipo de lógica es la lógica que utiliza expresiones que no son totalmente ciertas ni totalmente falsas, es decir, es una lógica aplicada a conceptos que pueden tomar un valor indeterminado de veracidad dentro de un conjunto de valores cuyos extremos son la verdad absoluta o la falsedad absoluta. Por así decirlo es una lógica que expresa la falta de definición del objeto al que se aplica. Si quiere dar una definición mucho más específica se puede definir a este tipo de lógica como una técnica de la inteligencia computacional que ayuda o permite trabajar con información que es imprecisa y no está bien definida. Pertenece a la lógica multivaluada pero la lógica difusa se diferencia de ésta en que permite introducir valores intermedios entre la afirmación completa o la negación absoluta.

2.4.1 Conjuntos difusos

Para ilustrar el concepto de la lógica difusa y los conjuntos difusos se va a explicar el primer ejemplo que puso Zadeh. Para ello puso el ejemplo del conjunto de “los hombres altos”. Según la teoría de lógica clásica al conjunto de hombres altos solo pertenecen los que miden más de una determinada altura y esa altura límite es 1.80 metros, así un hombre es considerado alto cuando mide por ejemplo 1.81 metros y uno bajo cuando mide 1.79 metros. Esto no parece una razón muy lógica para catalogar a un hombre de alto o bajo ya que por ejemplo en el caso expuesto la altura de uno a otro solo se diferencia en 2 centímetros. Ahí, en casos como este donde no es fácil catalogar algo, se introduce la lógica difusa. Según la lógica difusa, el conjunto de “hombres altos” es un conjunto que no tiene una frontera clara que indique que perteneces a ese grupo o no.

El evaluar si un hombre es alto o bajo, se hace mediante una función que define la transición entre alto a bajo y para ello asigna a las distintas alturas un valor entre 0 y 1. Según sea este valor se considera que se pertenece al conjunto o no. Aplicando esto al caso anterior, un hombre que mida 1.79 metros se puede decir que pertenece al conjunto de hombres altos con un grado de 0.75 y el hombre que medía 1.81 metros pertenece al conjunto de hombres altos con un grado de 0.8. Si se representa esto en una gráfica se obtendrá que la transición entre alto o bajo con la lógica difusa es una curva con cambios no abruptos mientras que con la lógica clásica, el paso de alto a bajo o viceversa es brusco, ver Figura 2.1.

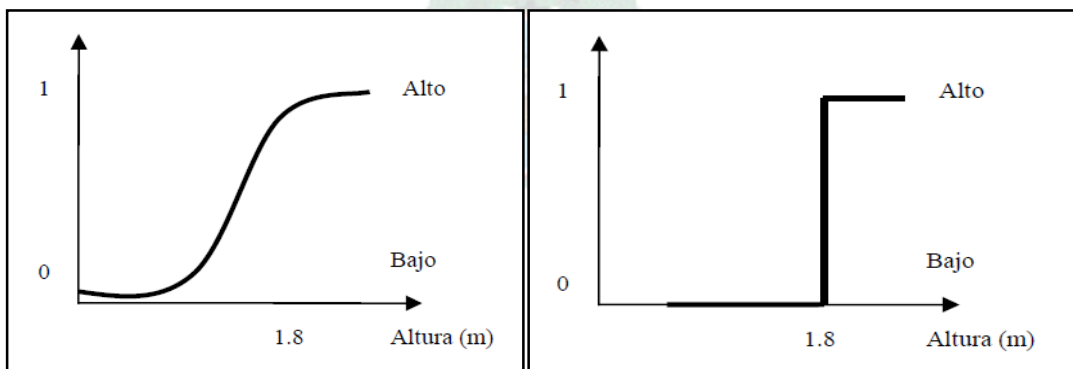


Figura 2.1: Lógica difusa Vs lógica clásica.

Fuente: (Benito, 2014)

2.4.2 Operaciones sobre conjuntos difusos

Existen seis tipos de operaciones:

1. Inclusión o subconjunto: A es un subconjunto de B:

$$\Leftrightarrow \mu_A(x) \leq \mu_B(x) \forall x$$

2. Unión: La unión de los conjuntos difusos A y B es el conjunto difuso C y se escribe como $C = A \text{ OR } B$; su función de pertenencia está dada por:

$$\mu_C(x) = \max(\mu_A(x), \mu_B(x)) = \mu_A(x) \cup \mu_B(x)$$

3. Intersección: La intersección de los conjuntos difusos A y B es el conjunto difuso C y se escribe como $C = A \text{ AND } B$; su función de pertenencia está dada por:

$$\mu_C(x) = \min(\mu_A(x), \mu_B(x)) = \mu_A(x) \cap \mu_B(x)$$

4. Negación o complemento: El complemento del conjunto difuso A, denotado por $\neg A$ o NOT A, se define como:

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x)$$

5. Producto cartesiano: Si A y B son conjuntos difusos en X e Y, el producto cartesiano de los conjuntos A y B en el espacio $X \times Y$ tiene la función de pertenencia:

$$\mu_{A \times B}(x, y) = \min(\mu_A(x), \mu_B(y))$$

6. Co-producto cartesiano: $A + B$ en el espacio $X \times Y$ tiene la función de pertenencia:

$$\mu_{A+B}(x, y) = \max(\mu_A(x), \mu_B(y))$$

Cabe añadir que las operaciones de intersección, unión y complemento cumplen al igual que en la teoría clásica de conjuntos, las propiedades asociativa, conmutativa, distributiva y las leyes de Morgan. Sin embargo, a diferencia de la teoría clásica, los conjuntos difusos no cumplen el principio de contradicción ni el de exclusión.

2.4.3 Sistema basado en lógica difusa

Un sistema basado en lógica difusa siempre estará compuesto por los siguientes bloques mostrados en la Figura 2.2.

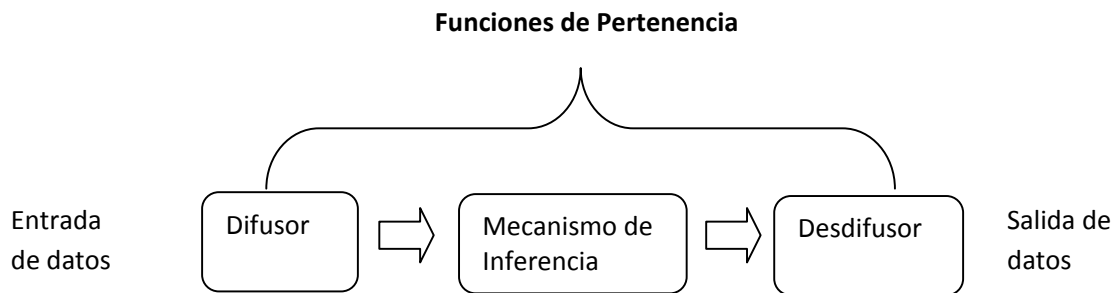


Figura 2.2: Sistema basado en lógica difusa

Fuente: (Benito, 2014)

- Bloque difusor: en este bloque a cada dato de entrada se le asigna un grado de pertenencia a cada uno de los conjuntos difusos considerados mediante la función característica ya comentada en apartados anteriores. Las entradas a este bloque son valores concretos de la variable a analizar y los datos de salida son los grados de pertenencia a los conjuntos estudiados.
- Bloque de inferencia: este bloque relaciona conjuntos difusos de entrada y de salida y representa a las reglas que definen el sistema.
- Desdifusor: en este bloque a partir de los conjuntos difusos procedentes de la inferencia se obtiene un resultado concreto mediante la aplicación de métodos matemáticos de desdifusión.

2.4.4 Métodos directos de lógica difusa

- Método directo de Mamdani: Este método emplea reglas de inferencia con la siguiente estructura:

$$IF(x \text{ is } A) \text{ and } (y \text{ is } B) THEN(z \text{ is } C)$$

$$IF(x \text{ is } A) \text{ or } (y \text{ is } B) THEN(z \text{ is } C)$$

Donde A, B y C son conjuntos difusos.

- Método directo de Takagi & Sugeno: Este método emplea reglas de inferencia con la siguiente estructura:

$$IF\ x\ is\ A\ and\ y\ is\ B\ THEN\ z = ax + by + c$$

Donde A, B son conjuntos difusos.

- Método Sugeno simplificado: Emplea reglas de inferencia con la siguiente estructura:

$$IF\ x\ is\ A\ and\ y\ is\ B\ THEN\ z = c$$

Donde A, B son conjuntos difusos y c es un valor nítido (crisp).

2.5 ARDUINO

Arduino es una plataforma de prototipos electrónica de código abierto basada en hardware y software flexibles y fáciles de usar. Está pensado para artistas, diseñadores, como hobby y para cualquiera interesado en crear objetos o entornos interactivos.

Arduino puede sentir el entorno mediante la recepción de entradas desde una variedad de sensores y puede afectar a su alrededor mediante el control de luces, motores y otros artefactos. El microcontrolador de la placa se programa usando el Arduino Programming Language (basado en Wiring1) y el Arduino Development Environment (basado en Processing2). Los proyectos de Arduino pueden ser autónomos o se pueden comunicar con software en ejecución en un ordenador (por ejemplo con Flash, Processing, MaxMSP, etc.).

Las placas se pueden ensamblar a mano o encargarse preensambladas; el software se puede descargar gratuitamente. Los diseños de referencia del hardware (archivos CAD) están disponibles bajo licencia open-source, por lo que eres libre de adaptarlas a tus necesidades. (Enríquez, 2009).

Un Arduino contiene un microchip, que es un ordenador muy pequeño que se puede programar. Se le pueden conectar sensores para que pueda medir condiciones como también puede controlar otros objetos para que reaccionen a esas condiciones. Arduino puede tomar información del entorno a través de sus pines de entrada de toda una gama de sensores y puede afectar aquello que le rodea controlando luces motores y otros actuadores.



Figura 2.3: Placa Arduino UNO
Fuente: (Arduino, 2014).

2.5.1 Tipos y características de Arduino

En la Tabla 2.1 se detalla el modelo y características de las placas Arduino más notables o utilizadas con mayor frecuencia en el medio local.

Tabla 2.1: Tipos y características de Arduino
Fuente:(Arduino, 2014)

MODELO	CARACTERÍSTICAS
Arduino UNO	<ul style="list-style-type: none"> • Microcontrolador: ATmega328 • Voltaje de funcionamiento: 5 V • Pines I/O digitales: 14 (de los cuales 6 proveen salida PWM) • Pines de entradas análogas: 6 • Corriente DC por cada pin I/O: 40 mA • Corriente DC en el pin de 3.3 V: 50 mA • Memoria Flash: 32 KB (ATmega328) de los cuales 0.5 KB son utilizados por el bootloader • SRAM: 2 KB (ATmega328) • EEPROM: 1 KB (ATmega328) • Velocidad de reloj: 16 MHz
Arduino Leonardo	<ul style="list-style-type: none"> • Microcontrolador: ATmega32u4 • Voltaje de funcionamiento: 5 V • Pines I/O digitales: 20 • Canales PWM: 7 • Pines de entradas análogas: 12 • Corriente DC por cada pin I/O: 40 mA • Corriente DC en el pin de 3.3 V: 50 mA • Memoria Flash: 32 KB (ATmega32u4) de los cuales 4 KB son utilizados por el bootloader • SRAM: 2 KB (ATmega32u4) • EEPROM: 1 KB (ATmega32u4) • Velocidad de reloj: 16 MHz
Arduino Due	<ul style="list-style-type: none"> • Microcontrolador: AT91SAM3X8E • Voltaje de funcionamiento: 3.3 V • Pines I/O digitales: 54 (de los cuales 12 proveen salida PWM)

	<ul style="list-style-type: none"> • Pines de entradas análogas: 12 • Corriente DC total en todos los pines I/O: 130 mA • Corriente DC en el pin de 5 V: 800 mA • Corriente DC en el pin de 3.3 V: 800 mA • Memoria Flash: 512 KB disponibles para las aplicaciones de usuario. • SRAM: 96 KB (dos bancos: 64KB Y 32 KB) • Velocidad de reloj: 84 MHz
Arduino Yún	<ul style="list-style-type: none"> • Microcontrolador AVR Arduino: ATmega32u4 • Voltaje de funcionamiento: 5 V • Pines I/O digitales: 20 • Canales PWM: 7 • Pines de entradas análogas: 12 • Corriente DC por cada pin I/O: 40 mA • Corriente DC en el pin de 3.3 V: 50 mA • Memoria Flash: 32 KB (de los cuales 4 KB son utilizados por el bootloader) • SRAM: 2.5 KB • EEPROM: 1 KB • Velocidad de reloj: 16 MHz • Procesador Linux: Atheros AR9331 • Arquitectura: MIPS @400MHz • Ethernet: IEEE 802.3 10/100Mbit/s • WiFi: IEEE 802.11b/g/n • USB Tipo A: 2.0 • Lector de tarjeta: sólo Micro-SD • RAM: 64 MB DDR2 • Memoria Flash:
Arduino Ethernet	<ul style="list-style-type: none"> • Microcontrolador: ATmega328 • Voltaje de funcionamiento: 5 V • Pines I/O digitales: 14 (de los cuales 4 proveen salida PWM) • Pines de entradas análogas: 6 • Corriente DC por cada pin I/O: 40 mA • Corriente DC en el pin de 3.3 V: 50 mA • Memoria Flash: 32 KB (ATmega328) de los cuales 0.5 KB son utilizados por el bootloader • SRAM: 2 KB (ATmega328) • EEPROM: 1 KB (ATmega328) • Velocidad de reloj: 16 MHz • Controlador embebido Ethernet W5100 TCP/IP • Tarjeta MicroSD, con adaptadores activos de voltaje

2.6 SENSORES

Los sensores son herramientas tecnológicas que registran cambios ambientales en función del tiempo o el tiempo que transcurre entre un hecho y otro. Los sensores analógicos miden variables tales como temperatura, luz, presión o humedad. Los sensores digitales miden si un interruptor está prendido o apagado y generalmente se usan para cronometrar o señalar acontecimientos.

Estas herramientas tecnológicas permiten recolectar y presentar datos en tiempo real. Existen una gran variedad de sensores; Los que se pueden conectar a la computadora mediante las interfaces colectoras de datos (data loggers) y las sondas inteligentes que no requieren de intermediación de una interface ya que pueden conectarse directamente a las computadoras.

Los sensores se pueden clasificar según:

1. Su principio de funcionamiento: Se encuentra dos tipos de sensores:

- Activos: Son aquellos que generan señales representativas de las magnitudes a medir en forma autónoma, sin requerir de fuente alguna de alimentación.
- Pasivos: Son aquellos que generan señales representativas de las magnitudes a medir por intermedio de una fuente auxiliar.

2. En el tipo de señal de salida se encuentran tres tipos de sensores:

- Sensores digitales: Los sensores digitales son aquellos que frente a un estímulo pueden cambiar de estado ya sea de cero a uno o de uno a cero (hablando en términos de lógica digital) en este caso no existen estados intermedios y los valores de tensión que se obtienen son únicamente dos, 5V y 10V (o valores muy próximos).
- Sensores análogos: es aquel que, como salida, emite una señal comprendida por un campo de valores instantáneos que varían en el tiempo, y son proporcionales a los efectos que se están midiendo.
- Sensores temporales: Son aquellos que entregan una señal variable en el tiempo la cual puede ser una onda sinusoidal, triangular o cuadrada.

3. El rango de valores de salida: Se cuenta con dos tipos de sensores:

- Sensores On/Off: También conocidos como sensores si-no, sensores 0-1, sensores on-off, o sensores binarios son en general dispositivos mecánicos simples, los más comunes son:
 - Interruptores Reed que se conectan por la proximidad de un imán.
 - Interruptor de péndulo, donde un peso cuelga de un hilo conductor dentro de un anillo metálico y las vibraciones o movimiento del anillo producen el cierre del circuito.

- Sensores de Medida: En estos sensores se obtiene una salida proporcional a la señal de entrada.

4. El nivel de integración: Se cuenta con tres tipos de sensores:

- Sensores discretos: Sensor en el que el circuito de acondicionamiento se realiza mediante componentes electrónicos separados e interconectados entre sí.
- Sensores integrados: Elemento sensor y circuito acondicionador (al menos este último) construidos en un único circuito integrado, monolítico o híbrido.
- Sensores inteligentes: Realiza al menos una de las siguientes funciones
 - Cálculos numéricos
 - Comunicación en red (No un punto a punto)
 - Auto-calibración y auto-diagnóstico
 - Múltiples medidas con identificación del sensor.

5. El tipo de variable medida: Se encuentran diez tipos de sensores:

- Sensores mecánicos: Son dispositivos que cambian su comportamiento bajo la acción de una magnitud física que pueden directa o indirecta. Los sensores mecánicos son utilizados para medir: Desplazamiento, posición, tensión, movimiento, presión, flujo. previamente transmitir una señal que indica cambio.
- Sensores eléctricos: Un sensor es un dispositivo capaz de detectar magnitudes físicas o químicas, llamadas variables de instrumentación, y transformarlas en variables eléctricas. Las variables de instrumentación pueden ser por ejemplo: temperatura, intensidad lumínica, distancia, aceleración, inclinación, desplazamiento, presión, fuerza, torsión, humedad, movimiento, pH.
- Sensores magnéticos: Se sirve del efecto Hall para la medición de campos magnéticos o corrientes o para la determinación de la posición.
- Sensores térmicos: Se usan para la medición precisa de la temperatura, proporcionan una indicación visual o una señal de realimentación mecánica o eléctrica que puede ser utilizada en un sistema de lazo cerrado para permitir el control automático de procesos térmicos.

- Sensores termo resistivos
- Sensores termoeléctricos
- Sensores monolíticos o de silicio
- Sensores piro eléctricos
- Sensores acústicos: Los micrófonos son los sensores que facilitan la conversión de una señal acústica en eléctrica. Se pueden aplicar diversos principios a su realización siendo la más común la combinación de fenómenos mecánico-acústicos y su conversión electromecánica.
 - Capacitivos
 - Piezoeléctricos
 - Electrodinámicos
- Sensores ultrasónicos: Los sensores ultrasónicos son interruptores electrónicos que trabajan sin contacto. La parte emisora genera pulsos de sonidos muy fuertes dentro del rango del ultrasonido.
- Sensores químicos: Los sensores químicos están formados por: Un receptor que se encarga de reconocer selectivamente a la especie química a detectar y un transductor que se encarga de convertir la señal química en señal eléctrica.
- Sensores ópticos: Un sensor óptico se basa en el aprovechamiento de la interacción entre la luz y la materia para determinar las propiedades de ésta. Una mejora de los dispositivos sensores, comprende la utilización de la fibra óptica como elemento de transmisión de la luz.
- Sensores de radiación: Son necesarios para determinar las propiedades de las partículas (E, q, m) de la radiación, fundamentales para:
 1. Información acerca de los núcleos.
 2. Protección y control de radiactividad.
- Sensores laser: Cuando se inventaron, en 1960, los láseres se calificaron como "una solución a la espera de un problema". Desde entonces, se han vuelto omnipresentes y actualmente pueden encontrarse en miles de aplicaciones, en campos muy variados, como la electrónica de consumo, la tecnología de la información, la investigación científica, la medicina, la industria y el sector militar.

2.7 SENSOR DE MOVIMIENTO

El sensor de movimiento es el modulo PIR HC-SR501, tiene 3 pines de conexión +5v, OUT (3,3v) y GND, y dos resistencias variables de calibración (Ch1 y RL2).

- Ch1: Con esta resistencia se puede establecer el tiempo que se va a mantener activa la salida del sensor. Una de las principales limitaciones de este módulo es que el tiempo mínimo que se puede establecer es de más o menos 3s. Si se cambia la resistencia por otra de 100K, se puede bajar el tiempo mínimo a más o menos 0,5 s.
- RL2: Esta resistencia variable permite establecer la distancia de detección que puede variar entre 3-7m.

La posibilidad de mantener activa la salida del módulo durante un tiempo determinado permite poder usarlo directamente para prácticamente cualquier aplicación sin necesidad de usar un microcontrolador, Ver Figura 2.4 donde se presenta el elemento electrónico.

2.7.1 Características

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



Figura 2.4: Sensor de movimiento PIR HC-SR501
Fuente: (Electronilab, 2015)

2.8 SENSOR DE COLOR

El TCS3200 es un completo sensor de color RGB con 4 diodos led incorporados y puede detectar casi un infinito rango de colores. Sus aplicaciones son varias como por ejemplo medir la luz ambiental, clasificación por color etc. Internamente el sensor tiene un array de pequeño phototransistores cada uno con varios filtros de color, lo que le permite ser bastante preciso y sensible. La salida del sensor devuelve un pulso de duración proporcional a la intensidad del color detectado, ver Figura 2.5.

2.8.1 Características

- Alimentación: 2.7V a 5.5V
- Salida digital TTL
- Escala y filtros programables
- Función power down
- Tamaño: 28.4x28.4mm



Figura 2.5: Sensor de color TCS3200
Fuente: (Electronilab, 2015)

2.9 SENSOR DE TEMPERATURA

El sensor LM35, es un sensor de temperatura integrado de precisión cuyos rangos de operación oscilan desde los -55°C hasta los 150°C , teniendo en cuenta que este tipo de sensores ofrecen una precisión de $\pm 1.4^{\circ}\text{C}$ a temperatura ambiente. Además de ello, son del tipo lineal; es decir, que no es necesario forzar al usuario a realizar conversiones debido a que otros sensores están basados en grados Kelvin. Un LM35 puede funcionar a partir de los 5 V (en corriente continua), sea por alimentación simple o por doble alimentación (+/-), ver Figura 2.6.

2.9.1 Características

- Configurado para ser leído en grados Celsius
- Factor de escala lineal de $+10\text{ mV} / ^{\circ}\text{C}$
- Rango de trabajo entre -55°C hasta 150°C
- Apropiado para aplicaciones remotas
- Bajo costo
- Funciona con tensiones entre 4 V hasta 30 V
- Menos de 60 μA de consumo
- Baja impedancia de salida, 0.1 W, para cargas de 1 mA

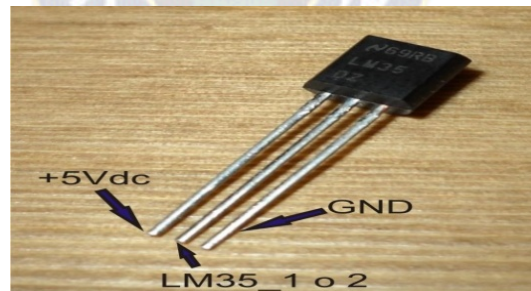


Figura 2.6: Sensor de temperatura LM35
Fuente: (Electronilab, 2015)

2.10 APLICACIÓN MÓVIL

Una aplicación móvil es un programa diseñado para ser ejecutado en teléfonos inteligentes, tabletas y otros dispositivos móviles que “permite visualizar, recoger y actualizar tu información geográfica en un dispositivo móvil como Tablet PC, sistemas montados en

vehículos, teléfonos inteligentes de Windows, iPhone y iPad y dispositivos Android”(Arcgis, 2013).Esta diseñada para educar entretener o ayudar en la vida diaria a los usuarios, estas funcionalidades antes solo estaban disponibles en PC ahora pueden ser ejecutadas en los dispositivos móviles. Por general se encuentran disponibles a través de plataformas de distribución. Existe una variedad de aplicaciones para móviles que se mencionan a continuación:

- **Aplicaciones de sistema**, estas aplicaciones estarán relacionadas al funcionamiento de nuestro terminal, como podrían ser compresores de archivos, seguridad del terminal, registro de llamadas, gestión de mensajes.
- **Aplicaciones ofimáticas**, las aplicaciones ofimáticas son aquellas que permiten trabajar con documentos de texto, hojas de cálculo excel, ficheros PDF, etc. Las más comunes son QuickOffice y Documents To Go, aunque son de pago.
- **Aplicaciones Web**: Para poder utilizar este tipo de aplicaciones hará falta una conexión de Internet en nuestro móvil. Un ejemplo de este tipo de aplicaciones serían: el GPS, correo electrónico, Google Maps, navegadores Web.
- **Aplicaciones de conexión con las redes sociales**, como: Facebook, Twitter, Hi5.
- **Aplicaciones de accesibilidad**: este tipo de aplicaciones facilitará el uso del terminal para personas con algún tipo de discapacidad. Ejemplos de este tipo de aplicaciones sería: reconocimiento de voz, reconocimiento de caracteres, lectura de texto, etc.
- Existen otros tipos de aplicaciones como las multimedia, los juegos. (Guadalinfo, 2011)

2.10.1 Tipos de aplicaciones móviles

Existen 3 tipos de aplicaciones móviles:

1. Aplicación nativa. Una aplicación nativa es la que se desarrolla de forma específica para un determinado sistema operativo, llamado Software Development Kit o SDK. Cada una de las plataformas, Adroid, iOS o Windows Phone, tienen un sistema diferente, por lo que si quieres que tu app esté disponible en todas las plataformas se deberán de crear varias apps con el lenguaje del sistema operativo seleccionado. (LanceTalent, 2014)

Tabla 2.2: Ventajas en inconvenientes de una aplicación nativa.

