



INSTITUTO POLITÉCNICO NACIONAL

**CENTRO DE INVESTIGACIÓN Y DESARROLLO
DE TECNOLOGÍA DIGITAL**



MAESTRIA EN CIENCIAS EN SISTEMAS DIGITALES

**“SISTEMA ADMINISTRADOR DE SERVICIOS PARA SMART HOME,
ESCALABLE Y CONFIGURABLE POR EL USUARIO”**

TESIS

QUE PARA OBTENER EL GRADO DE

MAESTRO EN CIENCIAS

P R E S E N T A:

ING. JAIME GONZÁLEZ GORDILLO

BAJO LA DIRECCIÓN DE:

**DR. JOSÉ CRUZ NÚÑEZ PÉREZ
M.C. ERNESTO EDUARDO QUIROZ MORONES**

NOVIEMBRE DE 2011

TIJUANA, B.C., MÉXICO



INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REVISIÓN DE TESIS

En la Ciudad de Tijuana, B.C. siendo las 14:00 horas del día 14 del mes de noviembre del 2011 se reunieron los miembros de la Comisión Revisora de Tesis, designada por el Colegio de Profesores de Estudios de Posgrado e Investigación de CITEDI para examinar la tesis titulada:

SISTEMA ADMINISTRADOR DE SERVICIOS PARA SMART HOME, ESCALABLE Y CONFIGURABLE POR EL USUARIO.

Presentada por el alumno:

GONZÁLEZ

Apellido paterno

GORDILLO

Apellido materno

JAIME

Nombre(s)

Con registro:

B	0	9	1	8	3	8
---	---	---	---	---	---	---

aspirante de:

MAESTRÍA EN CIENCIAS EN SISTEMAS DIGITALES

Después de intercambiar opiniones, los miembros de la Comisión manifestaron **APROBAR LA TESIS**, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

LA COMISIÓN REVISORA

Directores de tesis

M. C. ERNESTO EDUARDO QUIROZ MORONEZ

DR. JOSÉ CRUZ NÚÑEZ PÉREZ

DR. JUAN JOSE TAPIA ARMENTA



S. E. P.

INSTITUTO POLITÉCNICO NACIONAL
CENTRO DE INVESTIGACIÓN Y DESARROLLO
DE TECNOLOGÍA DIGITAL
DIRECCIÓN

M.C. ANDRÉS CALVILLO TÉLLEZ

DR. VÍCTOR HUGO DÍAZ RAMÍREZ

PRESIDENTE DEL COLEGIO DE PROFESORES

DR. LUIS ARTURO GONZÁLEZ HERNÁNDEZ



INSTITUTO POLITÉCNICO NACIONAL
SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

CARTA CESIÓN DE DERECHOS

En la Ciudad de Tijuana, Baja California el día 14 del mes Noviembre del año 2011, el (la) que suscribe Jaime González Gordillo alumno (a) del Programa de MAESTRIA EN CIENCIAS EN SISTEMAS DIGITALES con número de registro B091838, adscrito al CENTRO DE INVESTIGACION Y DESARROLLO DE TECNOLOGIA DIGITAL, manifiesta que es autor (a) intelectual del presente trabajo de Tesis bajo la dirección del Dr. José Cruz Núñez Pérez y el M.C. Ernesto Eduardo Quiroz Morones y cede los derechos del trabajo intitulado Sistema Administrador de Servicios para Smart Home, Escalable y Configurable por el Usuario, al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y/o director del trabajo. Este puede ser obtenido escribiendo a la siguiente dirección Av. del Parque No. 1310, Mesa de Otay, Tijuana, Baja California, México, C.P. 22510. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.

Jaime González Gordillo

Nombre y firma

*Las ciencias tienen las raíces amargas,
pero muy dulces los frutos.*

Aristóteles

DEDICATORIA

**A mis padres: Gabriel González Camacho
Blanca Flor Gordillo de González
y a mis hermanos
Carlos Arturo González Gordillo
Gabriel González Gordillo**

AGRADECIMIENTOS

Agradezco a Dios, por haberme guiado y dado la fortaleza durante mis estudios de la maestría.

Agradezco a mis directores de tesis al Dr. José Cruz Núñez Pérez y al M. C. Ernesto Eduardo Quiroz Morones, por el apoyo, consejo y tiempo que me brindaron en el trayecto de este trabajo de tesis.