Fuente: (LanceTalent, 2014)

Ventajas	Inconvenientes
<ul style="list-style-type: none"> • Acceso completo al dispositivo • Mejor experiencia del usuario • Visibilidad en APP Store • Envío de notificaciones “avisos” a los usuarios • La actualización de la app es constante 	<ul style="list-style-type: none"> • Diferentes habilidades/idiomas/herramientas para cada plataforma de destino • Tienden a ser más caras de desarrollar • El código del cliente no es reutilizable entre las diferentes plataformas.

2.- Aplicación Web.- Una aplicación Web o Webapp es la desarrollada con lenguajes muy conocidos por los programadores, como es el HTML, Javascript y CSS. La principal ventaja con respecto a la nativa es la posibilidad de programar independiente del sistema operativo en el que se usará la aplicación. De esta forma se pueden ejecutar en diferentes dispositivos sin tener que crear varias aplicaciones. Las aplicaciones Web se ejecutan dentro del propio navegador Web del dispositivo a través de una URL. Por ejemplo en Safari, si se trata de la plataforma iOS. El contenido se adapta a la pantalla adquiriendo un aspecto de navegación APP.

La diferencia con una aplicación nativa (además de los inconvenientes que se muestran en la tabla) es que no necesita instalación por lo que no pueden estar visibles en app store y la promoción y comercialización debe realizarse de forma independiente. De todas formas se puede crear un acceso directo que sería como “instalar” la aplicación en el dispositivo. . (LanceTalent, 2014)

Tabla 2.2: Ventajas en inconvenientes de una aplicación Web.

Fuente: (LanceTalent, 2014)

Ventajas	Inconvenientes
<ul style="list-style-type: none"> • El mismo código base reutilizable en múltiples plataformas • Proceso de desarrollo más sencillo y económico • No necesitan ninguna aprobación externa para publicarse (a diferencia de las para estar visibles en app store) • El usuario siempre dispone de la última versión • Pueden reutilizarse sitios “responsive” ya diseñados 	<ul style="list-style-type: none"> • Requiere de conexión a internet • Acceso muy limitado a los elementos y características del hardware del dispositivo • La experiencia del usuario (navegación, interacción) y el tiempo de respuesta es menor que una app nativa • Requiere de mayor esfuerzo en promoción y visibilidad.

3.- Aplicación híbrida.- Una aplicación híbrida es una combinación de las dos anteriores, se podría decir que recoge lo mejor de cada una de ellas. Las apps híbridas se desarrollan con lenguajes propios de las Webabpp, es decir, HTML, Javascript y CSS por lo que permite su uso en diferentes plataformas, pero también dan la posibilidad de acceder a gran parte de las características del hardware del dispositivo. La principal ventaja es que a pesar de estar desarrollada con HTML, Java o CSS, es posible agrupar los códigos y distribuirla en app store.

PhoneGap es uno de los frameworks más utilizados por los programadores para el desarrollo multiplataforma de aplicaciones híbridas. Otro ejemplo de herramienta para desarrollar apps híbridas es Cordova. (LanceTalent, 2014)

Tabla 2.3: Ventajas en inconvenientes de una aplicación híbrida.

Fuente: (LanceTalent, 2014)

Ventajas	Inconvenientes
<p>Es posible distribuirla en las tiendas de iOS y Android</p> <p>Instalación nativa pero construida con JavaScript, HTML y CSS</p> <p>El mismo código base para múltiples plataformas</p> <p>Acceso a parte del hardware del dispositivo</p>	<p>Experiencia del usuario más propia de la aplicación Web que de la app nativa.</p> <p>Diseño visual no siempre relacionado con el sistema operativo en el que se muestre.</p>

2.11 ANDROID

En los últimos 10 años los teléfonos móviles han experimentado una gran evolución, desde los primeros terminales, grandes y pesados, pensados sólo para hablar por teléfono en cualquier parte, a los últimos modelos, con los que el término “medio de comunicación” se queda bastante pequeño. Es así como nace Android.

Android es un sistema operativo y una plataforma software, basado en Linux un núcleo de sistema operativo libre, gratuito y multiplataforma para teléfonos móviles. Además, también usan este sistema operativo tablets, netbooks, reproductores de música e incluso PC's. Android permite programar en un entorno de trabajo (framework) de Java, aplicaciones sobre una máquina virtual. Además, lo que le diferencia de otros sistemas operativos, es que cualquier persona que sepa programar puede crear nuevas aplicaciones, widgets⁵, o incluso,

⁵ Un widget es una pequeña aplicación que facilita el acceso a funciones frecuentes

modificar el propio sistema operativo, dado que Android es de código libre, por lo que sabiendo programar en lenguaje Java, va a ser muy fácil comenzar a programar en esta plataforma.

2.11.1 Arquitectura de Android

En la Figura 2.7 se muestra la arquitectura de Android. Como se puede ver está formada por cuatro capas. Una de las características más importantes es que todas las capas están basadas en software libre.

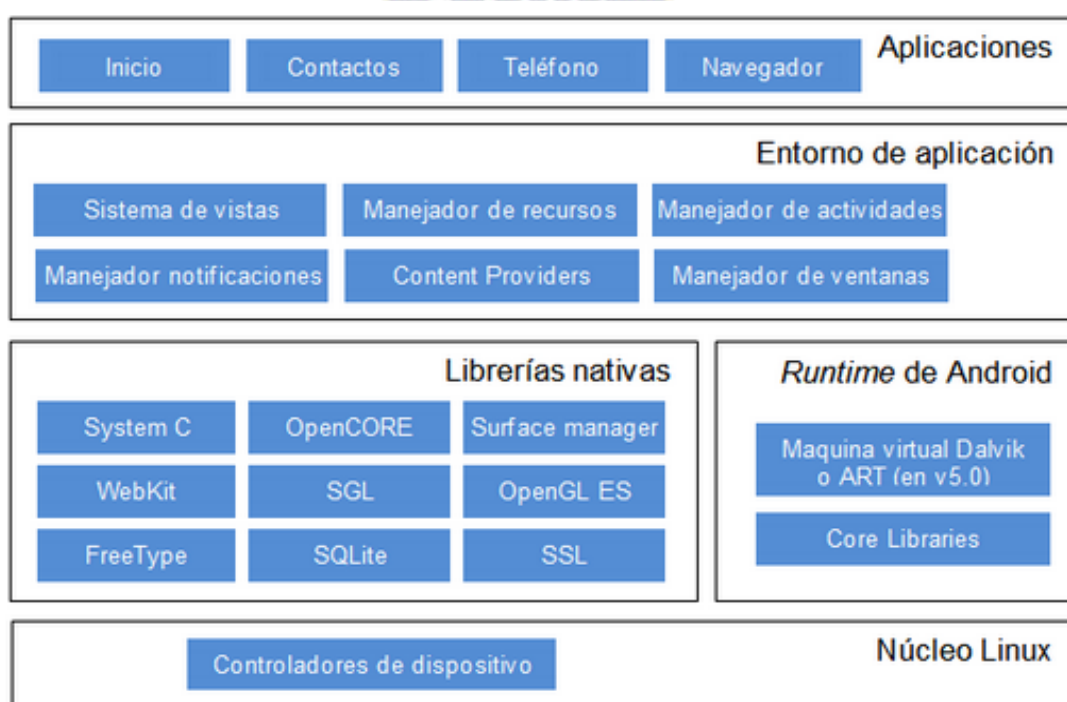


Figura 2.7: Arquitectura de Android
Fuente: (UCM, 2005).

2.11.1.1. El núcleo Linux

El núcleo de Android está formado por el sistema operativo Linux versión 2.6. Esta capa proporciona servicios como la seguridad, el manejo de la memoria, el multiproceso, la pila de protocolos y el soporte de controladores para dispositivos. Esta capa del modelo actúa como capa de abstracción entre el hardware y el resto de la pila. Por lo tanto, es la única que es dependiente del hardware.

2.11.1.2. Runtime de Android

Está basado en el concepto de máquina virtual utilizado en Java. Dado las limitaciones de los dispositivos donde ha de correr Android (poca memoria y procesador limitado) no fue posible utilizar una máquina virtual Java estándar. Google tomó la decisión de crear una nueva, la máquina virtual Dalvik, que respondiera mejor a estas limitaciones.

Algunas características de la máquina virtual Dalvik que facilitan esta optimización de recursos son: que ejecuta ficheros Dalvik ejecutables (.dex) –formato optimizado para ahorrar memoria. Además, está basada en registros. Cada aplicación corre en su propio proceso Linux con su propia instancia de la máquina virtual Dalvik. Delega al kernel de Linux algunas funciones como threading y el manejo de la memoria a bajo nivel.

A partir de Android 5.0 se reemplaza Dalvik por ART. Esta nueva máquina virtual consigue reducir el tiempo de ejecución del código Java hasta en un 33%. También se incluye en el Runtime de Android el “core libraries” con la mayoría de las bibliotecas disponibles en el lenguaje Java.

2.11.1.3. Bibliotecas nativas

Incluye un conjunto de bibliotecas en C/C++ usadas en varios componentes de Android. Están compiladas en código nativo del procesador. Muchas de las bibliotecas utilizan proyectos de código abierto. Algunas de estas bibliotecas son:

- **System C library**, una derivación de la librería BSD de C estándar (libc), adaptada para dispositivos embebidos basados en Linux.
- **Media Framework**, librería basada en PacketVideo's OpenCORE; soporta codecs de reproducción y grabación de multitud de formatos de audio vídeo e imágenes MPEG4, H.264, MP3, AAC, AMR, JPG y PNG.
- **Surface Manager**, maneja el acceso al subsistema de representación gráfica en 2D y 3D.

- **WebKit**, soporta un moderno navegador Web utilizado en el navegador Android y en la vista Webview. Se trata de la misma librería que utiliza Google Chrome y Safari de Apple.
- **SGL**, motor de gráficos 2D.
- **Bibliotecas 3D**, implementación basada en OpenGL ES 1.0 API. Las bibliotecas utilizan el acelerador hardware 3D si está disponible, o el software altamente optimizado de proyección 3D.
- **FreeType**, fuentes en bitmap y renderizado vectorial.
- **SQLite**, potente y ligero motor de bases de datos relacionales disponible para todas las aplicaciones.
- **SSL** proporciona servicios de encriptación Secure Socket Layer (capa de conexión segura).

2.11.1.4. Entorno de aplicación

Proporciona una plataforma de desarrollo libre para aplicaciones con gran riqueza e innovaciones: sensores, localización, servicios, barra de notificaciones, etc.

Esta capa ha sido diseñada para simplificar la reutilización de componentes. Las aplicaciones pueden publicar sus capacidades y otras pueden hacer uso de ellas, sujetas a restricciones de seguridad. Este mismo mecanismo permite a los usuarios reemplazar componentes.

Los servicios más importantes que incluye son:

- **Views:** Extenso conjunto de vistas, (parte visual de los componentes).
- **Resource Manager:** Proporciona acceso a recursos que no son en código.
- **Activity Manager:** Maneja el ciclo de vida de las aplicaciones y proporciona un sistema de navegación entre ellas.
- **Notification Manager:** Permite a las aplicaciones mostrar alertas personalizadas en la barra de estado.
- **Content Providers:** Mecanismo sencillo para acceder a datos de otras aplicaciones (como los contactos).

2.11.1.5. Aplicaciones

Este nivel está formado por el conjunto de aplicaciones instaladas en una máquina Android. Todas las aplicaciones han de correr en la máquina virtual Dalvik para garantizar la seguridad del sistema.

Normalmente las aplicaciones Android están escritas en Java. Para desarrollar aplicaciones en Java se puede utilizar el Android SDK. Existe otra opción consistente en desarrollar las aplicaciones utilizando C/C++. Para esta opción se puede utilizar el Android NDK (Native Development Kit).

2.12 AGENTES

Un agente, de manera general, es todo aquello que puede considerarse que percibe su ambiente mediante sensores y que responde o actúa en tal ambiente mediante efectores (Russell,1996).

La palabra “Agente” se refiere a todo ente que posee la habilidad, capacidad y autorización para actuar en nombre de otro. A diario, los agentes humanos asisten a las personas en tareas que requieren recursos especializados o conocimiento específico en un dominio, ver Figura 2.8.

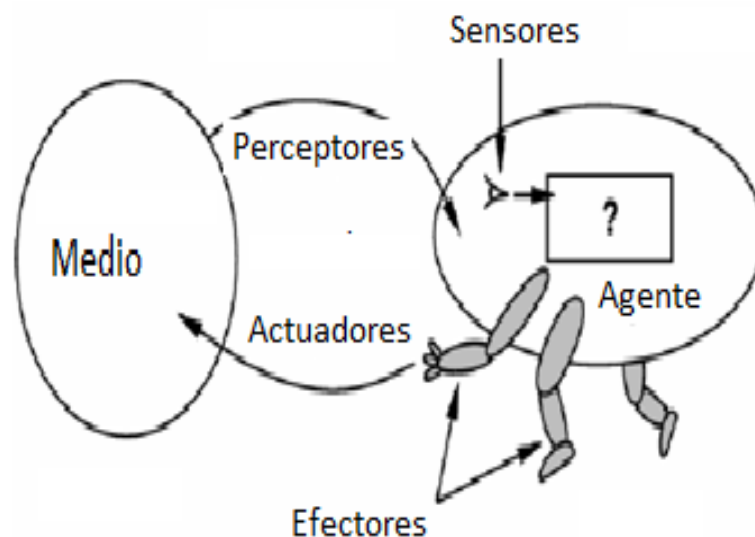


Figura 2.8: Diagrama de Agente en interacción con el ambiente.

Fuente: (Russell y Norvig, 1996).

De esta manera es posible encontrar diversos tipos de agentes entre ellos los agentes humanos, en los que los cinco sentidos sirven de sensores y las extremidades sirven de efectores; los agentes robóticos, en el que los sensores son suplantados por cámaras de video y los efectores son ciertos mecanismos; y los agentes de software, que son un programa de computación que se ejecuta en un ambiente y realiza acciones dentro de éste para alcanzar las metas para las cuales fue diseñado y sus percepciones y acciones están dadas por instrucciones de programas en algún lenguaje en particular. (Franklin 1996).

Los agentes cumplen con los requerimientos para los cuales fueron entrenados. El usuario “delega” en el agente una o varias tareas que debe llevar a cabo quedando a la espera de los resultados. Dichas tareas son a menudo fáciles de especificar, pero - en algunos casos - complejas de realizar, ver Figura 2.9 (Hernan, 1999).

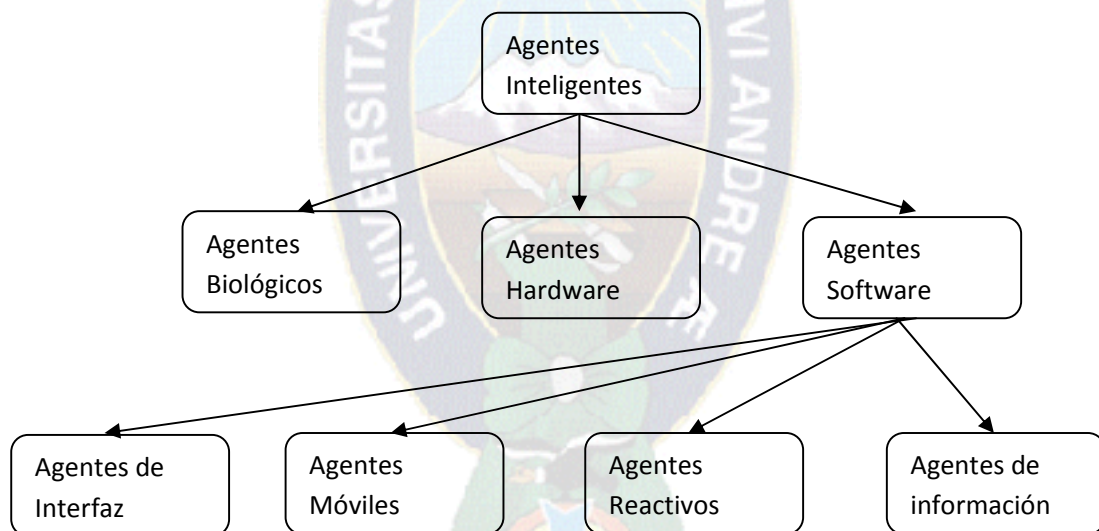


Figura 2.9: Clasificación de Agentes Inteligentes
Fuente: (Belloni y Campo, 2000)

La cuestión de qué es un agente (Foner, 1993), está aún siendo debatida al correr el riesgo de que cualquier programa sea denominado agente (Franklin, 1996). Se pueden distinguir dos nociones extremas de agentes (Wooldrige, 1995):

- Una noción débil de agente consiste en definir un agente como a una entidad que es capaz de intercambiar mensajes utilizando un lenguaje de comunicación de agentes.

Esta definición es la más utilizada dentro de la ingeniería del software basada en agentes, cuyo fin es conseguir la interoperabilidad entre aplicaciones a nivel semántico utilizando la emergente tecnología de agentes.

- Una noción más fuerte o restrictiva de agente es la enunciada por Shoham, en su propuesta de programación orientada a agentes (AOP), donde un agente se define como una entidad cuyo estado es visto como un conjunto de componentes mentales, tales como creencias, capacidades, elecciones y acuerdos. (Shoham, 1993)

2.12.1 Agentes móviles

Los agentes móviles son procesos capaces de “viajar” por una red de computadoras, interactuando con hosts externos, recolectando información en nombre de su dueño y retornando a “casa” luego de completarlas tareas establecidas.

Los agentes forman un nivel de abstracción más para el usuario, detrás del cual se encuentran soluciones a cuestiones técnicas en algunos casos complicadas. Una de estas cuestiones es la distribución, es decir, como manejar recursos computacionales distribuidos. Con la idea de agentes móviles los recursos distribuidos no son completamente ocultos al usuario pero tampoco completamente expuestos.

2.13 PLATAFORMA DE AGENTES JAVA

Jade⁶ (Java Agent DEvelopment Framework) es una plataforma desarrollada íntegramente en Java para la creación de sistemas multi-agente.

Además de proporcionar un API para la creación de agentes y elementos relacionados, pone a nuestra disposición una interfaz gráfica y una serie de herramientas para el control y la depuración de nuestro sistema durante el desarrollo del mismo.

Una de las principales características de la plataforma es que los desarrollos están dentro del estándar FIPA⁷. El intercambio de mensajes entre agentes, así como las performativas empleadas se corresponderán con lo especificado en este estándar. Al ser una plataforma

⁶ Java Agent Development Environment, plataforma de sistemas multi-agente.

⁷ Foundation for Intelligent Physical Agents

desarrollada en Java será posible utilizarla en cualquier sistema operativo que disponga de una máquina virtual de Java, ver Figura 2.10.

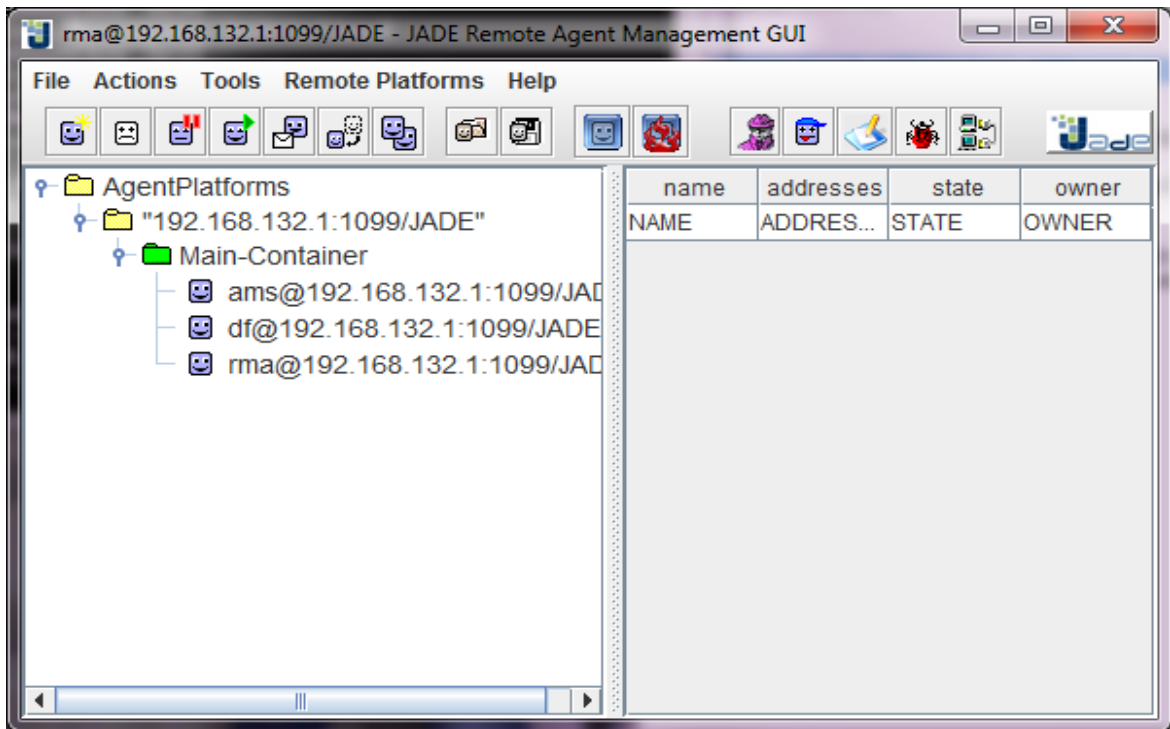


Figura 2.10: Interfaz principal JADE
Fuente: (Tilab, 2014)

2.13.1 Paquete de agentes Java

JADE está compuesto por una serie de paquetes en Java. Los principales son los siguientes:

- ✓ **JADE.core** es el núcleo del sistema. Proporciona la clase `JADE.core.Agent` que es imprescindible para la creación de SMA. También contiene el paquete `JADE.core.behaviour` que incluye las clases de los comportamientos que ha de tener todo agente.
- ✓ **JADE.lang** contiene las clases específicas para soportar un lenguaje de comunicación entre agentes. Está formado por el paquete `Java.lang.acl`.
- ✓ **JADE.content** contiene las clases específicas para soportar un lenguaje de comunicación entre agentes (ACL). Está formado por los paquetes `JADE.content.lang.acl` y `JADE.content.lang.sl`. En éste mismo se encuentra todas las

clases necesarias para la realización e implementación de la ontología (JADE.content.onto).