Agradezco a mi comité tutorial conformado por: Dr. Juan Tapia Armenta, Dr. Víctor Hugo Díaz Ramírez, M. C. Andrés Calvillo Téllez, por sus consejos y críticas durante el desarrollo de mi investigación.

Agradezco al Instituto Politécnico Nacional, al Centro de Investigación y Desarrollo de Tecnología Digital y al Consejo Nacional de Ciencia y Tecnología, por haberme dado la oportunidad de formar parte del equipo de investigación del área de Tecnologías de la Información y la Comunicación.

Finalmente, agradezco profundamente a mis padres por el apoyo incondicional que me han brindado siempre, por su cariño y comprensión que me permitieron concluir este nuevo logro en mi vida.

SISTEMA ADMINISTRADOR DE SERVICIOS PARA SMART HOME, ESCALABLE Y CONFIGURABLE POR EL USUARIO

RESUMEN

El presente trabajo consiste en el diseño de un Sistema Administrador de Servicios para *Smart Home*, con la característica de ajustar todos los servicios a las demandas de cada usuario.

El sistema descrito plantea una alternativa de solución a la problemática de actualización del sistema bajo los requerimientos de cada usuario. La solución propuesta consiste en la escalabilidad y configurabilidad que brinda el sistema por medio de una herramienta de diseño CASE (por sus siglas en inglés, *Computer Aided Software Engineering*) para realizar tal objetivo. Además, con esta herramienta de diseño CASE asegura una larga vida al sistema. También, se desarrolló un estudio de cada parte que integra el sistema, con el fin de brindar un sistema robusto y confiable, que permita anexar en un futuro elementos al sistema de manera práctica y sin complejidad.

Este documento de tesis describe todas las etapas (análisis, diseño, implementación, pruebas y resultados) que se realizaron para el desarrollo del sistema planteado en este trabajo de tesis, el cual brinda fundamentos teóricos y prácticos para futuras investigaciones en el área de *Smart Home*.

La importancia de esta investigación consiste en la automatización de la etapa de mantenimiento, dando como resultado el ahorro de tiempo, reducción de costos, servicios personalizados del sistema y reutilización de código, que evita el incremento de errores en la codificación; lo anterior justifica el desarrollo del presente trabajo, el cual, pretende inducir a voltear la mirada hacia la automatización de las etapas de desarrollo de sistemas.

Palabras Claves: Sistema Administrador, *Smart Home*, Escalable, Configurable, CASE, Automatización, Reutilización de Código, Etapas de Desarrollo, Sistema.

SCALABLE AND USER CONFIGURABLE MANAGEMENT SYSTEM FOR SMART HOME SERVICES

ABSTRACT

The present work undertakes the design of a Management System for Smart Home Services, characterized by the capability of user commanded services adaptability.

The system developed focuses on an approach to the problem of system adaptation to each user's requirements. The proposed solution utilizes a CASE design tool which delivers software scalability and configurability. Additionally, the CASE design tool assures a long life to the system. Also, a study was developed of each part that comprises the system, with the purpose of offering a robust and reliable system, that allows in the future to add elements to the system in a practical and straightforward manner.

This thesis document describes all the stages (analysis, design, implementation, tests and results) that were conducted for development of the prototype Smart Home system, and offers theoretical and practical foundations for future investigations in the area of Smart Home.

The importance of this research work consists on the automation of the maintenance stage, which yields time saving, cost reduction, customized services and code reutilization, and minimization of codification errors.

Keywords: Management System, Smart Home, Scalable, Configurable, CASE, Automation, Code Reuse, Stages of Development, System.