- ✓ **JADE.domain** contiene todas las clases para representar la plataforma de agentes y los modelos del dominio, tales como entidades para la administración de agentes, lenguajes y ontologías (AMS, DF, ACC...).
- ✓ **JADE.proto** paquete que contiene los protocolos de interacción entre agentes FIPA.
- ✓ **JADE.tools** contiene algunas herramientas que facilitan el desarrollo de aplicaciones y la administración de la plataforma:
 - **Remote Management Agent (RMA)**: Es el agente que se encarga de la interfaz gráfica de JADE para la administración y control del sistema.
 - **Dummy Agent**: Es una herramienta de monitorización y depuración, compuesta por una interfaz gráfica y un agente JADE. Permite la creación de mensajes ACL y enviarlos a otros agentes pudiendo visualizarlos.
 - **Sniffer**: Agente que intercepta mensajes ACL y los visualiza, resulta muy útil para el seguimiento de una conversación.
 - **Introspector Agent**: Agente que permite la monitorización del ciclo de vida de un agente.
 - **SocketProxyAgent**: Agente que actúa como gateway bidireccional entre la plataforma JADE y una conexión TCP/IP.
- ✓ **JADE.gui** contiene herramientas para construir interfaces gráficas.

2.13.2 Componentes principales de la plataforma de agentes

Para Jade, los agentes residen en un entorno predefinido llamado plataforma. Cada plataforma está dividida en contenedores y cada uno de ellos contendrá agentes. Los contenedores podrían asemejarse al dominio de los agentes.

La plataforma se enciende en una máquina determinada, mientras que los agentes podrían ejecutarse en cualquier estación; esto ofrece un comportamiento distribuido del SMA que será la base para la incorporación del LEAP.

Tal como se define en la FIPA, en cada plataforma se inician dos agentes que tienen unas funciones vitales para el funcionamiento del sistema:

a. Directory facilitator (DF): Podría definirse como un servicio de “páginas amarillas”, es decir, contendrá todos los servicios que proporcionan los agentes que se han dado de alta en la plataforma. Será de especial utilidad a la hora de requerir algún servicio, puesto que informará de quien puede proporcionárnoslo.

b. Agent management system (AMS): Se podría considerar como un servicio de “páginas blancas”, donde se registran las direcciones de los agentes que se han dado de alta. Dado que JADE se fundamenta en entornos distribuidos, será básico para conocer la localización de los agentes.

Gracias a estos agentes la comunicación en la plataforma es muy sencilla de realizar y permite una gran flexibilidad y transparencia. Por ejemplo, si un agente A quiere enviar un mensaje a un agente B que desarrolla la tarea T, el agente A preguntará al DF qué agentes me pueden proporcionar dicho servicio T. Cuando se quiera conocer la dirección física de un agente determinado, preguntará al AMS. Para que este sistema funcione correctamente será necesario que, durante el proceso de inicialización del agente, éste informe al DF de qué servicios dispone y el tipo de agente que es (para poder identificarlo); al hacerlo, AMS le dará una dirección física (dependiendo de dónde se ejecute), para que otros agentes se puedan poner en contacto con él.

2.13.3 Lenguaje de comunicación

El lenguaje FIPA-SL (Semantic Language) es un lenguaje formal con el cual se pretende poder resolver una gran variedad de problemas. En FIPA-SL se pueden formar expresiones lógicas, intercambiar información entre agentes y expresar acciones a realizar. La notación utilizada es similar al Lisp: Se definen un conjunto de predicados genéricos y cada uno de ellos tiene una serie de atributos y parámetros.

2.13.4 Plataforma de agentes para Android

LEAP⁸ es un addon para JADE, que permite obtener un entorno de ejecución para agentes que cumplan las especificaciones FIPA⁹ y se ejecuten en dispositivos de recursos limitados, que soporten Java.

Hasta la versión 2.1 LEAP fue desarrollado dentro del LEAP IST project. El hecho de desarrollarse de forma externa al proyecto JADE provocaba ciertos problemas e inconvenientes, tanto a la hora de instalar el entorno JADE-LEAP, como a la hora de ejecutar los agentes en él (básicamente se encontraba ciertas limitaciones que se convertían en dificultades añadidas). Las cosas cambiaron radicalmente desde la llegada de JADE 3.0 que vino acompañada de su propio addon LEAP 3.0.

La necesidad de LEAP viene dada por la incapacidad de JADE de ejecutarse correctamente en condiciones extremas de recursos, situación que se encuentra claramente al trabajar en móviles o PDAs¹⁰. JADE requiere un mínimo de JDK1.2, problema añadido puesto que la mayoría de estos dispositivos solo soportan el llamado PersonalJava¹¹ o MIDP¹². Por último también hay que destacar la naturaleza de las conexiones de estos dispositivos: Bajo ancho de banda, conexión intermitente, IPs dinámicas. Todo estos aspectos deben ser tratados de forma especial, es por ello que LEAP nació. Su base principal es crear un contenedor para cada dispositivo, donde sus agentes se ejecutarán. Estos contenedores, supeditados al contenedor principal (main container), formarán parte de una misma plataforma, que permitirá a todos los agentes trabajar conjuntamente.

Dicho esto, recordar que el addon LEAP puede ser descargado gratuitamente en la página oficial de JADE (<http://JADE.cselt.it>), en la cual también se encuentra un tutorial muy acertado sobre LEAP, que ayudará a profundizar al lector que se muestre interesado por esta plataforma.

⁸Lightweight Extensible Agent Platform (Plataforma Ligera Agente Extensible).

⁹Foundation for Intelligent Physical Agents (Fundación para Agentes Físicos Inteligentes).

¹⁰Personal Digital Assistant (Asistente Digital Personal).

¹¹Maquina virtual para PDAs. A día de hoy se utiliza su versión 1.2 (JeodeVM)

¹²Entorno facilitado por el J2ME de sun. Es el utilizado por los móviles. Las aplicaciones que ejecuta se llaman midlets

2.13.5 Entorno de ejecución

LEAP, al ser combinado con JADE, substituye ciertas partes del kernel de JADE, creando un entorno modificado que se identifica como JADE-LEAP, que puede funcionar correctamente en dispositivos muy limitados desde el punto de vista de recursos. LEAP ofrece 3 versiones diferentes pensadas para cada uno de los 3 tipos posibles de entornos Java disponibles:

- **j2se (Simple Edition)**, la indicada para ejecutarse en estaciones con jdk1.2 o superior
- **pjava (PersonalJava)**, idónea para dispositivos con PersonalJava como las PDAs.
- **midp (Micro Edition)**, para dispositivos que soportan MIDP1.0 como teléfonos móviles.

Aunque internamente muy diferentes, las 3 versiones ofrecen la misma API a los programadores, sólo presentando ciertas diferencias las versiones j2se y pjava respecto a la midp.

2.14 Matlab

Matlab es un multi-paradigma de la computación numérica medio ambiente y lenguaje de programación de cuarta generación. Desarrollado por The MathWorks, Matlab permite manipulación de matrices, trazado de funciones y datos, implementación de algoritmos , la creación de interfaces de usuario y la interfaz con los programas escritos en otros idiomas, incluyendo C , C ++ , Java , Fortran y Python .

Aunque Matlab está pensado principalmente para computación numérica, es una caja de herramientas opcional utiliza el MuPAD motor simbólico, permitiendo el acceso a la computación simbólica. Un paquete adicional, Simulink, añade gráfica de simulación multidominio y diseño basado en modelos dinámicos y sistemas embebidos.

En 2004, Matlab tenía alrededor de un millón de usuarios en toda la industria y el mundo académico, los usuarios de Matlab provienen de diversos orígenes de la ingeniería, la ciencia y la economía. Matlab es ampliamente utilizado en las instituciones académicas y de investigación, así como las empresas industriales.

2.14.1 Entorno de Matlab

La ventana de Matlab cuenta con diversas secciones. La más importante es la command window. Aquí es donde se puede usar Matlab de forma interactiva. Las otras secciones son: el current directory, que es la carpeta en la cual se está trabajando, el workspace, donde se visualiza las variables que han sido creadas, y el command history, que guarda todas las acciones realizadas en el command window. En la esquina superior izquierda, se encuentra la opción de crear un nuevo archivo M. En estos archivos (que tienen la extensión .m) se escriben los programas donde se usa el lenguaje Matlab. Otras opciones que tiene son la de iniciar Simulink y GUIDE, una IDE que facilita la creación de interfaces gráficas, ver Figura 2.11.

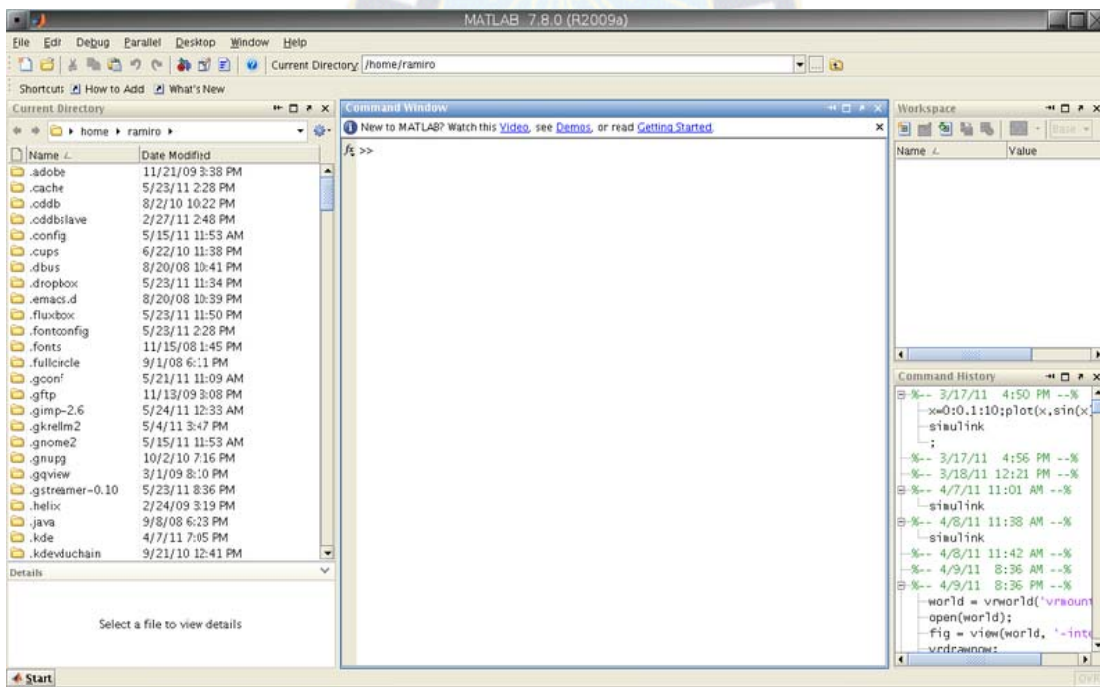


Figura 2.11: Entorno de Matlab
Fuente: (ISA, 2013)

2.15 Simulink

Simulink es una extensión gráfica de Matlab para el modelado y simulación de sistemas. Una de las principales ventajas de Simulink es la capacidad de modelar un sistema no lineal, que una función de transferencia es incapaz de hacerlo. Otra ventaja de Simulink es la capacidad

de asumir las condiciones iniciales. Cuando se construye una función de transferencia, las condiciones iniciales se supone que son cero.

En Simulink, los sistemas se dibujan en la pantalla como diagramas de bloques. Muchos elementos de diagramas de bloques están disponibles, tales como funciones de transferencia, uniones sumadores, etc., así como entrada virtual y los dispositivos de salida, tales como generadores de funciones y osciloscopios. Simulink se integra con Matlab y los datos pueden ser transferidos fácilmente entre los programas. Simulink está soportada en UNIX, Macintosh y entornos de Windows, y se incluye en la versión para estudiantes de Matlab para ordenadores personales.

El conjunto de componentes incluido junto al programa Simulink, incluye bibliotecas de fuentes de señal, dispositivos de presentación de datos, sistemas lineales y no lineales, conectores y funciones matemáticas. En caso de que sea necesario, se pueden crear nuevos bloques, la cantidad que requiere el usuario. El programa Simulink se inicia desde el botón “Simulink Library Browser” de la ventana de comandos de Matlab.

Una vez iniciado el programa el entorno de trabajo queda dividido en tres partes.

- Ventana de comandos de Matlab: Desde la que se puede ejecutar cualquier comando del mismo, dar valores a variables y controlar la ejecución de las simulaciones.
- Ventana de la biblioteca de Simulink: Desde la que se selecciona los componentes que se van a insertar en el sistema a simular.
- Ventanas de los modelos: En las que se dibujan los modelos, se realizan y controlan las simulaciones.

2.15.1 Elementos de Simulink

Hay dos clases principales de artículos en Simulink: bloques y líneas. Los bloques se utilizan para generar, modificar, combinar, de salida, y mostrar señales. Las líneas se utilizan para transferir señales de un bloque a otro.

a. Bloques: Hay varias clases generales de bloques dentro de la biblioteca Simulink

- Fuentes, utilizados para generar diferentes señales
- Sumideros, se utiliza para señales de salida o de visualización

- Continua, los elementos del sistema de tiempo continuo (funciones de transferencia, modelos de espacio de estado, controladores PID, etc.)
- Discreta, lineal, los elementos del sistema de tiempo discreto (funciones de transferencia discreta, los modelos de espacio de estado discretos, etc.)
- Operaciones matemáticas, contiene muchas operaciones comunes de matemáticas (ganancia, suma, producto, valor absoluto, etc.)
- Puertos y subsistemas, contiene bloques útiles para construir un sistema de Bloques tienen cero a varios terminales de entrada y de cero a varios terminales de salida.
- Terminales de entrada no utilizados se indican mediante un pequeño triángulo abierto.
- Terminales de salida no utilizados se indican mediante un pequeño punto triangular.

b. Líneas

Las líneas transmiten señales en la dirección indicada por la flecha. Las líneas siempre deben transmitir señales desde el terminal de salida de un bloque hasta la terminal de entrada de otro bloque, una señal puede ser una señal escalar o una señal vector, existen excepciones donde una línea puede tocar fuera de otra línea.

- Para una sola entrada, Single-salida (SISO) sistemas, generalmente se utilizan señales escalares.
- Para múltiples entradas, múltiples salidas sistemas (MIMO), las señales vectoriales son de uso frecuente, las cuales consiste en dos o más señales escalares.

Las líneas utilizadas para transmitir señales escalares y vectoriales son idénticos. El tipo de señal transportada por una línea se determina por los bloques en cada extremo de la línea.

2.16 BASE DE DATOS

PostgreSQL es un Sistema de gestión de bases de datos relacional orientado a objetos y libre, publicado bajo la licencia BSD.

Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL no es manejado por una empresa y/o persona, sino que es dirigido por una comunidad de desarrolladores que

trabajan de forma desinteresada, altruista, libre y/o apoyada por organizaciones comerciales. Dicha comunidad es denominada el PGDG (PostgreSQL Global Development Group).

El uso de caracteres en mayúscula en el nombre PostgreSQL puede confundir a algunas personas a primera vista. Las distintas pronunciaciones de "SQL" pueden llevar a confusión. Es también común oír abreviadamente como simplemente "Postgres", el que fue su nombre original. Debido a su soporte del estándar SQL entre la mayor parte de bases de datos relacionales, la comunidad consideró cambiar el nombre al anterior Postgres. Sin embargo, el PostgreSQL Core Team anunció en 2007 que el producto seguiría llamándose PostgreSQL. El nombre hace referencia a los orígenes del proyecto como la base de datos "post-Ingres", y los autores originales también desarrollaron la base de datos Ingres.

2.16.1 Características de PostgreSQL

Algunas de sus principales características son, entre otras:

- Alta concurrencia

Mediante un sistema denominado MVCC (Acceso concurrente multiversión, por sus siglas en inglés) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de lo último a lo que se le hizo commit. Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases, eliminando la necesidad del uso de bloqueos explícitos.

- Amplia variedad de tipos nativos

PostgreSQL provee nativamente soporte para:

- ✓ Números de precisión arbitraria.
- ✓ Texto de largo ilimitado.
- ✓ Figuras geométricas (con una variedad de funciones asociadas).
- ✓ Direcciones IP (IPv4 e IPv6).
- ✓ Bloques de direcciones estilo CIDR.
- ✓ Direcciones MAC.
- ✓ Arrays.

Adicionalmente los usuarios pueden crear sus propios tipos de datos, los que pueden ser por completo indexables gracias a la infraestructura GiST de PostgreSQL. Algunos ejemplos son los tipos de datos GIS creados por el proyecto PostGIS.

2.17 INGENIERÍA DEL SOFTWARE

Es el establecimiento y uso de principios robustos de la ingeniería a fin de obtener económicamente software que sea fiable y que funcione eficientemente sobre máquinas reales. El IEEE¹³ ha desarrollado una definición más completa: La aplicación de un enfoque sistemático, disciplinado y cuantificable hacia el desarrollo, operación y mantenimiento del software; es decir, la aplicación de ingeniería al software. El estudio de enfoques como en:

- Proceso, métodos y herramientas
- Una visión general de la ingeniería del software

2.17.1 Proceso, métodos y herramientas

La Ingeniería del software es una tecnología multicapa. Como muestra la Figura 2.12, cualquier enfoque de ingeniería debe apoyarse sobre un compromiso de organización de calidad.

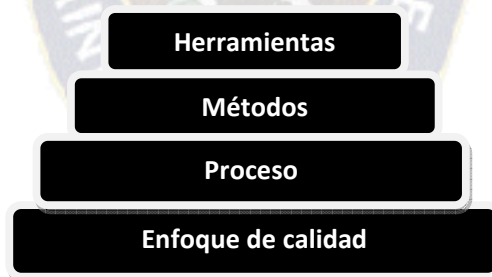


Figura 2.12: Capas de la ingeniería del software
Fuente: (Pressman, 2005)

El fundamento de la ingeniería del software es la capa de proceso. El proceso de la ingeniería del software es la unión que mantiene juntas las capas de tecnología y que permite un desarrollo racional y oportuno de la ingeniería del software. Las áreas claves del proceso forman la base del control de gestión de proyectos del software y establecen el contexto en el que se aplican los métodos técnicos, se obtienen productos del trabajo se establecen hitos, se asegura la calidad y el cambio se gestiona adecuadamente.

¹³ Institute for electrical and electronics Engineers es un Instituto internacional localizado en Norte América.

Los métodos de la ingeniería del software indican cómo construir técnicamente el software. Los métodos abarcan una gran gama de tareas que incluyen análisis de requisitos, diseño, construcción de programas, pruebas y mantenimiento. Los métodos de la ingeniería del software dependen de un conjunto de principios básicos que gobiernan cada área de la tecnología e incluyen actividades de modelado y otras técnicas descriptivas.

Las herramientas de la Ingeniería del software proporcionan un enfoque automático o semi-automático para el proceso y para los métodos. Cuando se integran herramientas para que la información creada por una herramienta la pueda utilizar otra, se establece un sistema de soporte para el desarrollo del software llamado ingeniería del software asistida por computadora (CASE).

2.17.2 Una visión general de la ingeniería del software

La ingeniería es el análisis, diseño, construcción, verificación y gestión de entidades técnicas.

La fase de definición se centra sobre el qué. Es decir, durante la definición, el que desarrolla el software intenta identificar qué información ha de ser procesada, qué función y rendimiento se desea, qué comportamiento del sistema, qué interfaces van a ser establecidas, qué restricciones de diseño existen, y qué criterios de validación se necesitan para definir un sistema correcto. Por tanto, han de identificarse los requisitos clave del sistema y del software.

La fase de desarrollo se centra en el cómo. Es decir, durante el desarrollo un ingeniero del software intenta definir cómo han de diseñarse las estructuras de datos, cómo ha de implementarse la función dentro de una arquitectura de software, cómo han de implementarse los detalles procedimentales, cómo han de caracterizarse interfaces, cómo ha de traducirse el diseño en un lenguaje de programación (o lenguaje no procedimental) y cómo ha de realizarse la prueba.

La fase de mantenimiento se centra en el cambio que va asociado a la corrección de errores, a las adaptaciones requeridas a medida que evoluciona el entorno del software y a cambios

debidos a las mejoras producidas por los requisitos cambiantes del cliente. Durante la fase de mantenimiento se encuentran cuatro tipos de cambios:

- **Corrección.** Incluso llevando a cabo las mejores actividades de garantía de calidad, es muy probable que el cliente descubra los defectos en el software. El mantenimiento correctivo cambia el software para corregir los defectos.
- **Adaptación.** Con el paso del tiempo, es probable que cambie el entorno original para el que se desarrolló el software. El mantenimiento adaptativo produce modificación en el software para acomodarlo a los cambios de su entorno externo.
- **Mejora.** Conforme se utilice el software, el cliente/ usuario puede descubrir funciones adicionales que van a producir beneficios. El mantenimiento perfectivo lleva al software más allá de sus requisitos funcionales originales.
- **Prevención.** El software de computadora se deteriora debido al cambio, y por esto el mantenimiento preventivo también llamado reingeniería del software, se debe conducir a permitir que el software sirva para las necesidades de los usuarios finales. En esencia, el mantenimiento preventivo hace cambios en programas de computadora a fin de que se puedan corregir, adaptar y mejorar más fácilmente.

2.17.3 Casos de uso

Una vez recopilados los requisitos, bien por reuniones informales, el ingeniero del software puede crear un conjunto de escenarios que identifiquen una línea de utilización para el sistema que va a ser construido. Los escenarios, algunas veces llamados casos de uso, facilitan una descripción de cómo el sistema se usará.

Para crear un caso de uso, el analista debe primero identificar los diferentes tipos de personas (o dispositivos) que utiliza el sistema o producto. Estos actores actualmente representan papeles que la gente (o dispositivos) juegan como impulsores del sistema. Definido más formalmente, un actor es algo que comunica con el sistema o producto y que es externo al sistema en sí mismo.

Es importante indicar que un actor y un usuario no son la misma cosa. Un usuario normal puede jugar un número de papeles diferentes cuando utiliza un sistema, por lo tanto un actor representa una clase de entidades externas (a veces, pero no siempre personas) que lleva a cabo un papel.

2.17.4 El modelo de clases

Un modelo de clases es una descripción de las clases en un sistema y sus relaciones. No describe el comportamiento dinámico del sistema, por ejemplo el comportamiento de objetos individuales. El primer elemento de un diagrama de clases es una descripción de clases individuales. La Figura 2.13 muestra como se describe una clase.

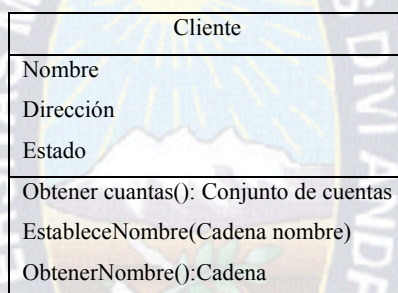


Figura 2.13: Clase descrita en UML

Fuente: (Pressman, 2005)

2.17.4.1 Generalización

Esta relación es la que se mantiene entre una clase X y otra clase Y, cuando la clase Y es el ejemplo más específico de la clase X. Por ejemplo, una relación de generalización existe entre una clase Cuenta, la cual representa una cuenta bancaria general, y una cuenta corriente, que es un ejemplo específico de una cuenta.

Las otras relaciones importantes son la agregación y la composición. Existen dos relaciones que establecen que una clase genera objetos, que son parte de un objeto definido por otra clase. Por ejemplo, un sistema para un fabricante tendrá la necesidad de mantener los datos acerca de los elementos que se están fabricando, y de aquellos que se están haciendo.