TABLA DE CONTENIDO

LISTA DE FIGURAS.....	I
LISTA DE TABLAS.....	III
LISTA DE ACRÓNIMOS	IV
GLOSARIO DE TÉRMINOS	V
CAPÍTULO 1 INTRODUCCIÓN	1
1.1. Antecedentes	1
1.2. Concepto de <i>Smart Home</i>	2
1.3. Planteamiento del Problema	3
1.4. Justificación.....	4
1.5. Objetivos Generales y Particulares.....	5
1.6. Aportación o Contribución.....	6
1.7. Organización del Documento.....	6
CAPÍTULO 2 MARCO TEÓRICO.....	7
2.1. Proyecto <i>Oxygen</i>	7
2.2. Interfaz de Usuario de Sistema Sensible al Contexto para el Control de <i>Smart Home</i>	8
2.3. Reconocimiento Facial en Tiempo Real para Aplicaciones de <i>Smart Home</i>	9
2.4. Construcción de un Control Remoto de Supervisión de un Sistema en Red para Aplicaciones de <i>Smart Home</i>	10
2.5. Laboratorio <i>Crescent</i> : Laboratorio <i>Smart Home</i> para Estudiantes	11
2.6. <i>Health Smart Home</i> - Hacia una Herramienta Auxiliar para la Evaluación Automática de la Dependencia de los Ancianos	12
2.7. Detección de Anomalías Usando Minería de Datos en un Ambiente <i>Smart Home</i>	13
2.8. Seguimiento de las Actividades del Usuario en Entornos de <i>Smart Home</i>	14
2.9. Diseño y Evaluación de un Sistema de Sensores Inalámbricos en el Cuerpo para Vigilancia de Salud en <i>Smart Home</i>	15
2.10. Diseño de un Objeto Virtual que Representa la Interacción Hombre-Máquina para el Control de la Reproducción de Música en <i>Smart Home</i>	16
2.11. Análisis e Implementación de un Sistema de Control para <i>Smart Home</i>	17
2.12. Reconocimiento y Rastreo de Actividades en un <i>Smart Environment</i>	18
2.13. Aplicación de RF4CE Basado en la Arquitectura Inalámbrica Automática de Configuración para <i>Smart Home</i> Ubicua.....	19
2.14. Resumen del Capítulo	20
CAPÍTULO 3 DISEÑO DEL SOFTWARE.....	21
3.1. Introducción	21
3.2. Arquitectura del Sistema	21
3.3. Análisis del Software	23
3.4. Metodología del Modelado	24
3.5. Método del Modelado del Software	24
3.6. Modelo de Datos	25
3.7. Requerimientos del Sistema	26
3.7.1. <i>Vista de Casos de Uso</i>	26
3.7.2. <i>Diagrama de Casos de Uso</i>	28

3.7.3.	<i>Formatos de Casos de Uso</i>	28
3.7.4.	<i>Diagrama de Canal</i>	35
3.7.5.	<i>Paquete de Análisis</i>	38
3.7.6.	<i>Diagrama de Agregación</i>	39
3.8.	Diseño.....	40
3.9.	CASE.....	40
3.9.1.	<i>Clasificación de CASE</i>	41
3.9.2.	<i>Componentes de CASE</i>	42
3.10.	Diagramas de Diseño.....	44
3.10.1.	<i>Diagrama de Estado</i>	45
3.10.2.	<i>Diagrama de Secuencia</i>	47
3.10.3.	<i>Diagrama de Actividades</i>	51
3.10.4.	<i>Modelo de la Base de Datos Relacional</i>	53
3.11.	Resumen del Capítulo.....	55
CAPÍTULO 4 IMPLEMENTACIÓN EN HARDWARE		56
4.1.	Subsistema de Control y Monitoreo.....	56
4.1.1.	<i>Sensores del Sistema</i>	57
4.1.2.	<i>Actuadores del Sistema</i>	58
4.1.3.	<i>Controlador</i>	60
4.1.4.	<i>Red Inalámbrica en el Interior del Sistema</i>	61
4.2.	Consulta y Control Remoto.....	64
4.3.	Resumen del Capítulo.....	65
CAPÍTULO 5 PRUEBAS Y RESULTADOS.....		66
5.1.	Pruebas del Sistema.....	66
5.1.1.	<i>Pruebas y Resultados del Sistema Administrador</i>	66
5.1.2.	<i>Menú CASE</i>	68
5.1.3.	<i>Agregar Módulo</i>	68
5.1.4.	<i>Modificar Módulo</i>	71
5.1.5.	<i>Quitar Módulo</i>	74
5.1.6.	<i>Pruebas y Resultados de Monitoreo y Control</i>	75
5.1.7.	<i>Control de Luces</i>	75
5.1.8.	<i>Control de Persianas</i>	76
5.1.9.	<i>Monitoreo de Presencia</i>	76
5.1.10.	<i>Monitoreo de Humo</i>	76
5.1.11.	<i>Programación de Tareas</i>	77
5.2.	Aplicación de Instalación del Sistema.....	77
5.3.	Resumen del Capítulo.....	79
CAPÍTULO 6 CONCLUSIONES Y TRABAJOS FUTUROS		80
6.1.	Conclusiones Generales.....	80
6.2.	Comentarios y Recomendaciones para Trabajos Futuros.....	81
7.	REFERENCIAS BIBLIOGRÁFICAS	83