2.17.5 Diagrama de secuencia

Durante la ejecución de un sistema orientado a objetos, los objetos interactuarán con cada uno de los otros. Por ejemplo, en un sistema bancario, un objeto Cuenta puede enviar un mensaje a un objeto transacción para crear una transacción que ha ocurrido en una cuenta, por ejemplo una cuenta de cargo. Este tipo de información es importante para el diseñador de un sistema orientado a objetos, durante el proceso de la identificación y validación de clases. Por esta razón, UML tiene dos notaciones equivalentes para definir las interacciones, ver Figura 2.14

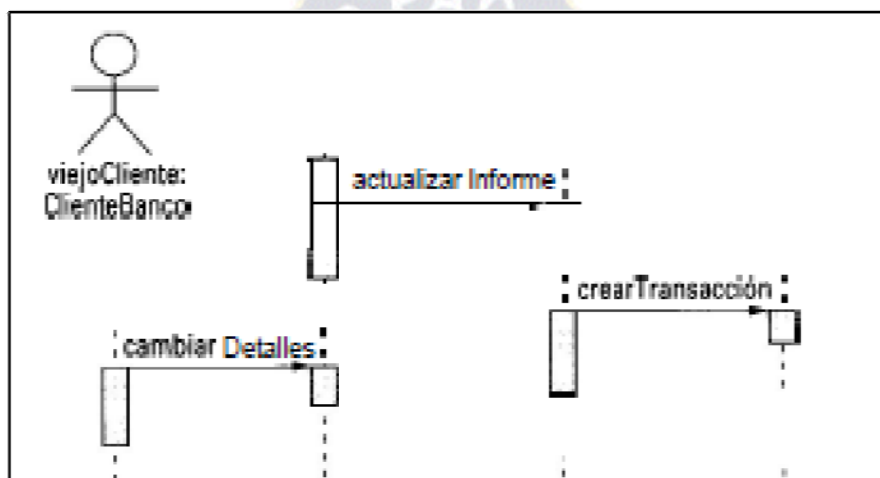


Figura 2.14: Diagrama de secuencia
Fuente: (Pressman, 2005)

2.18 METODOLOGÍA SCRUM

Scrum al ser una metodología de desarrollo ágil tiene como base la idea de creación de ciclos breves para el desarrollo, que comúnmente se llaman iteraciones y que en Scrum se llamarán “Sprints”.

Para entender el ciclo de desarrollo de Scrum es necesario conocer las 5 fases que definen el ciclo de desarrollo ágil.

1. **Conceptos:** Se define de forma general las características del producto y se asigna el equipo que se encargara de su desarrollo.
2. **Especulación:** En esta fase se hacen disposiciones con la información obtenida y se establecen los límites que marcaran el desarrollo del producto, tales como costes y agendas.

Se construirá el producto a partir de las ideas principales y se comprueban las partes realizadas y su impacto en el entorno.

Esta fase se repite en cada iteración y consiste, en rasgos generales, en:

- Desarrollar y revisar los requisitos generales.
- Mantener la lista de las funcionalidades que se esperan.
- Plan de entrega. Se establecen las fechas de las versiones, hitos e iteraciones. Medirá el esfuerzo realizado en el proyecto.

- 3. Exploración:** Se incrementa el producto en el que se añaden las funcionalidades de la fase de especulación.
- 4. Revisión:** El equipo revisa todo lo que se ha construido y se contrasta con el objetivo deseado.
- 5. Cierre:** Se entregara en la fecha acordada una versión del producto deseado. Al tratarse de una versión, el cierre no indica que se ha finalizado el proyecto, sino que seguirá habiendo cambios, denominados “mantenimiento”, que hará que el producto final se acerque al producto final deseado.

Scrum gestiona estas iteraciones a través de reuniones diarias, uno de los elementos fundamentales de esta metodología.

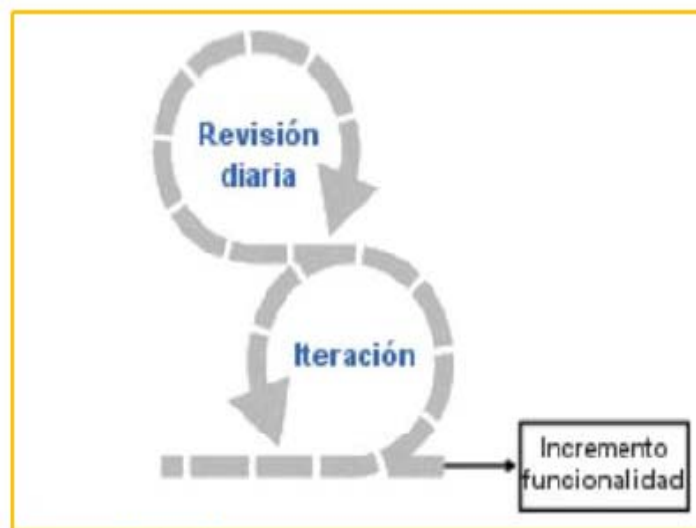


Figura 2.15: Ciclo principal de Scrum
Fuente: (Benito, 2014)

2.18.1 Componentes de Scrum

Para entender todo el proceso de desarrollo del Scrum, se describirá de forma general las fases y los roles. Estas fases y roles se detallaron de forma más concisa más adelante.

Scrum se puede dividir de forma general en 3 fases, que podemos entender como reuniones. Las reuniones forman parte de los artefactos de esta metodología junto con los roles y los elementos que lo forman.

2.18.1.1 Las reuniones

1. Planificación del Backlog

Se definirá un documento en el que se reflejarán los requisitos del sistema por prioridades.

En esta fase se definirá también la planificación del Sprint 0, en la que se decidirá cuales van a ser los objetivos y el trabajo que hay que realizar para esa iteración.

Se obtendrá además en esta reunión un Sprint Backlog, que es la lista de tareas y que es el objetivo más importante del Sprint.

2. Seguimiento del Sprint

En esta fase se hacen reuniones diarias en las que las 3 preguntas principales para evaluar el avance de las tareas serán:

¿Qué trabajo se realizó desde la reunión anterior?

¿Qué trabajo se hará hasta una nueva reunión?

Inconvenientes que han surgido y qué hay que solucionar para poder continuar.

3. Revisión del Sprint

Cuando se finaliza el Sprint se realizará una revisión del incremento que se ha generado. Se presentarán los resultados finales y una versión, esto ayudará a mejorar el feedback con el cliente.

2.18.1.2 Los roles

Los roles se dividen en 2 grupos: cerdos y gallinas, esto surge en el chiste sobre un cerdo y una gallina y su intención de poner un restaurante.

1. Los cerdos: Son las personas que están comprometidas con el proyecto y el proceso de Scrum.

- **Product owner:** Es la persona que toma las decisiones, y es la que realmente conoce le negocio del cliente y su visión del producto. Se encarga de escribir las ideas del cliente, las ordena por prioridad y las coloca en el Product Backlog.
- **ScrumMaster:** Es el encargado de comprobar que el modelo y la metodología funciona. Elimina todos los inconvenientes que hagan que el proceso no fluya e interactuará con el cliente y con los gestores.
- **Equipo de desarrollo:** Suele ser un equipo pequeño de una 5-9 personas y tienen autoridad para organizar y tomar decisiones para conseguir su objetivo. Está involucrado en la estimación del esfuerzo de las tareas del Backlog.

2. Las gallinas: Aunque no son parte del proceso de Scrum, es necesario que parte de la retroalimentación de la salida del proceso y así poder revisar y planear cada Sprint.

- **Usuarios:** Es el destinatario final del producto.
- **Stakeholders:** Las personas a las que el proyecto les producirá un beneficio. Participan durante las versiones del Sprint.
- **Managers:** Toma las decisiones finales participando en la selección de los objetivos y de los requisitos.

2.18.2 Elementos de Scrum

Los elementos que forman a Scrum son:

- **Product Backlog:** Lista de necesidades del cliente.
- **Sprint Backlog:** Lista de tareas que se realizan en un Sprint.
- **Incremento:** Parte añadida o desarrollada en un Sprint, es un parte determinada y totalmente operativa.

2.18.2.1 Product Backlog

Es el inventario en el que se almacenan todas las funciones o requisitos de forma de lista priorizada. Estos requisitos serán los que tendrán el producto o los que ira adquiriendo en sucesivas iteraciones.

Las tres características principales de esta lista de objetivos serán:

- Contendrá los objetivos del producto, se suele usar para expresarlas las historias de usuario.
- En cada objetivo, se indicará el valor que le da el cliente y el coste estimado; de esta manera, se realiza la lista, priorizando por valor coste, se basará en el ROI
- En cada lista se tendrá que indicar las posibles iteraciones y los releases que se han indicado al cliente.
- La lista ha de incluir los posibles riesgos e incluir las tareas necesarias para solventarlos.

2.18.2.2 Sprint Backlog

Es la lista de tareas que elabora el equipo durante la planificación de un Sprint. Se asignan las tareas a cada persona y el tiempo que queda para terminarlas.

De esta manera el proyecto se descompone en unidades más pequeñas y se puede determinar o ver en que tareas no se está avanzando e intentar eliminar el problema.

2.18.3 Incremento

Representa los requisitos que se han completado en una iteración y que son perfectamente operativos. Según los resultados que se obtengan, el cliente puede ir haciendo los cambios necesarios y replanteando el proyecto.



CAPITULO III

MARCO APLICATIVO



3. MARCO APLICATIVO

Resumen

En la primera sección de este capítulo se describe los requerimientos para el desarrollo del prototipo, de la misma manera el análisis de requerimientos, en la segunda sección se detalla el diseño del prototipo, en la tercera sección se muestra el desarrollo del prototipo y en la cuarta sección se detalla la implementación del prototipo su ejecución y análisis de resultados y por último se describe la contrastación de la hipótesis.

En este marco se utiliza el método de ingeniería del software (Pressman, 2005), en el que se define la misma como una tecnología compuesta por estratos: Calidad, proceso, método y herramientas.

En la aplicación del método se utiliza el ciclo de vida incremental evolutivo que de manera general está compuesto por los siguientes elementos:

- Requerimientos
- Análisis de requerimientos
- Diseño
- Desarrollo
- Pruebas
- Mantenimiento

3.1 DEFINICIÓN DE REQUERIMIENTOS

La especificación para el sistema integrado de control de seguridad en viviendas mediante agentes móviles consiste en:

- a) Diseñar un circuito electrónico que tenga los elementos necesarios para realizar una adquisición de datos de la vivienda.
- b) Diseñar un Modelo de bloques que realice un proceso de reducción de incertidumbre mediante lógica difusa.
- c) Diseñar un Modelo de datos para registrar la información abstraído y procesado de la vivienda.
- d) Diseñar agentes móviles que realicen envío y recepción de mensajes.

3.2 ANÁLISIS DE REQUERIMIENTOS

El objetivo de la fase de análisis es el reconocimiento de los elementos básicos del problema y como los percibe el usuario. Donde el análisis de los requerimientos del software permite especificar las características operacionales del software (función, datos y rendimiento) entendiendo el comportamiento del software en el contexto de acontecimiento que afectan al sistema, permitiendo describir el problema de manera que se pueda sintetizar un enfoque o solución global (Pressman, 2005).

3.2.1 Casos de Uso

Los casos de uso son una forma efectiva para capturar a los requerimientos funcionales potenciales de un sistema nuevo. Cada caso de uso es un documento narrativo que describe la secuencia de eventos de un actor que utiliza un sistema para completar un proceso. La vista de casos de uso modela la funcionalidad del sistema, enumerando a los actores, casos de uso y sus interacciones, a través de un diagrama (Rumbaugh et. al, 2000)(Larman, 1999).

En la Figura 3.1 se muestra el diagrama de casos de uso identificado para el prototipo, sistema integrado de control de seguridad en viviendas mediante agentes móviles, donde el actor es el Usuario.

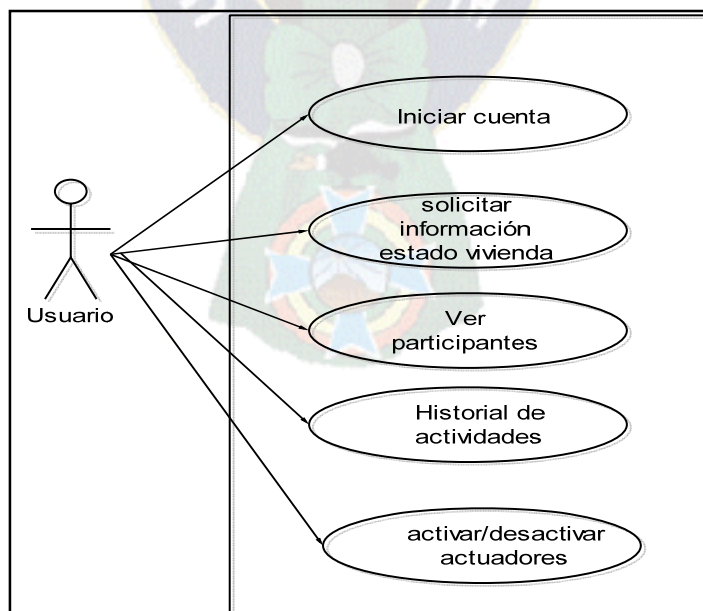


Figura 3.1: Diagrama de casos de uso del sistema integrado
Fuente: Elaborado con base en (Pressman, 2005)

Tabla 3.1: Caso de uso: Iniciar Sesión.

Caso de uso	Iniciar Cuenta
Descripción	El usuario inicia sesión introduciendo su Nick y contraseña para ingresar al sistema.
Precondición	Debe tener acceso a internet, y estar registrado en el sistema.
Actores	Usuario,
Condición de fracaso	Inscripción repetida de agentes, no se encuentra inscrito los agentes.
Condición de éxito	Inscripción realizada de manera correcta.

Tabla 3.2 Caso de uso: Solicitar información estado vivienda.

Caso de uso	Solicitar información estado vivienda
Descripción	El usuario realiza consultas sobre el estado de la vivienda, por estado se refiere a las entradas de cada sensor (temperatura, luz, movimiento).
Precondición	Debe tener acceso a internet, haber iniciado al usuario en la plataforma JADE.
Actores	Usuario
Condición de fracaso	Error de comunicación, en la recepción de la solicitud.
Condición de éxito	Datos presentados, acciones realizadas con éxito.

Tabla 3.3: Caso de uso: Activar/desactivar actuadores.

Caso de uso	Activar/desactivar actuadores
Descripción	El usuario puede activar o desactivar actuadores (luz)
Precondición	Debe tener acceso a internet, haber iniciado al usuario en la plataforma JADE.
Actores	Usuario
Condición de fracaso	Error de comunicación, en la recepción de la solicitud.
Condición de éxito	Datos presentados, acciones realizadas con éxito.

Tabla 3.4: Caso de uso: Historial de actividades.

Caso de uso	Historial de actividades
Descripción	El usuario puede solicitar información acerca de los sucesos realizados en la vivienda.
Precondición	Debe tener acceso a internet, haber iniciado al usuario en la plataforma JADE.
Actores	Usuario
Condición de fracaso	Error de comunicación, en la recepción de la solicitud.
Condición de éxito	Datos presentados, acciones realizadas con éxito.

Tabla 3.5: Caso de uso: Ver Usuarios

Caso de uso	Ver Usuarios
Descripción	El usuario ver quienes participan en la actividad de la gestión de seguridad de la vivienda.
Precondición	Debe tener acceso a internet, haber iniciado al usuario en la plataforma JADE.
Actores	Usuario
Condición de fracaso	Error de comunicación, en la recepción de la solicitud.
Condición de éxito	Ver usuarios participantes, acciones realizadas con éxito.

3.2.2 Diagrama de secuencia

En la Figura 3.2 se muestra el diagrama de secuencia, elaborado para observar el movimiento secuencial y los involucrados cuando se realiza una petición.

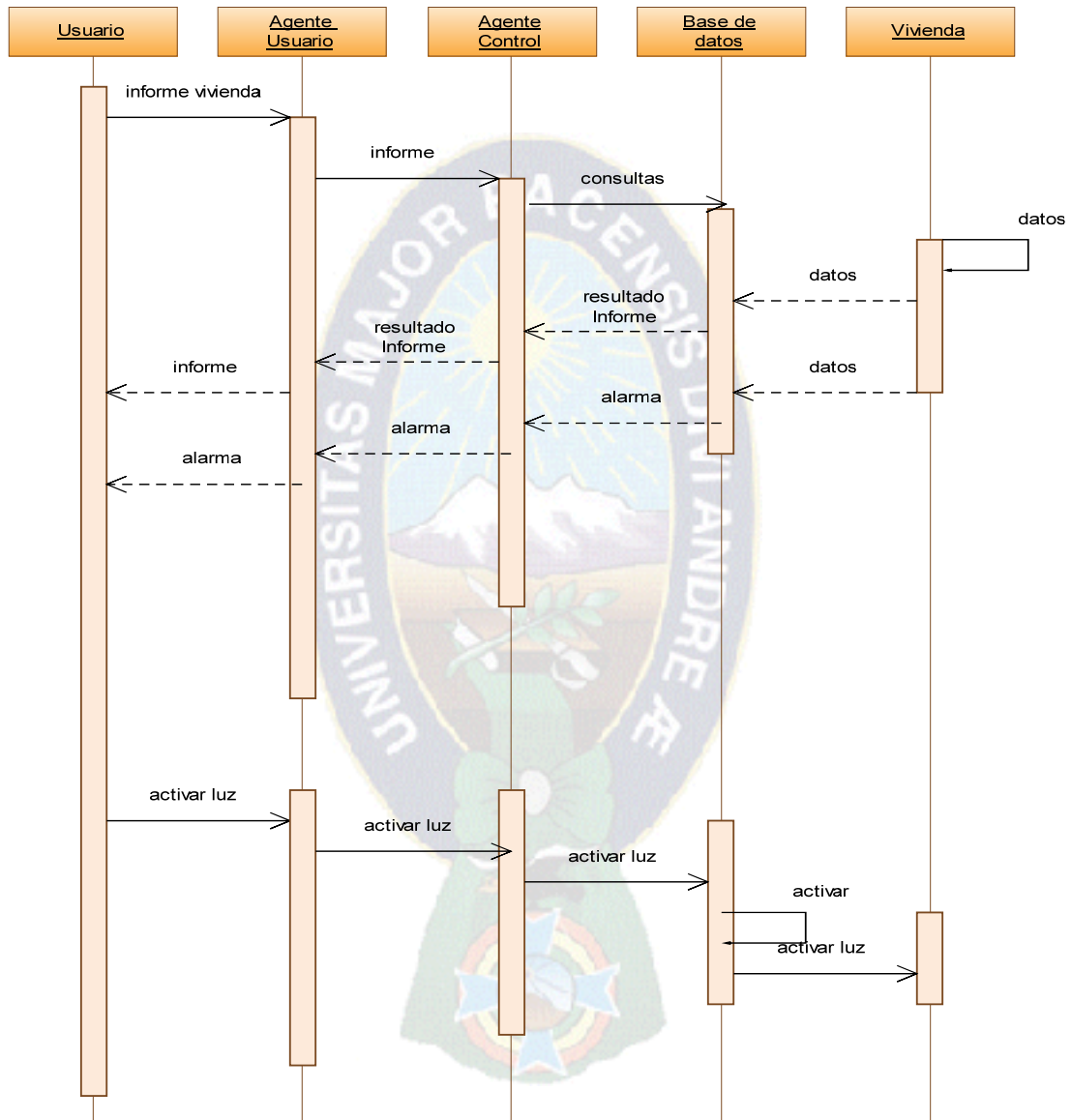


Figura 3.2: Diagrama de secuencia del sistema integrado
Fuente: Elaborado con base en (Pressman, 2005)

3.2.3 Diagrama de clases

A continuación se muestra el diagrama de clases del sistema de control, ver Figura 3.3

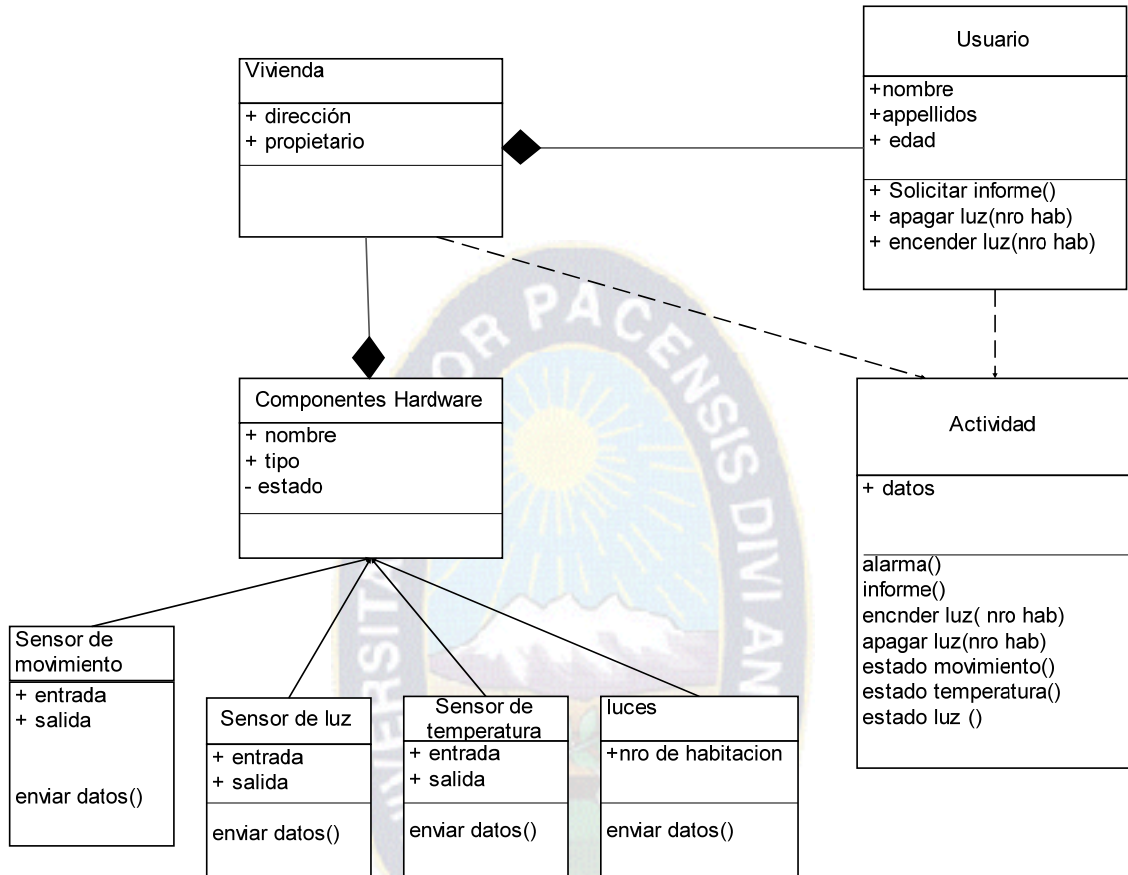


Figura 3.3: Diagrama de clases del sistema integrado

Fuente: Elaborado con base en (Pressman, 2005)

3.2.4 Identificación de tipos de agente

Se identifica los siguientes agentes:

1. Agente para usuario o dispositivo.
2. Agente para control.

Si se quiere ampliar, estos conceptos de agentes se definen en el capítulo 2 punto 2.12

Se determina un agente móvil para brindar información y permitir gestión en seguridad de la vivienda desde un dispositivo móvil mediante una aplicación móvil (Agente Usuario) y un agente control que supervise el estado de la vivienda mediante los datos adquiridos de cada sensor (Agente Control).