LISTA DE FIGURAS

Figura 1. Componentes de un <i>Smart Environment</i>	3
Figura 2. Una visión general de la infraestructura de <i>Oxígen</i>	8
Figura 3. a) Forma tradicional de control Remoto b) Control remoto con un control remoto universal.	9
Figura 4. Detección de la cara por los detectores en cascada.	10
Figura 5. Sistema de aplicación de control de <i>Smart Home</i>	11
Figura 6. Red de sensores.	13
Figura 7. Arquitectura basada en agentes para aplicaciones <i>Smart Home</i>	14
Figura 8. Un sistema de sensores inalámbricos para vigilancia de la salud.	15
Figura 9. Formato tradicional de CD de audio y su correspondiente Virtual-CD.	16
Figura 10. Topología del sistema de control para <i>Smart Home</i>	17
Figura 11. Componentes principales del sistema propuesto para el descubrimiento y seguimiento de las actividades en [24].	18
Figura 12. Arquitectura general del sistema propuesto en [25].	19
Figura 13. Elementos del sistema propuesto.	22
Figura 14. Metodología del modelado propuesto.	24
Figura 15. Casos de uso del sistema.	28
Figura 16. Diagrama de canal para la función de ajuste del sistema.	35
Figura 17. Diagrama de canal para la función servicios del sistema.	36
Figura 18. Diagrama de canal para la función programar tareas.	37
Figura 19. Paquetes de análisis del sistema.	38
Figura 20. Diagrama de agregación del software.	39
Figura 21. Diagrama de agregación del hardware del sistema.	39
Figura 22. Diagrama de estado del menú CASE.	45
Figura 23. Diagrama de estado del menú servicio.	46
Figura 24. Diagrama de secuencia para el escenario de crear interfaz.	47
Figura 25. Diagrama de secuencia para el escenario de modificar interfaz.	48
Figura 26. Diagrama de secuencia para el escenario de quitar interfaz.	48
Figura 27. Diagrama de secuencia para el escenario de control de actuadores.	49
Figura 28. Diagrama de secuencia para el escenario de monitoreo de sensores.	49
Figura 29. Diagrama de secuencia para el escenario de programación de tareas.	50
Figura 30. Diagrama de actividad para el monitoreo del puerto COM4.	51
Figura 31. Diagrama de actividad para realizar las tareas de programadas.	52
Figura 32. Diseño de la base de datos del sistema parte (a).	54
Figura 33. Diseño de la base de datos del sistema parte (b).	55
Figura 34. Imagen del sensor Sharp 2Y0A02.	57
Figura 35. Imagen del sensor MQ2.	57
Figura 36. Imagen del servomotor Futaba 3003.	58
Figura 37. Señal PWM para Futaba 3003.	59
Figura 38. Matriz de Leds.	60
Figura 39. Proceso de comunicación para realizar el monitoreo y control.	60
Figura 40. Tarjeta Arduino UNO.	61
Figura 41. Módulo XBee Serie 1.	64

Figura 42. Red <i>XBee</i> .	64
Figura 43. Aplicación para la conexión remota.	65
Figura 44. Interfaz principal diseñada.	68
Figura 45. Formulario agregar.	69
Figura 46. Creación de un módulo de servicio de luces.	70
Figura 47. Registro del panel creado.	70
Figura 48. Registros de las características del módulo.	70
Figura 49. Registros de los estados de cada elemento del módulo.	71
Figura 50. Registro del botón del módulo.	71
Figura 51. Formulario modificar.	72
Figura 52. Modificación de la interfaz.	73
Figura 53. Registro modificado.	73
Figura 54. Formulario quitar.	74
Figura 55. Mensaje de confirmación.	74
Figura 56. Registro eliminado.	75
Figura 57. Interfaz de instalación.	78