Tabla 3.6: Descripción del entorno de agentes.
Fuente: Elaborado con base en (Russell y Norving, 1999)

Tipo de Agente	Medidas de Rendimiento	Entorno	Actuadores	Sensores
Agente Usuario	1. Usuario satisfecho.	1. Usuario (dispositivo móvil)	1. Envío de mensajes 2. Recepción de mensajes.	1. Interfaz de aplicación móvil.
Agente Control	1. Recuperar información de acuerdo a la petición del usuario.	1. Detalle de entradas y estado de sensores y actuadores	1. Envío de mensajes 2. Recepción de mensajes.	1. Código de recuperar información.

3.2.5 Información de despliegue del agente

El Diagrama de despliegue representa la disposición de las instancias de componentes de ejecución de los agentes en instancias de nodos, un nodo es un recurso de ejecución. Ver Figura 3.4.

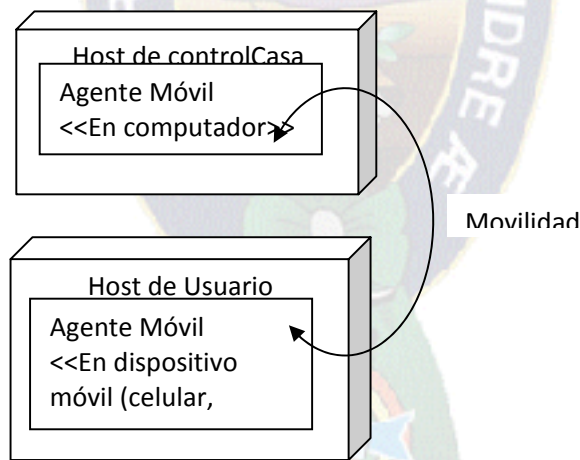


Figura 3.4: Diagrama de implementación (nivel instancia)
Fuente: Elaborado con base en (Pressman, 2005)

3.2.6 Identificación de elementos hardware

A continuación se muestra los elementos hardware que permite realizar el sistema integrado de control de seguridad.

Tabla 3.7: Identificación de elementos hardware.
Fuente: Elaborado con base en (Arudino, 2010)

Nombre	Descripción
Placa Arduino UNO	Placa electrónica que a través de los pines de entrada se obtiene los datos de cada sensor y a través de sus pines de salida encarga a los actuadores sus funciones.
Sensor de movimiento PIR	Sensor de movimiento, está alerta de algún movimiento extraño que pueda suceder dentro de la vivienda.
Sensor de color main actiuity	Sensor de luz que detalla la intensidad de luz que existe dentro de la vivienda.
Sensor de temperatura LM35	Sensor de temperatura y humedad, detalla la intensidad de temperatura y humedad dentro de la vivienda.
Led (actuador)	Dispositivo electrónico que emite luz, activados cuando lo desee el usuario.
Dispositivo móvil	Celular o Tablet con sistema operativo Android versión 4.0 en adelante y memoria interna disponible de 10 mg en adelante.

3.3 DISEÑO Y ARQUITECTURA

El diseño implementa todos los requisitos explicito de la fase de análisis del sistema también proporciona una imagen completa del software. Esta fase se enfoca a realizar la arquitectura del sistema integrado.

3.3.1 Arquitectura del prototipo

En la Figura 3.5 se detalla la arquitectura del sistema integrado, con los elementos y herramientas de funcionamiento.

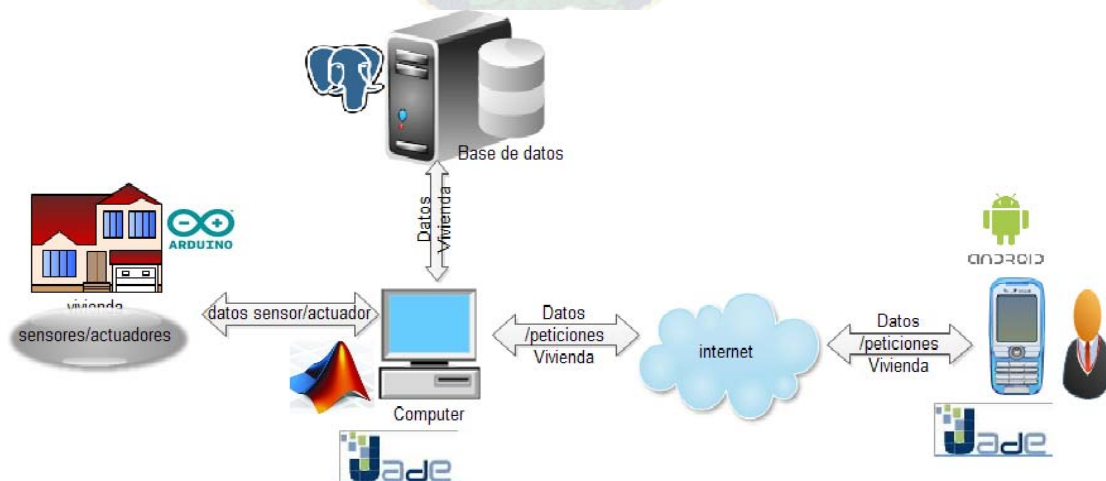


Figura 3.5: Estructura funcional del sistema integrado.
Fuente: Elaborado con base en (Pressman, 2005)

El prototipo se encuentra dividido en 3 módulos. Adquisición de datos, Control inteligente y Comunicación al usuario final. Ver Figura 3.6.

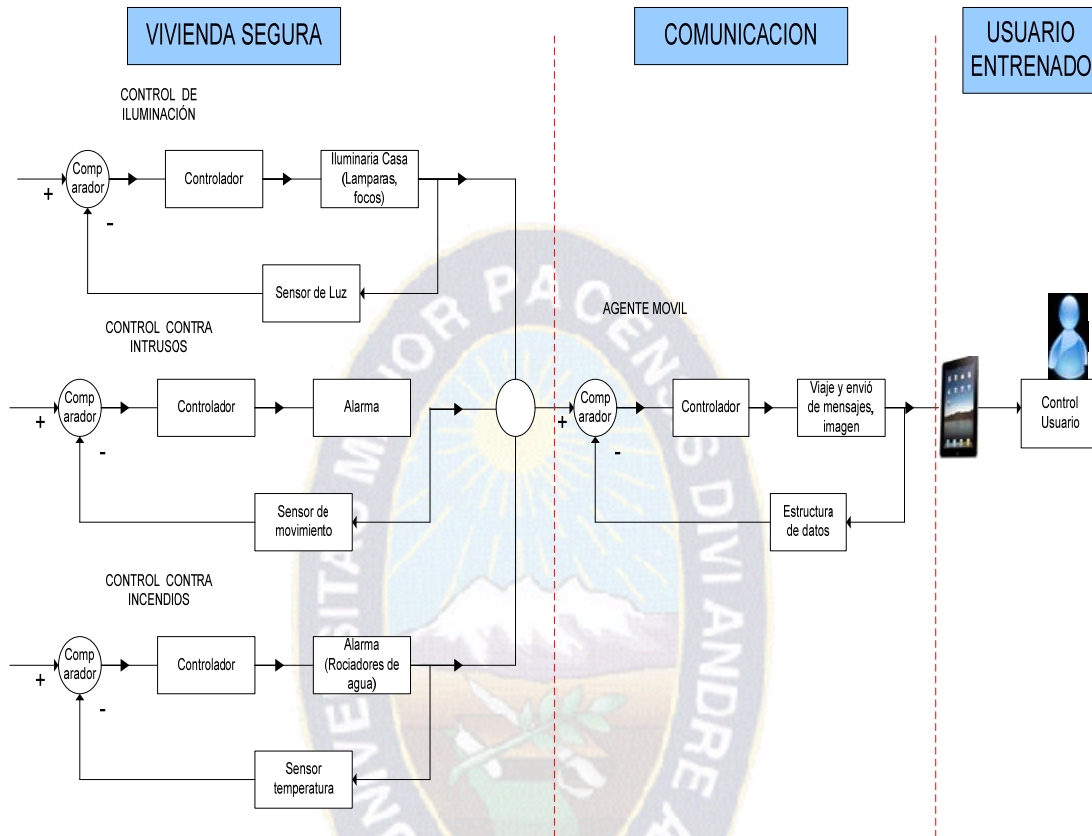


Figura 3.6: Estructura de implicados en el modelo

Fuente: elaborado con base en (Pressman, 2005)

3.3.1.1 Adquisición de datos: Se extrae información del ambiente (vivienda) a través sensores conectados a la placa Arduino UNO quien interviene para mandar información a Matlab-Simulink en donde se realiza un modelo de bloques para trabajar con los sensores y actuadores.

3.3.1.2 Control inteligente: La información recolectada gracias al primer módulo es procesada por la herramienta fuzzy logic toolbox de Simulink, quien determina si se envían mensajes de alerta o informe general, la cual es clasificada de acuerdo a reglas de lógica difusa de las entradas de cada sensor, las mismas son guardadas en la base de datos Historial.

3.3.1.3 Comunicación al usuario final: La información enviada al usuario final se realiza a través de agentes móviles, para tal comunicación se aplica dos agentes: Agente control y Agente usuario.

- **Agente control:** Se encuentra en una computadora convencional, es el agente que realiza la comunicación con los datos adquiridos del ambiente, extrae los últimos datos guardados en la base de datos y los envía al agente usuario.
- **Agente usuario:** Se encuentra en un dispositivo móvil, el usuario tiene una aplicación con la cual puede solicitar informes o enviar instrucciones a actuadores, los mismos que se enviarán al agente control quien realiza la comunicación con los actuadores para ejecutar las instrucciones enviadas por el usuario.

La comunicación entre los agentes se realiza mediante la plataforma JADE e internet, ambos agentes tienen conexión a internet, pues el usuario puede realizar las actividades ya mencionadas desde cualquier punto siempre y cuando esté conectado a internet.

3.3.2 Arquitectura de componentes electrónicos de control de seguridad

A continuación en la Figura 3.7 se encuentra el circuito electrónico en el cual se implementan los sensores y actuadores que interactúan en la vivienda, manteniendo actualizada la información del ambiente de la vivienda.

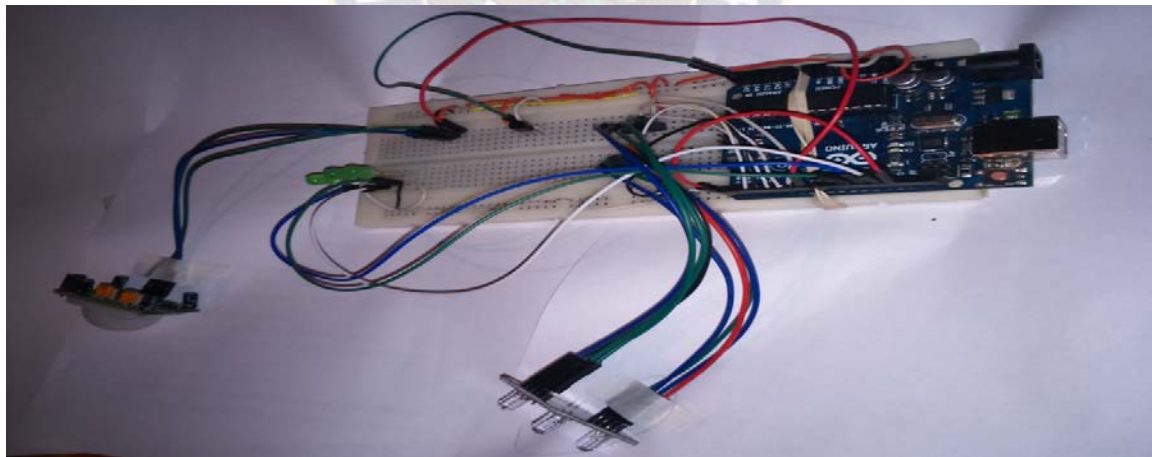


Figura 3.7: Arquitectura de componentes electrónicos del sistema integrado de seguridad.

3.3.3 Diseño de bloques de control de seguridad de vivienda en Simulink

A continuación en la Figura 3.8 se observa el modelo de bloques del sistema de control, el diseño en el cual se obtiene la información de los elementos de hardware instalados en la vivienda, procesa los datos por un modelo de lógica difusa y los registra en la base de datos.

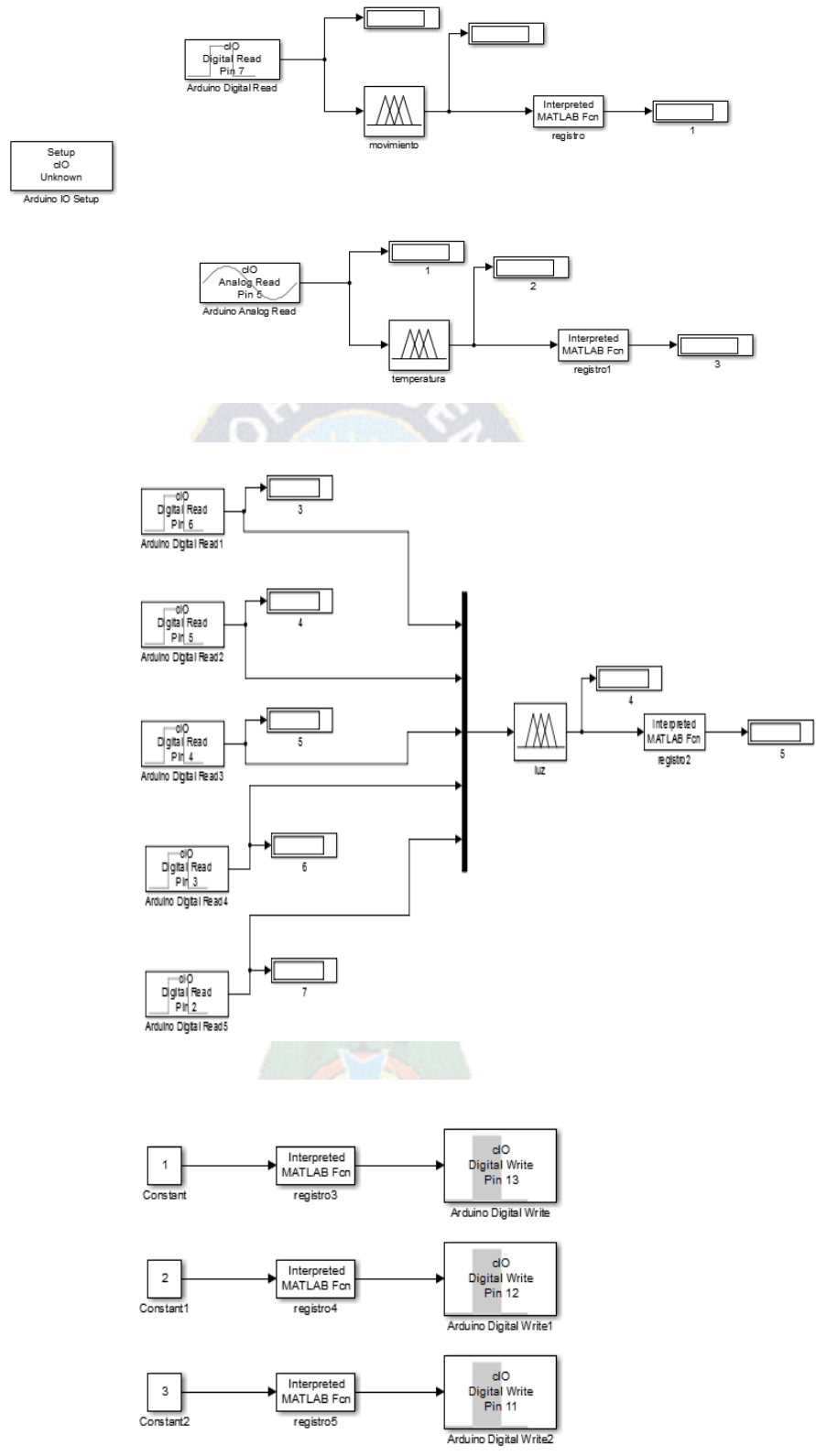


Figura 3.8: Modelo de bloques del sistema integrado de seguridad.

3.3.4 Diseño de interfaz de Usuario.

El diseño de interfaz de usuario muestra la interfaz de la aplicación Android para el usuario, misma que permite realizar una gestión de la vivienda recibiendo información y enviado peticiones de informe o de activado/desactivado de alarma o luz.

La aplicación Android tendrá las siguientes interfaces:

1. Autenticación: Se realiza la debida autenticación de usuarios, ver Figura 3.9

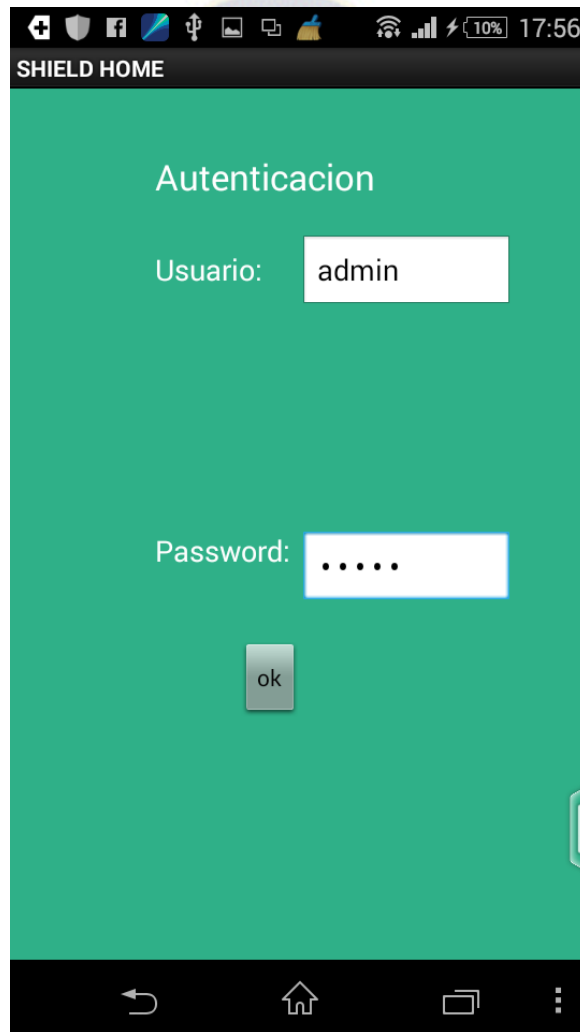


Figura 3.9: Interfaz de autenticación.

3. Informe. Se realiza consulta del estado actual de la vivienda, ver Figura 3.10



Figura 3.10: Interfaz de informe

4. Encendido de luces. Muestra botones para encendido/apagado de luces, ver Figura 3.11



Figura 3.11: Interfaz de encendido de luces

5. Historial. Muestra actividades realizadas en la vivienda, ver Figura 3.12

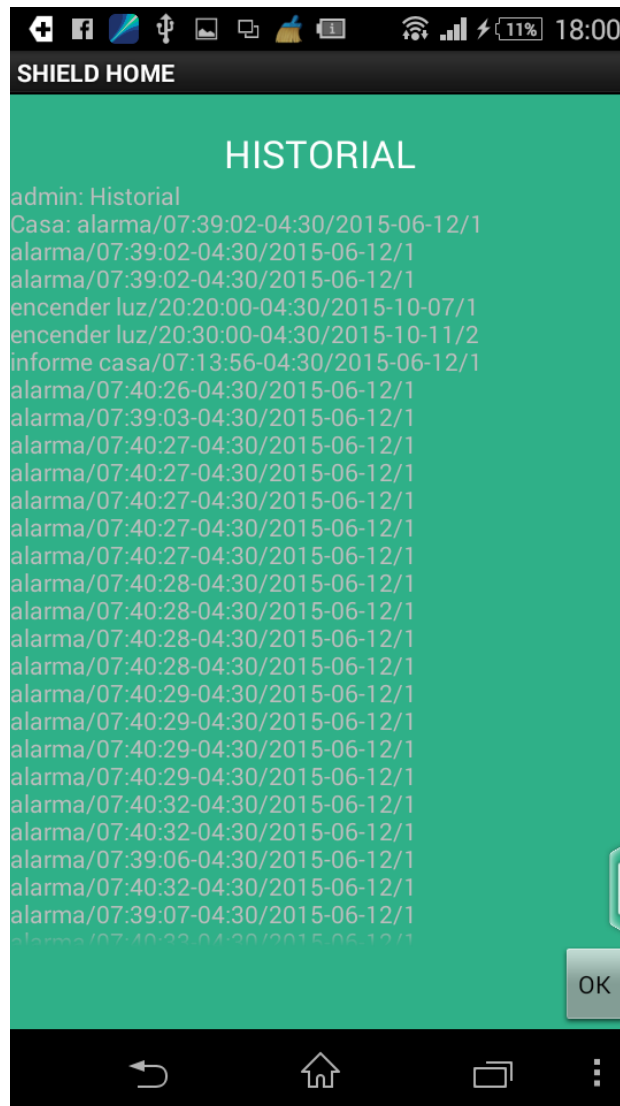


Figura 3.12: Interfaz de Historial

6. Usuario. Muestra usuarios participantes de la gestión de seguridad de la vivienda, ver Figura 3.13



Figura 3.13: Interfaz de usuarios

3.4 DESARROLLO

A continuación se detalla la elaboración del prototipo elaborada con la metodología Scrum que se detalla en el capítulo 2, punto 2.18.

Una vez establecidos los objetivos del prototipo y analizado el contexto en el que se llevará a cabo, es necesario planificar cómo se ejecutará el desarrollo del prototipo, es decir, realizar un diseño del mismo, se han considerado los siguientes aspectos:

1. En el diseño procedimental se trabaja con la metodología Scrum por ser una metodología iterativa incremental.
2. Asignación de Roles. Se realiza los detalles de los componentes y la asignación de tareas para desarrollar el prototipo.
3. Plan de Ejecución de desarrollo. Planificación inicial que se utilizará para ejecutar el desarrollo del proyecto y observar, a través de datos empíricos, los acontecimientos relativos a la evolución del producto.

3.4.1 Plan de Ejecución de desarrollo

Dado que las metodologías ágiles no siguen planificaciones detalladas a medio y largo plazo sino que se adaptan a las condiciones del mercado mediante planificaciones de ciclos de desarrollo cortos. Se trata de una planificación aproximada que puede sufrir variaciones a lo largo del desarrollo. Se ha considerado dividir en tres partes el estudio:

1. Analizar bibliografía relativa a las áreas principales en las que se enmarca el diseño del prototipo.
2. Se utiliza la metodología SCRUM en el desarrollo. Se ha planificado que esta parte conste de 9 Sprints de una duración aproximada de tres semanas cada uno, aunque se contemplan posibles variaciones que se presenten durante el desarrollo.

3.4.1.1 Diagrama de actividades planificado

A continuación, en las tablas 3.8, 3.9 y 3.10, se visualizan gráficamente las planificaciones de las distintas partes que forman el desarrollo temporal del experimento mediante diagramas de Gantt.

Tabla 3.8: Diagrama Gantt: Análisis preliminar e identificación del problema.

Fuente: Elaboración propia

Análisis e identificación de problemas												
Nombre	Mes 1				Mes 2				Mes 3			
	1° Sem	2° Sem	3° Sem	4° Sem	1° Sem	2° Sem	3° Sem	4° Sem	1° Sem	2° Sem	3° Sem	4° Sem
Estudio de Scrum												
Sprint 1: Estudio de Adquisición de datos												
Sprint 2: Estudio de Control inteligente												
Sprint 3: Estudio de aplicación para el usuario final												

Tabla 3.9: Diagrama Gantt: Desarrollo del proyecto.

Fuente: Elaboración propia

Desarrollo del prototipo																				
Nombre	Mes 3				Mes 4				Mes 5				Mes 6				Mes 7			
	1° Sem	2° Sem	3° Sem	4° Sem	1° Sem	2° Sem	3° Sem	4° Sem	1° Sem	2° Sem	3° Sem	4° Sem	1° Sem	2° Sem	3° Sem	4° Sem	1° Sem	2° Sem	3° Sem	4° Sem
Sprint 4																				
Sprint 5																				
Sprint 6																				
Sprint 7																				
Sprint 8																				
Sprint 9																				

Tabla 3.10: Diagrama Gantt: Evaluación y obtención de resultados.

Fuente: Elaboración propia

Evaluación y obtención de resultados								
Nombre	Mes 7				Mes 8			
	1° Sem	2° Sem	3° Sem	4° Sem	1° Sem	2° Sem	3° Sem	4° Sem
Obtención de resultados (pruebas)								
Elaborar documentación con resultados finales								

3.4.2 Asignación de Roles

Los roles se dividen en 2 grupos: comprometidos e involucrados.

Tabla 3.11 Lista de comprometidos con el desarrollo del sistema
Fuente: Elaboración propia

Persona	Contacto	Rol
Blanca Alicia Machaca Azpe	alicia_azpe@hotmail.com	Scrum manager
Blanca Alicia Machaca Azpe	alicia_azpe@hotmail.com	Product owner
Blanca Alicia Machaca Azpe	alicia_azpe@hotmail.com	Desarrollador

Tabla 3.12 Lista de involucrados con el desarrollo del sistema
Fuente: Elaboración propia

Persona	Contacto	Rol
Miriam Rosa Machaca Azpe	Miriam.machacaazpe@gmail.com	Usuario
Lic. Menfy Morales Rios		Stakeholder
Dr. Guillermo Choque Aspiazu	guillermo.choque@hotmail.com	Manager

3.4.3 Pila de Productos

Tabla 3.13: Pila de producto Scrum
Fuente: Elaboración propia

Id	Tipo	Nombre	Sprint
1	Historial	Investigar elementos hardware (sensores, actuadores) para adquisición de datos	1
2	Historial	Investigar, testear placa Arduino UNO	1
3	Historial	Investigar sobre bibliotecas y controladores para conexión Matlab-Simulink y Arduino UNO	1
4	Historial	Investigar fuzzy logic (logica difusa)	2
5	Historial	Investigar bibliotecas sobre Matlab-Simulink y fuzzy logic	2
6	Historial	Investigar sobre bibliotecas y controladores para conexión Matlab-Simulink y PostgreSQL	2
7	Historial	Investigar sobre bibliotecas y controladores para conexión Matlab-Java	3
8	Historial	Investigar sobre bibliotecas y controladores para conexión Java-PostgreSQL	3
9	Historial	Investigar sobre JADE y agentes ineligerentes	3
10	Historial	Investigar sobre JADE y aplicaciones Android	3
11	Historial	Diseñar circuito de control en Arduino UNO	4

12	Historial	Diseñar modelo de control en Simulink	5
13	Historial	Control fuzzy sensor temperatura, fuzzy sensor de luz, fuzzy sensor movimiento	6
14	Historial	Guardado base de datos	7
15	Historial	Agente de control en el ordenador	8
16	Historial	Agente de comunicación a usuario en aplicación Android	9

3.4.3.1 Historias de usuario

A continuación se observa la historia de usuarios elaborados para la pila de productos de SCRUM.

Tabla 3.14: Historia de Usuario 1: Investigar elementos hardware para adquisición de datos
Fuente: Elaborado con base en (Pressman, 2005)

ID: 1	Identificar elementos hardware
Determinar que componentes hardware trabajaran en el sistema de control de seguridad con respecto a la iluminación, temperatura y movimiento. Conseguir los elementos hardware para el sistema de control de seguridad.	
Estimación: 1 semana	
Prioridad: alta	Dependiente de: Ninguno
Pruebas de aceptación	

Tabla 3.15: Historia de Usuario 2: Investigar, testear placa Arduino UNO
Fuente: Elaborado con base en (Pressman, 2005)

ID: 2	Investigar, testear placa Arduino UNO
Investigar funciones que realiza Arduino UNO, pines de entrada, pines de salida.	
Estimación: 1 semana	
Prioridad: alta	Dependiente de: Ninguno
Pruebas de aceptación	
Conectar componentes a pines de entrada y salida, comprobar con las señales adquiridas.	

Tabla 3.16: Historia de Usuario 3: Investigar sobre bibliotecas y controladores para conexión Matlab-Simulink y Arduino UNO

Fuente: Elaborado con base en (Pressman, 2005)

ID: 3	Investigar librería, controladores Matlab-Simulink y Arduino UNO
Determinar bibliotecas y controladores necesarios para conexión Simulink y Arduino UNO	
Estimación: 2	
Prioridad: alta	Dependiente de: 2
Pruebas de aceptación	
Diseño de modelo de bloques de pines de entrada de Arduino en Simulink	

Tabla 3.17: Historia de Usuario 4: Investigar lógica difusa

Fuente: Elaborado con base en (Pressman, 2005)

ID: 4	Investigar lógica difusa
Conocer sobre variables de estado, variables de control, membresía, reglas difusas.	
Estimación: ½ semana	
Prioridad: media	Dependiente de: Ninguno
Pruebas de aceptación	
Realizar diseño de lógica difusa para un sensor y comprobar las entradas y salidas según reglas difusas.	

Tabla 3.18: Historia de Usuario 5: Investigar bibliotecas sobre Matlab-Simulink y fuzzy logic

Fuente: Elaborado con base en (Pressman, 2005)

ID: 5	Investigar bibliotecas sobre Matlab-Simulink y fuzzy logic
Determinar librería para trabajar con bloques que trabajen con lógica difusa, enviar datos de salida a workspace de Matlab	
Estimación: ½ semana	
Prioridad: alta	Dependiente de: Ninguno
Pruebas de aceptación	
Integrar fuzzy-toolbox al modelo de adquisición de datos y comprobar que los datos se intercambian entre Simulink y Matlab	

Tabla 3.19: Historia de Usuario 6: Investigar sobre bibliotecas y controladores para conexión Matlab-Simulink y PostgreSQL

Fuente: Elaborado con base en (Pressman, 2005)

ID: 6	Investigar sobre bibliotecas y controladores para conexión Matlab-Simulink y PostgreSQL
Determinar bibliotecas y controladores que permitan realizar la conexión con PostgreSQL e intercambiar datos con modelos en Simulink.	
Estimación: 1 semana	
Prioridad: alta	Dependiente de: Ninguno
Pruebas de aceptación	
<ul style="list-style-type: none"> • Realizar conexión Matlab y PostgreSQL. • Realizar consultas desde Matlab. 	

Tabla 3.20: Historia de Usuario 7: Investigar sobre bibliotecas y controladores para conexión Matlab-Java

Fuente: Elaborado con base en (Pressman, 2005)

ID: 7	Investigar sobre bibliotecas y controladores para conexión Matlab-Java
Identificar librería y drives para realizar conexión de Java con Matlab, ejecución de funciones, envío de parámetros	
Estimación: 1	
Prioridad: 100	Dependiente de: Ninguno
Pruebas de aceptación	
<ul style="list-style-type: none"> • Ejecutar función conexión con envío de parámetros, sin errores. • Enviar parámetros inválidos y comprobar que se indica error 	

Tabla 3.21: Historia de Usuario 8: Investigar sobre bibliotecas y controladores para conexión Java-PostgreSQL

Fuente: Elaborado con base en (Pressman, 2005)

ID: 8	Investigar sobre bibliotecas y controladores para conexión Java-PostgreSQL
Identificar librería y drives para realizar conexión de Java con PostgreSQL, crear base de datos, realizar consultas a la base de datos desde Java.	
Estimación: ½ semana	
Prioridad: alta	Dependiente de: Ninguno

Pruebas de aceptación	
<ul style="list-style-type: none"> • Crear base de datos desde Java. • Realizar consultas a base de datos desde Java 	

Tabla 3.22: Historia de Usuario 9: Investigar sobre JADE y agentes inteligentes

Fuente: Elaborado con base en (Pressman, 2005)

ID: 9	Investigar sobre JADE y agentes inteligentes
Identificar tipos de agente involucrado en el sistema Investigar JADE componentes, bibliotecas, controladores instalar JADE	
Estimación: 1 semana	
Prioridad: alta	Dependiente de: Ninguno
Pruebas de aceptación	
<ul style="list-style-type: none"> • Crear un agente de Prueba • Ejecutar agente de Prueba en JADE 	

Tabla 3.23: Historia de Usuario 10: Investigar sobre JADE y aplicaciones Android

Fuente: Elaborado con base en (Pressman, 2005)

ID: 10	Investigar sobre JADE y aplicaciones Android
Investigar cómo integrar JADE en una aplicación Android y su comunicación mediante internet	
Estimación: 2	
Prioridad: alta	Dependiente de: Ninguno
Pruebas de aceptación	
Crear un agente en una aplicación Android Instalar la aplicación en un dispositivo móvil con conexión a internet y comprobar q se ejecuta en JADE	

Tabla 3.24: Historia de Usuario 11: Diseñar circuito de control en Arduino UNO

Fuente: Elaborado con base en (Pressman, 2005)

ID: 11	Diseñar circuito de control en Arduino UNO
Reconocer pines de entrada , salida, y conexión de la placa con los sensores placa Diseñar un circuito con los componentes hardware necesarios para el prototipo de sistema de control inteligente.	
Estimación: 2 semana	
Prioridad: alta	Dependiente de: 1,2,3
Pruebas de aceptación	

Exponer sensor de luz a una variación de intensidad de luz, obtener datos variados según a la intensidad. Exponer sensor de temperatura a una variación temperatura, obtener datos variados según a la temperatura. Exponer sensor de movimiento a una variación de movimientos, obtener datos variados según la distancia.
Visualizar los datos en consola de Arduino.

Tabla 3.25: Historia de Usuario 12: Diseñar modelo de control en Simulink

Fuente: Elaborado con base en (Pressman, 2005)

ID: 12	Diseñar modelo de control en Simulink
<p>Instalar bibliotecas para interactuar placa Arduino UNO y Simulink Diseñar diagrama de bloques del prototipo de sistema de control de los sensores y actuadores conectados a los pines de entrada y salida de la placa.</p>	
Estimación: ½ semana	
Prioridad: 100	Dependiente de: 1,3
Pruebas de aceptación	
<p>Exponer sensor de luz a una variación de intensidad de luz, obtener datos variados según a la intensidad. Exponer sensor de temperatura a una variación temperatura, obtener datos variados según a la temperatura. Exponer sensor de movimiento a una variación de movimientos, obtener datos variados según la distancia. Visualizar los datos un componente de Simulink como “display”.</p>	

Tabla 3.26: Historia de Usuario 13: Lógica difusa para: Sensor temperatura, sensor de luz, sensor movimiento

Fuente: Elaborado con base en (Pressman, 2005)

ID: 13	Control fuzzy sensor temperatura, fuzzy sensor de luz, fuzzy sensor movimiento
Definir variables es estado y control, realizar funciones de pertenencia, reglas de producción y salidas, para las entradas de cada sensor.	
Estimación: 1 semana	
Prioridad: media	Dependiente de: 4,5,11,12
Pruebas de aceptación	
Enviar entradas de cada sensor y ver salida para determinar si se encuentra dentro del conjunto de soluciones	

Tabla 3.27: Historia de Usuario 14: Guardado base de datos

Fuente: Elaborado con base en (Pressman, 2005)

ID: 14	Guardado base de datos
<p>Diseñar una base de datos para los datos de entrada y salida del sistema de control Realizar funciones en Matlab que permitan acceder a la base de datos y realizar consultas.</p>	
Estimación: 1 semana	

Prioridad: media	Dependiente de: 6,11,12,13
Pruebas de aceptación	
Realizar consultas a partir de una función de Matlab con respuestas positivas.	

Tabla 3.28: Historia de Usuario 15: Agente de control en el ordenador

Fuente: Elaborado con base en (Pressman, 2005)

ID: 15	Agente de control en el ordenador
Instalar bibliotecas necesarias para la programación de agentes. Desarrollar un agente con el lenguaje de programación Java en eclipse. Investigar proyecto JADE CHAT. Realizar envío de datos obtenidos de la base de datos ya procesados.	
Estimación: 3 semanas	
Prioridad: alta	Dependiente de: 9,11,12,13,14
Pruebas de aceptación	
Agente de control se encuentre en la plataforma Jade dispuesto a trabajar con otro agente. Realizar envío de mensajes a otros agentes con respuestas positivas	

Tabla 3.29: Historia de Usuario 16: Agente de comunicación a usuario en aplicación Android

Fuente: Elaborado con base en (Pressman, 2005)

ID: 16	Agente de comunicación a usuario en aplicación Android
Desarrollar aplicación Android que tenga un agente móvil para la comunicación con el agente de ordenador situado en la vivienda.	
Estimación: 1 semana	
Prioridad: alta	Dependiente de: 10,15
Pruebas de aceptación	
<ul style="list-style-type: none"> • Ocasionar un pequeño incendio y comprobar que se obtiene datos del sensor de temperatura. • Esperar anochecer comprobar que se obtiene datos del sensor de luz. • Realizar movimientos inusuales y comprobar que se obtienen datos del sensor de movimiento. 	

3.4.4 Sprint 1- Sprint 3

El siguiente Sprint apoya al cumplimiento del objetivo específico número 1, “Buscar y seleccionar elementos computacionales de hardware y software para el diseño del prototipo de un sistema de control de vivienda”

Se considera importante el estudio del diseño del prototipo presentado en esta tesis, por tal razón se le asigno un tiempo considerable, los tres sprint realizan el análisis e identificación

del problema. Se pretende realizar un detallado análisis que permitiese obtener interpretaciones rigurosas y proporcionarse de información suficiente para realizar un desarrollo más detallado.

En el Sprint 1 se llevaron a cabo los estudios en los cuales se logran identificar los componentes que formaran parte del sistema de control, así mismo, se identifica bibliotecas que permitan tener una interacción de la placa Arduino UNO con Matlab.

En el Sprint 2 se llevaron a cabo los estudios en los cuales se realiza hacer un control de lógica difusa de los sensores, los cuales son entradas del ambiente de la vivienda. La información procesada por la lógica difusa de las entradas de los sensores es guardada en la base de datos llamada Historial.

En el Sprint 3 se llevaron a cabo los estudios en los cuales se identifican los recursos de software necesarios para la comunicación entre agentes, además de la su estructura, su programación y su alcance.

3.4.5 Sprint 4

El siguiente Sprint apoya al inicio del cumplimiento del objetivo específico número 2, “Diseñar un modelo de adquisición de datos de una vivienda utilizando los elementos hardware seleccionados”

La prioridad era tener en primera instancia el circuito de control que tendrá la vivienda. El objetivo del sprint 4 es realizar el diseño del circuito de control, este que interactúa con Arduino UNO y los componentes electrónicos necesarios para sistema de control de la vivienda.

Para realizar la planificación de este sprint se descompusieron las historias de usuario implicadas, 4 y 5 respectivamente, en tareas que habría que realizar para el cumplimiento de dichas historias de usuario.

Se realizo las siguientes tareas:

Tarea 1: Conectar sensores a pines de entrada de Arduino UNO

La tarea consiste en realizar un diseño de los sensores de entrada del sistema de control de la vivienda, los sensores a conectar son los siguientes:

1. Sensor de color (sensor de luz).- Es conectada al pin número 2, 3, 4, 5, 6 de la placa, este sensor es de entrada digital y percibe la intensidad de luminosidad existe en la vivienda.
2. LM35 (Sensor de temperatura).-Es conectada al pin número A5 de la placa, este sensor es de entrada analógica y percibe la intensidad de calor existente en la vivienda.
3. Sensor PIR (Sensor de movimiento).-Es conectada al pin número 7 de la placa, la misión del sensor es percibir el movimiento existen dentro de la vivienda.

Tarea 2: Conectar actuadores a pines de salida de Arduino UNO

La tarea consiste en realizar un diseño de los actuadores los cuales realizaran una tarea determinada por el usuario los componentes son los siguientes:

1. Luces (Led).- El led es conectado al pin de salida digital número 11, 12, 13 de la placa, a través del cual se emite una señal para su encendido o apagado.

3.4.6 Sprint 5

El siguiente Sprint que se encuentra vinculado con el Sprint 4 incrementa al inicio del cumplimiento del objetivo específico número 2, “Diseñar un modelo de adquisición de datos de una vivienda utilizando los elementos hardware seleccionados”

El objetivo del sprint 5 es realizar el diseño de bloques del circuito de control en Simulink, para realizar un control de lógica difusa a las entradas analógicas conectadas en la placa.

Para realizar la planificación de este sprint se descompusieron las historias de usuario implicadas, 6 y 7 respectivamente, en tareas que habría que realizar para el cumplimiento de dichas historias de usuario.

Se realizó las siguientes tareas:

Tarea 1: Instalar librería ArduinoIO en Simulink

Se instala un paquete llamado ArduinoIO el cual está programado para realizar modelo de bloques en Simulink, la librería tiene como componentes bloques de entrada y salida digital para pines de Arduino, bloques entrada y salida analógica para pines de Arduino, bloque de conexión al puerto al cual está conectada la placa Arduino.

Tarea 2: Realizar diseño del modelo del control se seguridad de la vivienda, se realiza un diseño con las especificaciones del circuito realizado en Sprint 4, una vez integrados los pines de la placa al modelo de bloques se pasa a realizar el tratamiento de lógica difusa.

3.4.7 Sprint 6

El siguiente Sprint que se encuentra vinculado con el Sprint 5 en la cual el cumplimiento del objetivo específico número 2, “Diseñar un modelo de adquisición de datos de una vivienda utilizando los elementos hardware seleccionados”, se encuentra cerca de concluirse gracias a las tareas realizadas en este Sprint.

El objetivo del Sprint 6 consiste en realizar reglas de control para las entradas analógicas o digitales y determinar si es necesario realizar una alarma o no. En este Sprint se realiza un control difuso con el método de Mamdani, las tareas a realizar son las siguientes:

Tarea 1: Realizar diseño de control difuso de entrada digital sensor de color

1. Identificar variables de estado y control.

Variables de estado: Se describe como la entrada y entre sus parámetros se encuentra: poco, medio, nada.

Variables de control: Se describe como la salida entre sus parámetros se encuentra: claro, medio, oscuro.

Identificar los términos lingüísticos vinculados a cada variable.

- **Variable de estado: Intensidad de luz**

Se pretende controlar la intensidad de luz de una vivienda a partir de los parámetros de intensidad de luz brindados por un sensor de color. El sensor de color detecta colores del mundo real como RGB.

Para el sistema de control se tendrá 5 entradas, las cuales son variaciones, comprendidas entre 0 y 1, de tal modo se cuenta con 5 funciones de pertenencia, A continuación en la Figura 3.14, 3.15, 3.16, 3.17, 3.18 se presenta la función de pertenencia del bloque de lógica difusa de la entrada de intensidad de luz.

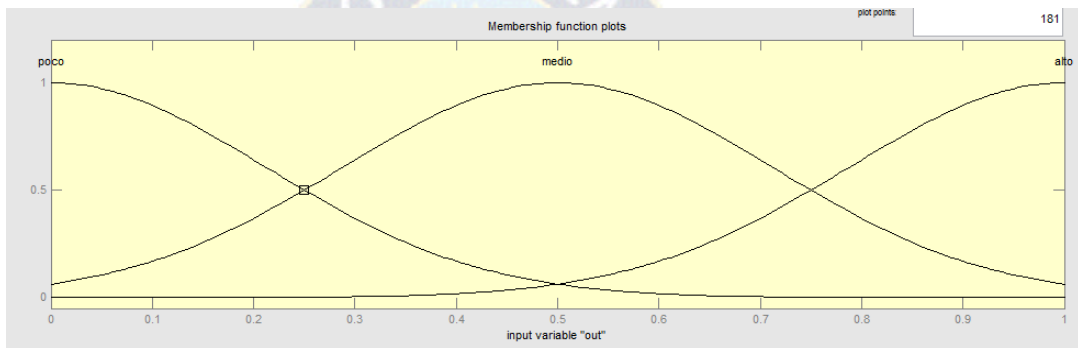


Figura 3.14: Función de pertenencia de la variable de estado: salida
Fuente: Elaboración propia

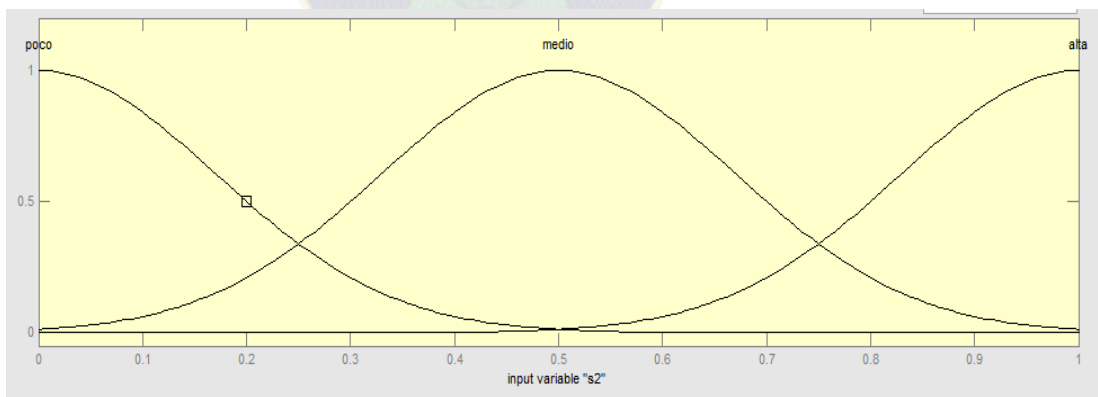


Figura 3.15: Función de pertenencia de la variable de estado: s2
Fuente: Elaboración propia

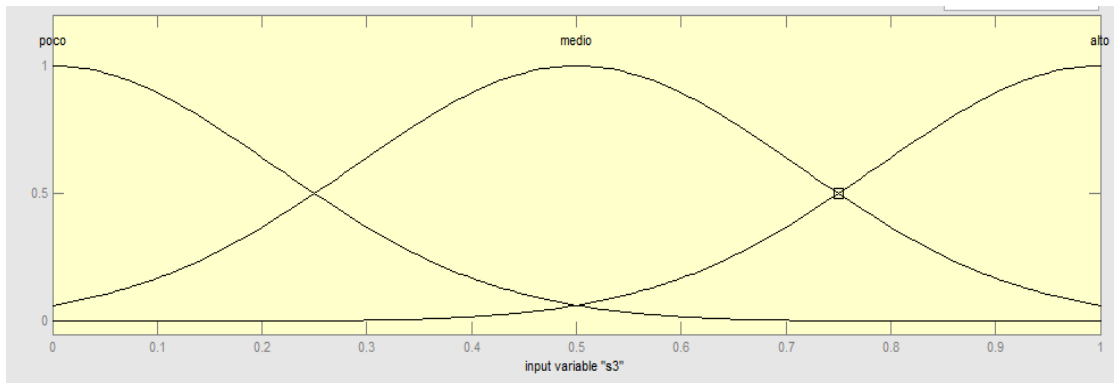


Figura 3.16: Função de pertinência de la variable de estado: s3
Fuente: Elaboración propia

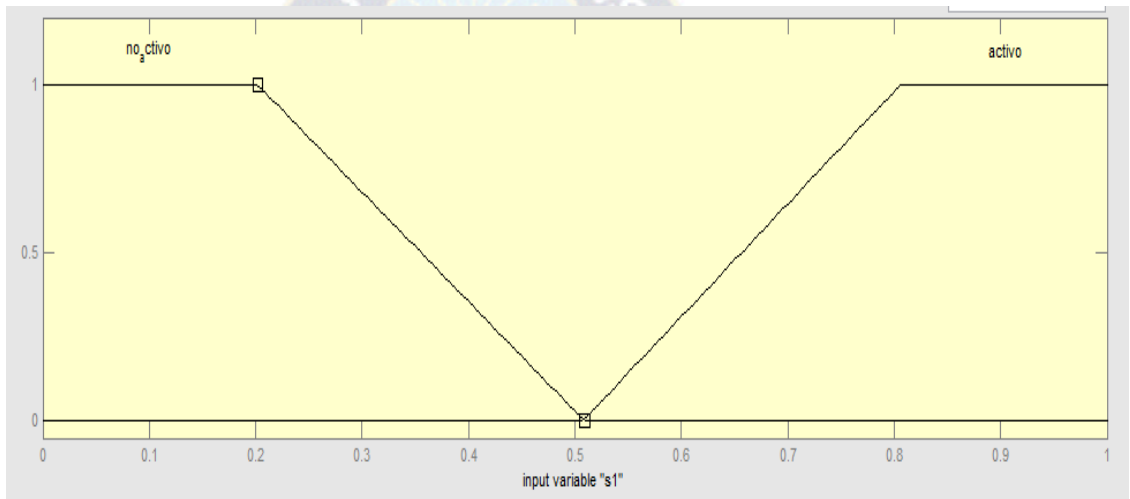


Figura 3.17: Função de pertinência de la variable de estado: s1
Fuente: Elaboración propia

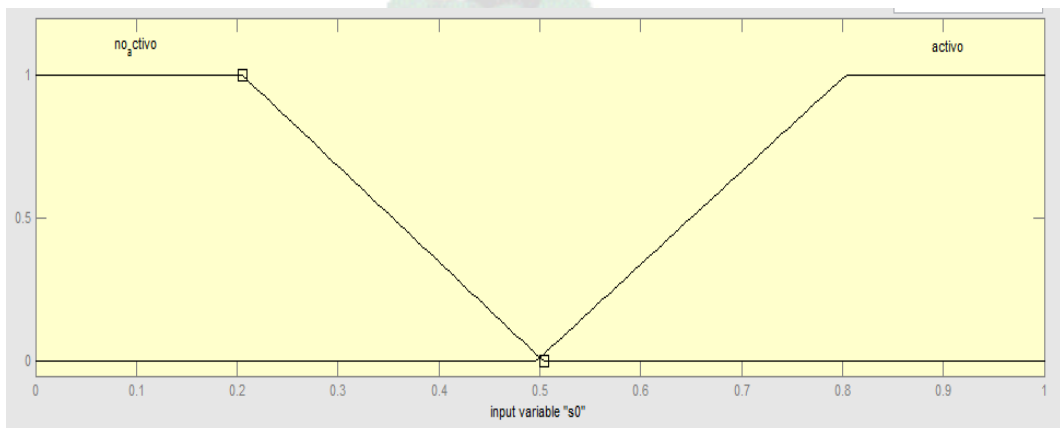


Figura 3.18: Função de pertinência de la variable de estado: s0
Fuente: Elaboración propia

- **Variable de control: Variación de luz**

Se asumen 2 etiquetas lingüísticas: informe y alarma. En la Figura 3.19 se observa la función de pertenencia a la cual está ligada.