LISTA DE TABLAS

Tabla 1. Formato de caso de uso No. 1.....	29
Tabla 2. Formato de caso de uso No. 2.....	30
Tabla 3. Formato de caso de uso No. 3.....	31
Tabla 4. Formato de caso de uso No. 4.....	32
Tabla 5. Formato de caso de uso No. 5.....	33
Tabla 6. Formato de caso de uso No. 6.....	34
Tabla 7. Clasificación de las herramientas CASE.....	41
Tabla 8. Características del Sharp 2YoA02.....	57
Tabla 9. Características del MQ2.....	57
Tabla 10. Características de Futaba 3003.....	59
Tabla 11. Características de redes para interiores.....	63

LISTA DE ACRÓNIMOS

CASE	Ingeniería de Software Asistida por Computadora (<i>Computer Aided Software Engineering</i>)
DVD	Disco Versátil Digital (<i>Digital Versatile Disc</i>)
DVD	Disco versátil digital (<i>Digital Versatile Disc</i>)
GUI	Interfaz Gráfica de Usuario (<i>Graphical User Interface</i>)
IP	Protocolo de Internet (<i>Internet Protocol</i>)
JDK	Equipo de Desarrollo de Java (<i>Java Development Kit</i>)
OSI	Modelo de Interconexión de Sistemas Abiertos (<i>Open System Interconnection</i>)
PC	Computadora Personal (<i>Personal Computer</i>)
PWM	Modulación por Ancho de Pulso (<i>Pulse-Width Modulation</i>)
RF	Radio Frecuencia (<i>Radio frequency</i>)
RF4CE	Radio Frecuencia para Electrónica de Consumo (<i>Radio Frequency for Consumer Electronics</i>)
RSSI	Indicador de Intensidad de Señal Recibida (<i>Receive Signal Strength Indication</i>)
SQL	Lenguaje de Consulta Estructurado (<i>Structured Query Language</i>)
TI	Tecnología de la Información (<i>Information technology</i>)
UART	Transmisor-Receptor Asíncrono Universal (<i>Universal Asynchronous Receiver-Transmitter</i>)
UML	Lenguaje Unificado de Modelado (<i>Unified Modeling Language</i>)
USB	Bus Universal en Serie (<i>Universal Serial Bus</i>)
VCR	Grabación de Video Cassette (<i>Video Cassette Recording</i>)
Wifi	Fidelidad Inalámbrica (<i>Wireless Fidelity</i>)
WSN	Redes de Sensores Inalámbricos (<i>Wireless Sensor Network</i>)
X-10	Protocolo Estándar de Transmisión por la Línea Eléctrica

GLOSARIO DE TÉRMINOS

Broadcast	Modo de transmisión de información donde un nodo emisor envía información a una multitud de nodos receptores de manera simultánea
Domótica	Instalación de sistemas de gestión técnica automatizada en casas
Gateway	Dispositivo que permite interconectar redes de computadoras con protocolos y arquitecturas diferentes a todos los niveles de comunicación
Inmótica	Instalación de sistemas de gestión técnica automatizada en edificios
Netbooks	Ordenador portátil de bajo costo y generalmente de reducidas dimensiones
Set-top box	Dispositivo encargado de la recepción y decodificación de una señal analógica o digital
Tarjeta Arduino	Es una plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo
XBee	Módulos de transmisión inalámbricos
ZigBee	Nombre de la especificación de un conjunto de protocolos de alto nivel de comunicación inalámbrica para su utilización con radiodifusión digital de bajo consumo