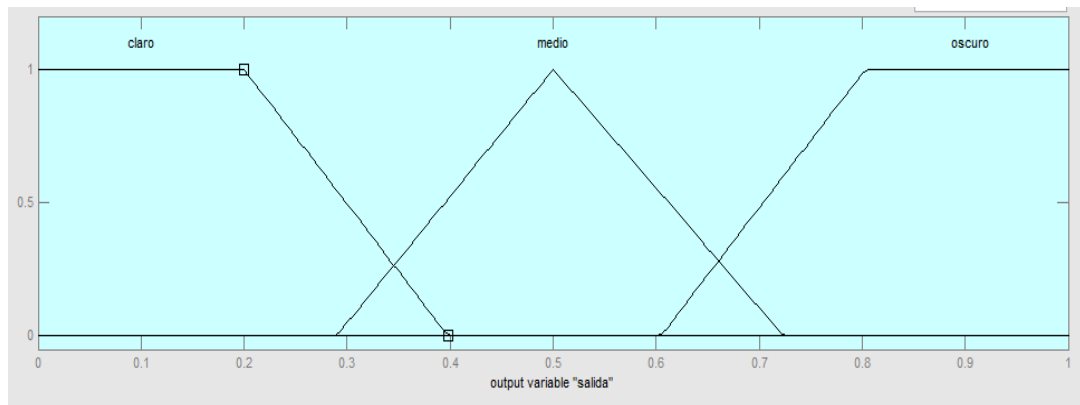


Figura 3.19: Función de pertenencia de la variable de control: variación de luz.

Fuente: Elaboración propia

2. Construir las reglas difusas que correspondan.

Se considera un listado de reglas de producción con operaciones de conectividad para la variable de control vinculado con la variable de estado. A continuación se realiza un detalle de las reglas de producción.

1. If (out is poco) and (s2 is poco) and (s3 is poco) and (s1 is activo) and (s0 is activo) then (salida is claro) (1)
2. If (out is alto) and (s2 is alto) and (s3 is alto) and (s1 is activo) and (s0 is activo) then (salida is oscuro) (1)
3. If (out is poco) and (s2 is medio) and (s3 is medio) and (s1 is activo) and (s0 is activo) then (salida is medio) (1)
4. If (out is medio) and (s2 is medio) and (s3 is medio) and (s1 is activo) and (s0 is activo) then (salida is medio) (1)
5. If (out is poco) and (s2 is poco) and (s3 is medio) and (s1 is activo) and (s0 is activo) then (salida is claro) (1)
6. If (out is alto) and (s2 is alto) and (s3 is medio) and (s1 is activo) and (s0 is activo) then (salida is oscuro) (1)

7. If (out is alto) and (s2 is alto) and (s3 is poco) and (s1 is activo) and (s0 is activo) then (salida is medio) (1)
8. If (out is alto) and (s2 is medio) and (s3 is poco) and (s1 is activo) and (s0 is activo) then (salida is medio) (1)
9. If (out is poco) and (s2 is medio) and (s3 is poco) and (s1 is activo) and (s0 is activo) then (salida is claro) (1)
10. If (out is alto) and (s2 is medio) and (s3 is alto) and (s1 is activo) and (s0 is activo) then (salida is oscuro) (1)
11. If (out is poco) and (s2 is medio) and (s3 is alto) and (s1 is activo) and (s0 is activo) then (salida is medio) (1)
12. If (out is alto) and (s2 is medio) and (s3 is poco) and (s1 is activo) and (s0 is activo) then (salida is medio) (1)

Tarea 2: Realizar diseño de control difuso de entrada analógica sensor de temperatura

1. Identificar variables de estado y control.

Variables de estado: Se describe como la entrada y entre sus parámetros se encuentra: muy bajo, bajo, normal, alta, muy alta.

Variables de control: Se describe como la salida entre sus parámetros se encuentra: alarma, informe.

2. Identificar los términos lingüísticos vinculados a cada variable.

- **Variable de estado: Temperatura**

Se pretende controlar el grado de temperatura de una vivienda a partir de los parámetros de temperatura brindados por sensor LM35. El sensor LM35 cuenta con un rango de funcionamiento de -55°C a 150°C , el valor podría tomar uno de los valores encontrados en el intervalo.

Para el sistema de control se tomara un intervalo de -20°C a 50°C , esto debido a que el sensor fue sujeto a pruebas que determinaban este intervalo de funcionamiento.

A continuación en la Figura 3.20 se presenta la función de pertenencia del bloque de lógica difusa de la variable de estado temperatura.

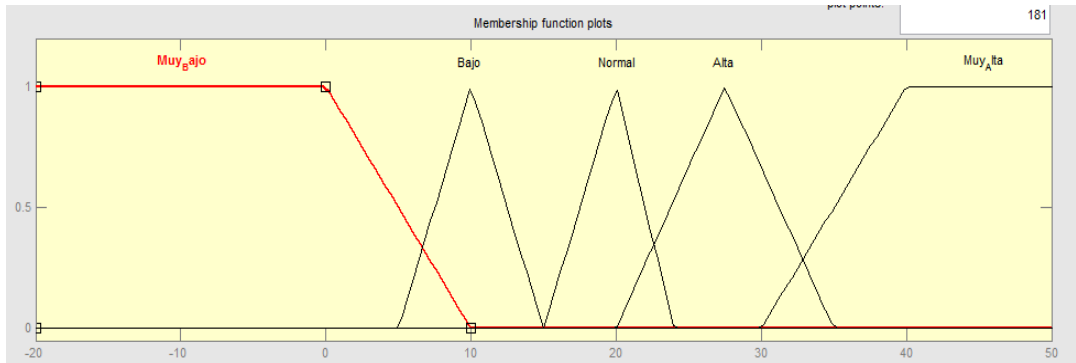


Figura 3.20: Función de pertenencia de la variable de estado: temperatura
Fuente: Elaboración propia

- **Variable de control: Variación de temperatura**

Se asumen 2 etiquetas lingüísticas: informe y alarma. En la figura 3.21 se observa la función de pertenencia a la cual está ligada.

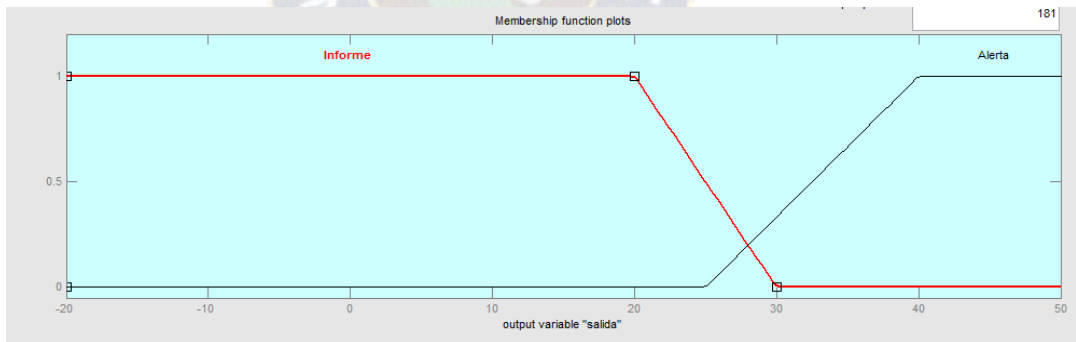


Figura 3.21: Función de pertenencia de la variable de control: variación de temperatura.
Fuente: Elaboración propia

3. Construir las reglas difusas que correspondan.

Se considera un listado de reglas de producción con operaciones de conectividad para la variable de control vinculado con la variable de estado. A continuación se realiza un detalle de las reglas de producción.

1. If (temperatura is Muy_Bajo) then (salida is Informe) (1)
2. If (temperatura is Bajo) then (salida is Informe) (1)
3. If (temperatura is Normal) then (salida is Informe) (1)
4. If (temperatura is Alta) then (salida is Informe) (1)
5. If (temperatura is Muy_Alta) then (salida is Alerta) (1)

Tarea 3: Realizar diseño de control difuso de entrada digital sensor de movimiento

1. Identificar variables de estado y control.

Variables de estado: Se describe como la entrada y entre sus parámetros se encuentra: nada, medio, mucho

Variables de control: Se describe como la salida entre sus parámetros se encuentra: alarma, informe.

2. Identificar los términos lingüísticos vinculados a cada variable.

- **Variable de estado: Movimiento**

Se pretende controlar los movimientos realizados de una vivienda a partir de los parámetros de movimiento brindados por el sensor PIR. El sensor PIR cuenta con un rango de funcionamiento de 0 y 1, el valor podría tomar uno de los valores encontrados en el intervalo.

Para el sistema de control se tomara un intervalo de 0 y 1, esto debido a que el sensor fue sujeto a pruebas que determinaban este intervalo de funcionamiento.

A continuación en la Figura 3.22 se presenta la función de pertenencia del bloque de lógica difusa de la variable de estado movimiento.

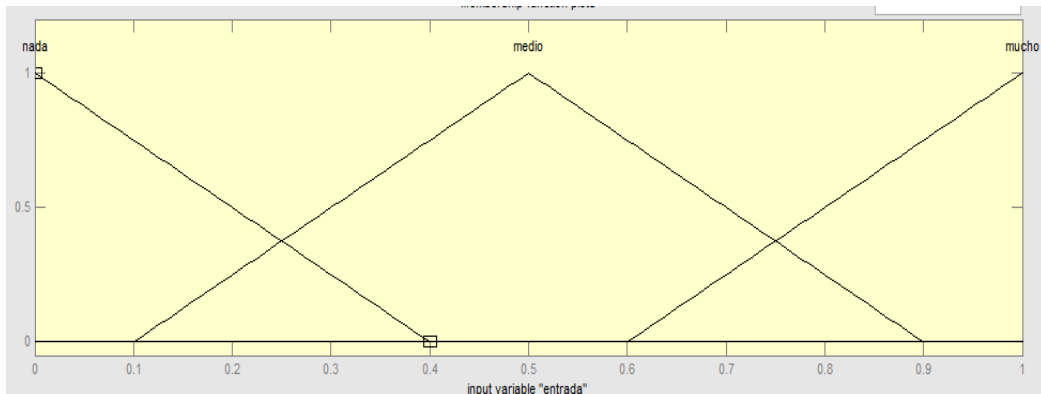


Figura 3.22: Función de pertenencia de la variable de estado: movimiento
Fuente: Elaboración propia

- **Variable de control: Variación de temperatura**

Se asumen 2 etiquetas lingüísticas: informe y alarma. En la figura 3.23 se observa la función de pertenencia a la cual está ligada.

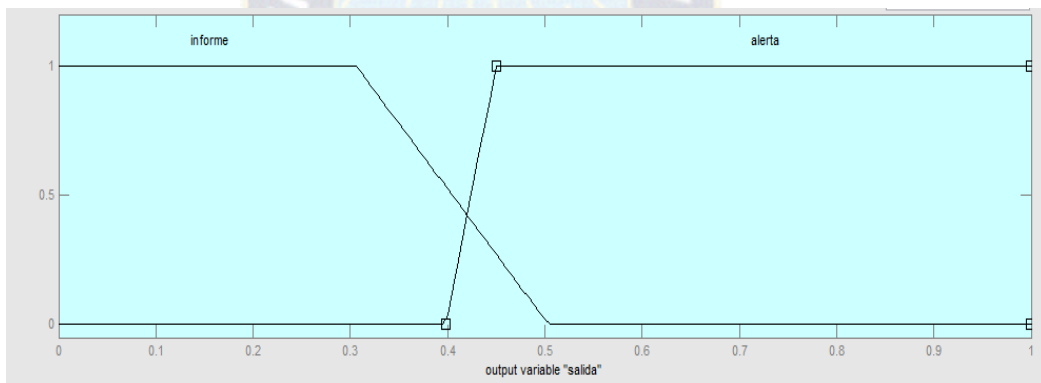


Figura 3.23: Función de pertenencia de la variable de control: variación de temperatura.
Fuente: Elaboración propia

3. Construir las reglas difusas que correspondan.

Se considera un listado de reglas de producción con operaciones de conectividad para la variable de control vinculado con la variable de estado. A continuación se realiza un detalle de las reglas de producción.

1. If (entrada is nada) then (salida is informe) (1)
2. 2. If (entrada is medio) then (salida is informe) (1)
3. 3. If (entrada is mucho) then (salida is alarma) (1)

3.4.8 Sprint 7

El siguiente Sprint que se encuentra vinculado con el Sprint 6 en la cual el cumplimiento del objetivo específico número 2, “Diseñar un modelo de adquisición de datos de una vivienda utilizando los elementos hardware seleccionados”, se encuentra concluido gracias a tareas realizadas.

Este Sprint tiene por objetivo realizar un diseño de base de datos al se guardarán lo datos recolectados por los sensores encontrados en la vivienda. Se realiza la siguiente tarea.

Tarea 1: Realizar diseño de base de datos para guardar datos de entrada y salida del sistema, la base de datos tendrá por nombre HISTORIAL.

Los datos serán guardados desde el modelo de bloques de Simulink mediante una función que realiza la conexión a la base de datos y posteriormente a la inserción de datos.

3.4.9 Sprint 8

El siguiente Sprint es de apoyo para el cumplimiento de la primera parte del objetivo específico número 3, “Diseñar un modelo de software mediado por un agente móvil que procese la información obtenida del modelo de hardware”,

En este Sprint se pasa a realizar al agente control que realizara el envío de alarmas o informes al agente usuario que se encuentra en el dispositivo del usuario.

Las tareas a realizar son las siguientes:

Tarea 1: Instalar Jade Java Eclipse: En primera instancia para realizar la programación de de agentes se instala Jade la plataforma que proporciona soporte en la comunicación de agentes.

Para la instalación es preciso descargar todo el paquete de jade 4.3.2 añadir el path a la computadora. Jade trabaja con el lenguaje de programación Java por tanto el IDE con el cual se trabaja es eclipse, para comprobar su correcta función se ejecuta jade.Boot donde se observa el interfaz en cual trabajaran los agente dicho interfaz se observa en la Figura 3.24.

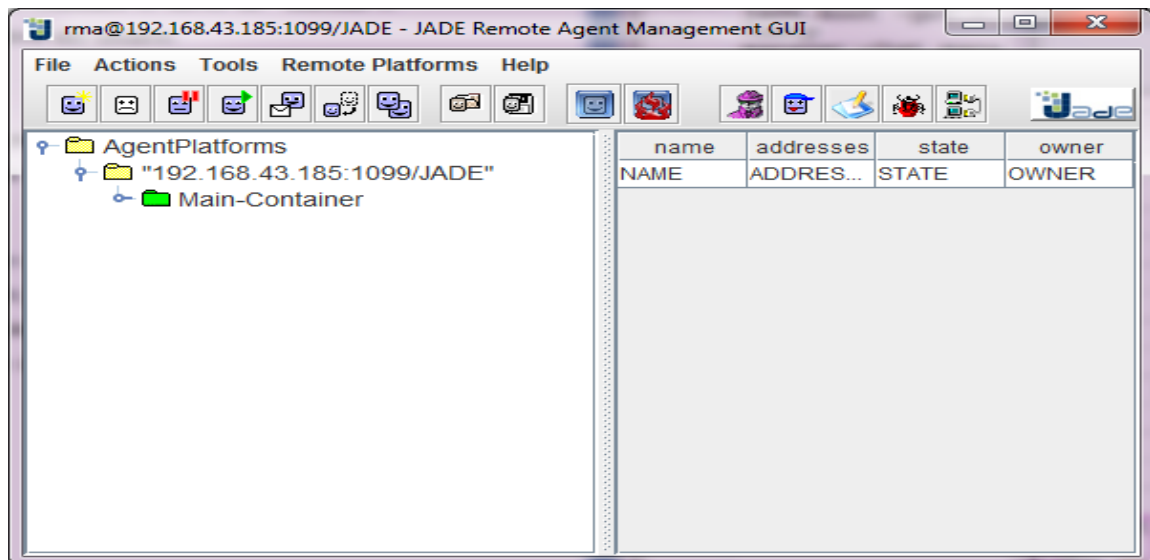


Figura 3.24: Interfaz de trabajo de Jade

Fuente: Elaboración propia

Tarea 2: Diseño de agente de control en la computadora: Finalizada la tarea numero 1 se procede a la programación del agente de control en el computador, como referencias se ha trabajado con un proyecto realizado con anterioridad llamado Jade chat, la presente tesis usa recursos de este proyecto.

Para trabajar con Jade se debe implementar la librería con la cual trabaja, en este caso en un archivo con extensión .jar, Jade.jar,

Se crea un agente en el ordenador llamado Agente control casa, el cual es el encargado de realizar una conexión con PostgreSQL y Matlab, en ambos casos se necesitan de bibliotecas que puedan colaborar para su interacción.

Para la conexión con Matlab se compila una función que entrega una paquete “.jar” la función esta en Matlab pero es ejecutada a través de Java mediante los envíos de petición del agente usuario.

3.4.10 Sprint 9

El siguiente Sprint es de apoyo para el cumplimiento de la total del objetivo específico número 3, “Diseñar un modelo de software mediado por un agente móvil que procese la información obtenida del modelo de hardware”, de la misma manera se realiza el cumplimiento del objetivo específico número 4 “Construir un prototipo de software y hardware que implemente el modelo planteado”.

El objetivo del Sprint 9 es realizar una aplicación móvil Android para el usuario, con la cual pueda tener acceso a la gestión del sistema de control implementada en la vivienda, de la misma manera para concluir se implementan los sensores utilizados en una maqueta que representa la vivienda.

Las tareas que realizan son las siguientes:

Tarea 1: Instalar bibliotecas al IDE de Android ADT-Eclipse

La tarea consiste en instalar bibliotecas que permitan tener en nuestro IDE, en este caso ADT-Eclipse Android para el diseño del agente usuario. Las bibliotecas con las cuales se trabaja son: chatOntology.jar y JadeAndroid.jar, estas bibliotecas colaboran de gran manera para la implementación del agente Usuario.

Tarea 2: Construir una maqueta que represente a la vivienda e implementar el sistema de control de seguridad.

3.5 PRUEBAS

Para la elaboración de esta sección se observa la Figura 3.25



Figura 3.25: Implementación del control de seguridad en la representación de una vivienda

Mediante la manipulación de aplicación, que realiza las tareas que requiera el usuario, se observa que son funcionales en el prototipo de vivienda que esta implementada con componentes seleccionados.

Mediante el cuestionario 2 se examina si el sistema cubre las necesidades de funcionamiento, acorde a las especificaciones de diseño. Está relacionado con lo que hace la aplicación y si cumple los requisitos funcionales. La tabla 3.30 muestra que se cumplieron con todos los requerimientos funcionales propuestos y también se explica cómo se realizo.

Tabla 3.30: Cumplimiento de Requerimientos Funcionales
Fuente: Basado en (Cuera, 2014)

Requerimientos	Se cumplió con el Requerimiento		Resultado	Solución
	Si	No		
La aplicación debe realizar el registro y acceso de usuarios a la aplicación	✓		Correcto	Se crearon las respectivas tablas, relaciones y funciones en la Base de Datos.
El usuario debe gestionar la seguridad de su vivienda mediante envió de peticiones de información	✓		Correcto	Se logro diseñar un modelo de adquisición de datos que controla el ambiente de la vivienda.
Debe realizar actividades desde su aplicación Ej: apagar luces.	✓		Correcto	Gracias al modelo de adquisición de datos también se puede enviar instrucciones a los actuadores situados en la vivienda.
Debe realizar la comunicación entre usuarios y el sistema	✓		Correcto	Se integró con el agente móvil para la creación de alertas, dicha comunicación se realiza mediante agente control y agente usuario
Debe realizar la generación de un Historial de actividades.	✓		Correcto	Se visualiza las actividades realizadas por el usuario
Debe visualizar a los diferentes usuarios a	✓		Correcto	Con el modelo de envió de mensajes mediante agentes

cargo de la gestión de la vivienda				móviles se puede visualizar quien participa en la gestión de la vivienda.
------------------------------------	--	--	--	---

3.6 Prueba de Hipótesis

Para demostrar la validez de la hipótesis se realiza la prueba de hipótesis para comprobar de forma estadística se acepta o se rechaza. Existen varias formas de sacar la prueba de hipótesis ya sea en forma abstracta o directa, en esta sección se siguen nueve pasos.

Primer paso: Determinar la hipótesis nula

H_0 = El modelo informático integrado conformado por elementos computacionales adecuados de hardware y software, no permite un control eficiente de la gestión de seguridad de la vivienda a través de una aplicación móvil

Segundo paso: Determinar la hipótesis alternativa

H_a = El modelo informático integrado conformado por elementos computacionales adecuados de hardware y software, permite un control eficiente de la gestión de seguridad de la vivienda a través de una aplicación móvil

Tercer paso: Determinar el nivel de significancia, en este caso será: $\alpha = 0,05$ con un margen de confiabilidad de 95%.

Cuarto paso: Determinar el estadístico de prueba con la formula con la que se realizara el resultado.

$$x_{calc}^2 = \sum \frac{(f_o - f_e)^2}{f_e}$$

f_o = Frecuencia del valor observado

f_e = Frecuencia del valor esperado

Quinto paso: Determinar tabla de valores esperados y observados.

Tabla de valores observados

Descripción	Grupo de control	Grupo de estudio experimental
Están de acuerdo	3	1
Neutrales	5	2
No están de acuerdo	2	7

Tabla de valores esperados

Descripción	Grupo de control	Grupo de estudio experimental
Están de acuerdo	2	3
Neutrales	5	5
No están de acuerdo	3	2

Sexto paso: Calcular Chi-cuadrado

$$x_{calc}^2 = \frac{(3-2)^2}{2} + \frac{(1-3)^2}{3} + \frac{(5-5)^2}{5} + \frac{(2-5)^2}{5} + \frac{(2-3)^2}{3} + \frac{(7-2)^2}{2}$$

$$x_{calc}^2 = 12,90$$

Séptimo paso: Determinar Chi-cuadrado crítica

Para calcular el grado de libertad (v) se realiza:

$$v = (\text{cantidad de filas} - 1)(\text{cantidad de columnas} - 1)$$

$$= (3-1)(2-1)$$

$$= (2)(1) = 2$$

Valor crítico=5,99

Octavo paso: Comparar valores de chi-cuadrado calculada y chi-cuadrado crítico.

$$x_{calc}^2 \leq \text{valor crítico}$$

$$12,90 \leq 5,99$$

No cumple con la condición, por tanto se rechaza la hipótesis nula

Noveno paso: Conclusiones

- Entonces se rechaza la hipótesis nula “El modelo informático integrado conformado por elementos computacionales adecuados de hardware y software, **no** permite un control eficiente de la gestión de seguridad de la vivienda a través de una aplicación móvil”
- Se acepta la hipótesis alternativa “El modelo informático integrado conformado por elementos computacionales adecuados de hardware y software, permite un control eficiente de la gestión de seguridad de la vivienda a través de una aplicación móvil”





CAPITULO V
CONCLUSIONES Y
RECOMENDACIONES



4. CONCLUSIONES Y RECOMENDACIONES

Resumen

En el presente capítulo se muestra las conclusiones de la investigación realizada además de las recomendaciones y estado de los objetivos e hipótesis.

4.1. Conclusiones

El trabajo de investigación realiza una integración de diferentes tecnologías como ser: circuitos con la placa Arduino UNO, modelo de sistemas de control con Simulink, agentes móviles y base de datos PostgreSQL.

Gracias a la integración e interacción de las tecnologías mencionadas en el anterior párrafo, se logra realizar un sistema de control inteligente de seguridad de una vivienda, la cual será de provecho para el usuario que desee realizar un seguimiento al estado actual en el que se encuentra su vivienda, a través de un dispositivo móvil que podría ser su celular.

El trabajo de investigación es un aporte a la investigación de tecnologías que llevan a vivir en un entorno más confortable para el ser humano, es un reto y motivación a la inteligencia ambiental que es el concepto de futuro no lejano.

4.2 Estado de los Objetivos

4.2.1 Estado del Objetivo General

Se cumplió con el objetivo general de “Indagar los elementos computacionales que permiten contar con un modelo informático integrado, que sea capaz de realizar el control inteligente de la seguridad en viviendas controlada a través de un dispositivo móvil”,

4.2.2 Estado de los Objetivo Específicos

El objetivo 1: “Buscar y seleccionar elementos computacionales de hardware y software para el diseño del prototipo de un sistema de control de vivienda” se cumplió en su totalidad observando las Tablas 3.7 y 3.8

El objetivo 2:” Diseñar un modelo de adquisición de datos de una vivienda utilizando los elementos hardware seleccionados.” se cumplió su construcción como se muestra en la Figura 3.7

El objetivo 3: “Diseñar un modelo de software mediado por un agente móvil que procese la información obtenida del modelo de hardware” se cumplió su construcción como se muestra en la Figura 3.8

El objetivo 4: “Construir un prototipo de software y hardware que implemente el modelo planteado”, se cumplió el objetivo como se muestra en la Figura 3.23 del capítulo 3 punto 3.5

El objetivo 5: “Evaluar el prototipo”, se cumplió el objetivo como se muestra en la Tabla 3.30 del capítulo 3 punto 3.5

4.3 Estado de la Hipótesis

La hipótesis del presente trabajo sostiene que “El modelo informático integrado conformado por elementos computacionales adecuados de hardware y software, permite un control eficiente de la gestión de seguridad de la vivienda a través de una aplicación móvil”. Con la construcción e implementación del prototipo, el sistema realiza en control de seguridad en viviendas mediante una aplicación, la hipótesis planteada se ha demostrado, como se observa capítulo 3 punto 3.6.

4.4. Recomendaciones

El trabajo presentado realiza el control del ambiente mediante tres elementos, sensor de luz, sensor de temperatura, sensor de movimiento, se recomienda ampliar estos elementos con cámaras, es decir, poder captar audio y video y enviarlos al usuario.

De la misma manera realizar la prueba de Raspberry Pi, un pequeño ordenador capaz de correr Linux y realizar tareas multimedia, se le puede adjuntar una cámara y tiene soporte de Simulink.

En relación a la aplicación Android, la siguiente mejora podría ser la implementación de una interfaz gráfica para la visión de imágenes, si se amplían los elementos como se recomendó anteriormente.



BIBLIOGRAFÍA

BIBLIOGRAFÍA

(Advisory, 2003)

Advisory Group. (2003). Ambient Intelligence: from vision to reality. EU Publication.

(Alamán, 1992)

Alamán X. (1992). Introducción a la I.A. para la asistencia al control en Tiempo Real. En el libro “La Inteligencia Artificial y el control en Tiempo Real, Ed. Repsol Colección Ensayo.

(CEDOM, 2008)

Asociación Española de Domótica CEDOM. (2008). Domotica

(Belloni y Campo, 2000)

Belloni E. y Campo, M. (2000). Una arquitectura de software para soportar agentes móviles inteligentes. Universidad Nacional del Centro de la Provincia de Buenos Aires.

(Benito, 2014)

Benito T. (2014). Lógica Borrosa. Universidad Carlos III

(Baez, Borrego, Cordero, Cruz y Gonzales, 2005)

Baez M., Borrego A., Cordero J., Cruz L., Gonzales M. (2005) Grupo y Tecnología UCM G-TeC. Introducción a Android

(Capart y Silvana, 2006)

Capart, C. y Silvana, M. (2006). Tesina Agentes de software móviles: un paradigma alternativo para diseñar sistemas distribuidos. Universidad Nacional de la Patagonia San Juan Boscon.

(Egido, 2009)

Egido, R. (2009). Instalación Domótica De Una Vivienda Unifamiliar Con El Sistema EIB.

(MORI, 2007)

Equipos MORI Consultores Asociados. (2007). Resumen Ejecutivo Informe de Estratificación social a escala nacional por nivel socioeconómico 2006-2007.

(Enríquez, 2009)

Enríquez R. (2009), Universidad de Córdoba, Guía de Usuario de Arduino.

(Galán, Jiménez, Sanz y Matía, 2000)

Galán R., Jiménez A., Sanz R., Matía F., (2000), Centro Departamento de Automática, Ingeniería Electrónica e Informática Industrial Universidad Politécnica de Madrid, Control Inteligente.

(Galicia, 2014)

Galicia A., Escola Tècnica Superior d'Enginyeria (ETSE) Universitat Rovira i Virgili (URV), Agentes al servicio del turista mediante telefonía móvil.

(Goldberg, 1989)

Goldberg, D.E.(1989), Genetics Algorithms in Search, Optimization and Machine Learning. Addison-Wesley. Mass. 1989.

(Hernan y Bordignon, 1999)

Hernan, G y Bordignon, F. (1999) Revisión: tecnología de agentes de software.

(Hernández, Fernández y Baptista, 1995)

Hernández R., Fernández C. Baptista P. (1995). “Metodología de la Investigación”. México D.F.

(ISA, 2013)

Ingenieria de Sistemas y Automática. (2013). Simulación de sistemas de control continuos con Matlab y Simulink. Universidad de Oviedo.

(Lestel, 2001)

Lestel, D (2001): Les Origines Animales de la Culture. Flammarion. Paris.

(Miranda, Villacrés y Villamar, 2009)

Miranda G. Villacrés R. Villamar F. (2009).Diseño De Un Sistema Domótico Aplicado A Una Clínica De Hemodiálisis

(MOPSV, 2013)

Ministerio de Obras Publicas, Servicios y Vivienda Vice ministerio de vivienda y urbanismo de Bolivia (MOPSV). (2013). Normas Técnicas De Vivienda.

(Morrillo, 2009)

Morrillo C. (2009). Diseño del Sistema de Automatización para un Edificio inteligente.

(Mamdani, 1981)

Mamdani H. Gaines E. (1981). Fuzzy Reasoning and its Applications. Academic Press.

(Galán, Jiménez, Sanz y Matía, 1992)

Matía F., Jiménez A., Galán R., Sanz R., Fuzzy (1992) Controllers: Lifting the linear – nonlinear frontier. Fuzzy Sets and Systems. Vol. 52, N° 2 Dec.

(Management, 2007)

Management 2007 El Método Scrum.

(Pallardó, 2010)

Pallardó L. (2010). Interconexión de sistemas distribuidos. Universidad Politécnica de Valencia.

(Pressman, 2005)

Pressman R. (2005) Ingeniería del software un enfoque practico. Consultora editorial área de informática y computación. México.

(Russell y Norvig 2004)

Russell S. Norvig P. (2004) Inteligencia Artificial un enfoque moderno. Pearson Prentice Hall. ISBN: 84-205-4003-X. Madrid.

(Valle, 2012)

Valle, G. (2012). Sistema Domótico Con Tecnología Eibkonnex Para La Automatización De Servicios, Confort Y Seguridad En La Empresa Sisteldata S.A. Ecuador.

(Vázquez y Ipiña, 2012)

Vázquez Ipiña D. (2012). Inteligencia Ambiental: presencia invisible. Facultad de Ingeniería. Universidad de Deusto.



REFERENCIAS DE INTERNET

(Alegsa, 2013)

Alegsa. (2013). Sistema Integrado. Recuperado de:
<http://www.alegsa.com.ar/Dic/sistema%20integrado.php> [Acceso noviembre 2013].

(Arduino, 2014)

Arduino. (2014). ¿Cuántos tipos diferentes de Arduino hay? Recuperado de:
<http://hacedores.com/cuantos-tipos-diferentes-de-arduino-hay/> [Acceso septiembre 2014].

(Arcgis, 2013)

Arcgis. ¿Qué es una aplicación móvil? Recuperado de:
<http://resources.arcgis.com/es/help/arcgisonline/index.html#/010q0000001m000000> [Acceso diciembre 2013]

(BCB, 2014)

Banco Central de Bolivia (2014) Cotización y tipos de cambio. Recuperado de:
<http://www.bcb.gob.bo/?q=indicadores/cotizaciones/euro> [Acceso octubre 2014]

(CESOP, 2013)

Centro de Estudios Sociales y de Opinión Pública. "Definición", en Vivienda. Recuperado de
www.diputados.gob.mx/cesop/ [Acceso diciembre 2013]

(Ciencia V, 2013)

Ciencia Virtual. Estrategias para el uso de Sensores. Recuperado de:
<http://www.prepa9.unam.mx/academia./cienciavirtual/sensores.htm> [Acceso diciembre 2013]

(Conceptomovil, 2013)

Concepto móvil 2013. Aplicaciones móviles. Recuperado de:
<http://www.conceptomovil.com/aplicaciones-moviles.html> [Acceso diciembre 2013]

(Domintell, 2013)

Domintell. ¿Cuánto cuesta la instalación domótica de una vivienda?. 2012 – 2013.
España S.L. Recuperado de: <http://www.domintell.es/presupuesto> [Acceso diciembre 2013]

(ElectroniLab, 2015)

ElectroniLab. 2015 Bolivia Recuperado de:
<http://electronilab.co/tienda/sensor-de-movimiento-pir-hc-sr501/> [Acceso marzo 2015]

(Guadalinfo, 2011)

Guadalinfo.es. (2011). Tipos de aplicaciones móviles. Recuperado de:
http://www.edukanda.es/mediatecaweb/data/zip/1164/page_07.htm [Acceso diciembre 2013]

(Lopez, 2012)

Lopez J. (2012) Adquisición de datos con Arduino y Matlab Recuperado de: <https://sites.google.com/a/uniboyaca.edu.co/ingjulianvelasco/adquision-de-datos> [Acceso mayo 2014]

(Maquinariapro, 2014)

Maquinariapro (2014) Tecnología. Recuperado de: <http://www.maquinariapro.com/tecnologia> [Acceso junio 2014]

(Oporto, 2012)

Oporto, H. (2012). Inseguridad Ciudadana y Criminalidad en Bolivia. Recuperado de: <http://henryoporto.blogspot.com/2013/04/inseguridad-ciudadana-y-criminalidad-en.html> [Acceso octubre 2013]

(Tilab, 2014)

Tilab (2014) Jade <http://JADE.cselt.it>





ANEXOS



ANEXO A
ÁRBOL DE PROBLEMAS

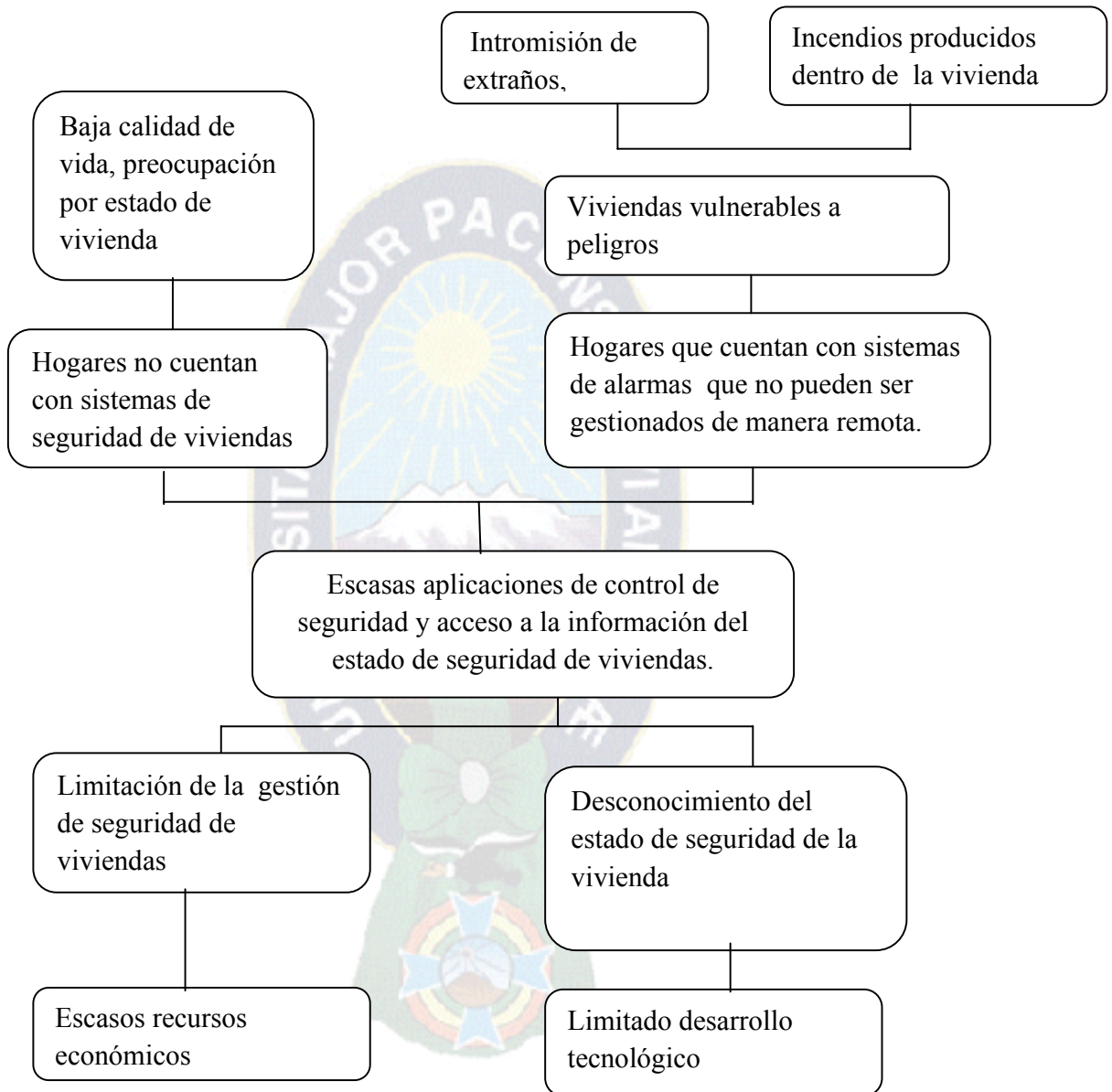


Figura A1: Árbol de Problemas
Fuente: Elaboración propia

ANEXO B

ÁRBOL DE OBJETIVOS

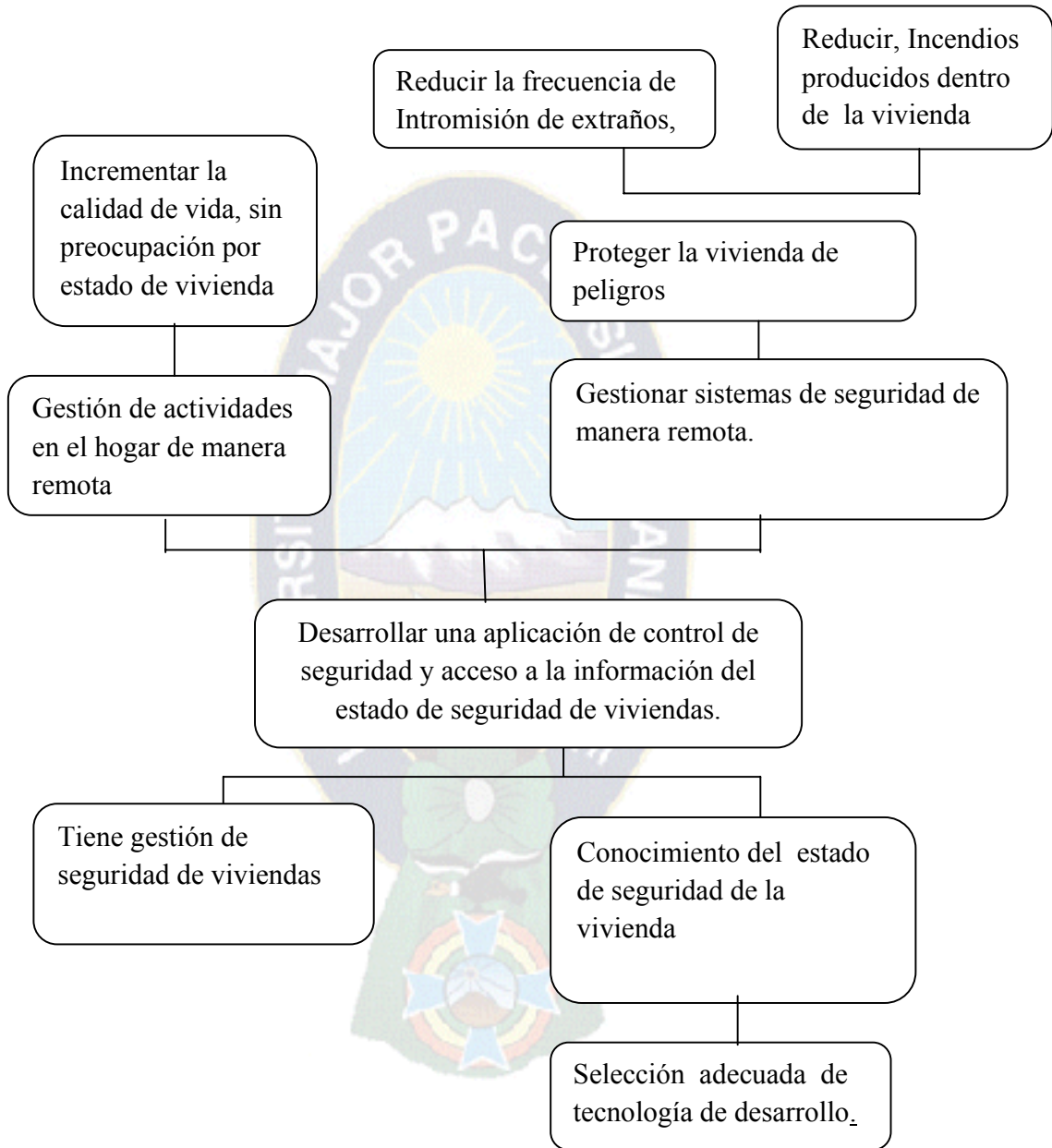


Figura A.2: Árbol de Objetivos
Fuente: Elaboración propia

ANEXO C

CUESTIONARIO 1: APLICADA A USUARIOS

NECESIDAD DE LA APLICACIÓN

Gracias por responder a este cuestionario. No tardará más de 3 minutos en completarla será de gran ayuda para conocer las necesidades de contar con un sistema de Seguridad en la vivienda.

A continuación, presentamos diversas opciones, por favor seleccione su respuesta con una X tomando en cuenta siguiente escala:

1	2	3	4	5
Nada	Poco	No se	Mucho	Demasiado

Edad _____

Sexo _____

1. ¿Cree usted que hoy en día existe mucha delincuencia?

1 2 3 4 5

2. ¿Ha sido sujeto de robos en su vivienda?

1 2 3 4 5

3. ¿Cuenta con algún sistema de seguridad en su vivienda?

1 2 3 4 5

4. ¿Le gustaría contar con un sistema de seguridad en su vivienda?

1 2 3 4 5

5. ¿Le gustaría que el sistema de seguridad sea controlado mediante una aplicación móvil en su celular?

1 2 3 4 5

6. ¿Estaría dispuesto a pagar por la aplicación que realice la gestión de seguridad de su vivienda?

Si

No

ANEXO D

CUESTIONARIO 2: APLICADA A USUARIOS

CALIDAD Y USABILIDAD DE APLICACIÓN

Gracias por responder a este cuestionario. No tardará más de 3 minutos en completarla será de gran ayuda para conocer el nivel de calidad y uso de la aplicación. Los datos que en ella se consignen se tratarán de forma anónima.

A continuación, presentamos diversas opciones, por favor seleccione su respuesta con una X tomando en cuenta siguiente escala:

1	2	3	4	5
Muy en desacuerdo	Algo en desacuerdo	Ni en acuerdo, ni en desacuerdo	Algo de acuerdo	Muy de acuerdo

Edad _____

Sexo _____

1. ¿ Te pareció que la aplicación es fácil de comprender y usar?

1 2 3 4 5

2. ¿Consideras la interfaz gráfica de usuario amigable?

1 2 3 4 5

3. ¿Consideras que la información brindada por la aplicación es ideal?

1 2 3 4 5

4. ¿Crees es importante tener la información brindada por la aplicación?

1 2 3 4 5

5. ¿Conocías alguna aplicación que realice el control de una vivienda?

1 2 3 4 5

6. ¿Crees que la aplicación favorecerá a la sociedad?

Si

No



DOCUMENTACIÓN