Capítulo 1

Introducción

1.1. Antecedentes

La automatización de servicios del hogar se remonta a los años setenta, este servicio se basaba en la tecnología X-10 (protocolo estándar de transmisión por la línea eléctrica) que permitía el control de diversos electrodomésticos y regulación de la temperatura del ambiente. Más tarde en los años ochenta y principios de los noventa, con el auge de las computadoras personales, se empezaron a incorporar Sistemas de Cableado Estructurado (SCE) en casas y edificios, con el fin de facilitar la conexión entre sí, de diversos aparatos, esta conexión se realizaba con cableado estándar que brindaba un mejor control de los diversos servicios del hogar. En la década de los noventa, se empezó a mencionar el término de Domótica en la literatura, que define la Asociación de Inmótica y Domótica Avanzada (AIDA) como la integración en los servicios e instalaciones residenciales de toda tecnología que permita una gestión energética eficiente, remota, confortable y segura, posibilitando una comunicación entre todos ellos. Sin embargo, con la evolución tecnológica de diferentes disciplinas, como la microelectrónica, las telecomunicaciones, la informática, entre otras, ha permitido llevar la automatización de servicios del hogar a niveles superiores como el cuidado de la salud, el monitoreo de actividades para predecir accidentes, entre otros, lo que ha desembocado a un nuevo término en la literatura *Smart Home*, que se ha convertido en una línea de investigación que ha inspirado nuevos retos a los investigadores [6].

1.2. Concepto de *Smart Home*

En la actualidad no existe un acuerdo general sobre la definición de *Smart Home*, existen diferentes definiciones para este término, algunas de las cuales son:

- Es la integración de los diferentes servicios dentro de una casa mediante el uso de un sistema de comunicación común [1].
- Es una vivienda que incorpora una red de comunicaciones que conecta a los aparatos eléctricos y los servicios principales y les permite tener control y acceso a distancia [2].
- Es la combinación de servicios de la red doméstica, la automatización del hogar, y los servicios de Internet, tanto en las oficinas como dentro y fuera de la casa [3].

Por lo tanto, se puede concluir que *Smart Home* lo constituyen tres partes: una red local dentro del hogar, la automatización de los diferentes servicios y la conexión a Internet.

En un inicio, la tecnología *Smart Home* se enfocó en el control de los sistemas ambientales, como la iluminación y calefacción. Sin embargo, recientemente con el desarrollo de tecnología inteligente se pueden manipular una gran variedad de los componentes eléctricos que integran el hogar [4] tales como los aparatos electrodomésticos, cámaras de video para vigilancia, diversos sensores, etc.

La investigación de *Smart Home* juega un papel importante en la planificación de los modelos de viviendas del futuro, y cada vez más fabricantes e instituciones de investigación están trabajando en este campo, tales como MIT (*Massachusetts Institute of Technology*), Siemens, Cisco, IBM, Xerox, Microsoft, etc. [2].

Actualmente, el desarrollo de sistemas *Smart Home* se focaliza en la búsqueda de beneficios de los ocupantes del hogar o la oficina, ayudándoles con las tareas difíciles como el control de la seguridad y la salud, y la notificación a terceros en caso de emergencias [5].

Los sistemas *Smart Home* son un subconjunto de los sistemas conocidos como *Smart Environment* los cuales se pueden definir como aquellos que son capaces de adquirir y aplicar conocimientos sobre el medio ambiente y sus habitantes con el fin de mejorar la experiencia en ese ambiente [6].

Los componentes que integran el *Smart Environment* se pueden observar en la Fig. 1.

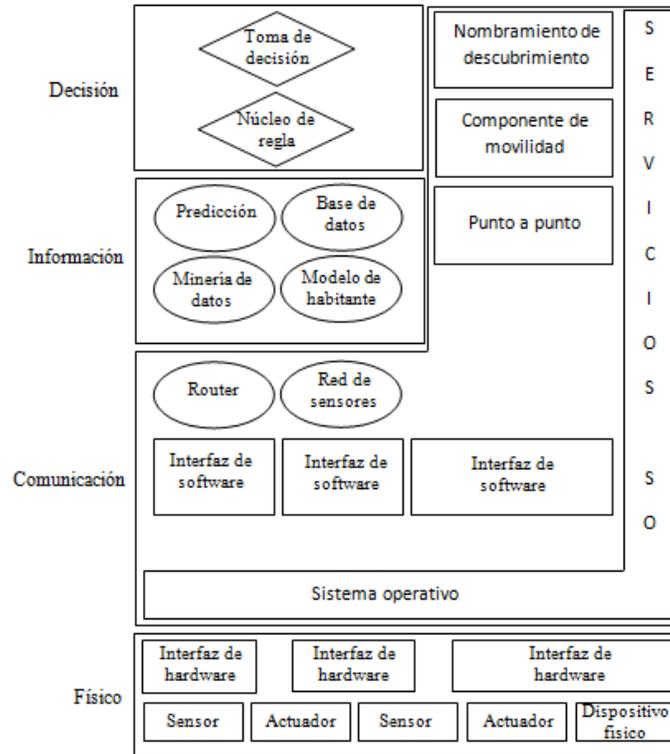


Figura 1. Componentes de un *Smart Environment*.

1.3. Planteamiento del Problema

En los últimos años se ha visto el surgimiento de varias tecnologías destinadas a la creación de entornos de vida inteligentes que sean capaces de observar e interactuar con su entorno [5].

Actualmente las tecnologías inteligentes de automatización observan su entorno e interactúan con el mismo; y están disponibles comercialmente. Sin embargo existen problemas claves para la proliferación y el avance de *Smart Homes* según [7][8], los cuales se mencionan a continuación:

- En la actualidad los consumidores no son conscientes de los beneficios de la red *Smart Home*. La mayoría de las redes domésticas se utilizan para conectar computadoras y realizar tareas como impresión y conexión compartida a Internet. Los consumidores aún no ven otros beneficios potenciales, tales como vídeo bajo demanda, comunicaciones de voz, y el control de seguridad a distancia. Debido a esta falta de conciencia, la demanda de productos de redes domésticas es aún poca.

- La necesidad de cables adicionales a través de las casas es costoso y representa una molestia para los consumidores. Para contrarrestar este problema, la industria ha desarrollado estándares inalámbricos que permiten a los usuarios interconectar dispositivos de información sin necesidad de instalar cables nuevos.
- La tecnología es demasiado compleja para la mayoría de los usuarios domésticos. A diferencia de otros equipos electrónicos del hogar, la tecnología que respalda el desarrollo de redes en el hogar no es intuitiva y requiere una mayor experiencia tecnológica. Además, con la alta gama de productos que los fabricantes están desarrollando actualmente en el mercado de *Smart Home*, la posibilidad de confusión puede ser desalentadora para las personas inexpertas.
- Posibles problemas de privacidad. Debido a que la red de la casa permite que la información fluya fuera de esta, esto puede provocar el rechazo de algunas familias debido a que su privacidad podría verse comprometida.
- Los problemas de la interfaz gráfica. Las interfaces de control no son adecuadas y la eficiencia de las mismas parece escasa, además, su entorno es muy complejo.

En este trabajo se abordan los problemas de comunicación de la red local por medio de cables, los problemas de la interfaz gráfica, complejidad de la tecnología en la configuración y la escalabilidad de los servicios, y se plantea un sistema cuyo diseño involucra estos conceptos.

1.4. Justificación

El diseño de *Smart Environment* es un reto que atrae la atención de los investigadores de diversas disciplinas, como la inteligencia artificial, la computación ubicua y móvil, la robótica, software basado en agentes, redes de sensores y la informática multimedia [6]. Además, con el surgimiento en los últimos años de tecnología destinada a *Smart Environment*, se presentan grandes oportunidades para el desarrollo de diversos proyectos que permiten mejorar la independencia y la calidad de vida de los habitantes [9].

El desarrollo de los sistemas *Smart Homes* ofrece oportunidades sin precedentes debido a la amplia variedad de aplicaciones que integran el sistema. Los sistemas *Smart Homes* proporcionan apoyo a las personas que tienen alguna discapacidad física o mental, como por ejemplo personas minusválidas requieren un sistema de control que permita manipular algún

servicio del hogar desde el lugar donde se encuentren, evitándole trayectos innecesarios; Otro ejemplo son las personas de la tercera edad, que requieren vigilancia, debido a sus limitaciones físicas, sensoriales y cognitivas [10][11], y en consecuencia ellos están propensos a accidentes frecuentes, por lo que es necesario velar por su seguridad. Así, por medio de un sistema *Smart Home* se les puede dar seguimiento y asistencia evitando posibles problemas de salud, además de que se evita que la persona salga de su casa.

Por otro lado los sistemas *Smart Homes* proporcionan una plataforma interesante para el desarrollo de aplicaciones de software adaptable y funcional, dentro de las diversas áreas de las ciencias de la computación [12]. Debido a lo anterior, es de particular interés realizar en esta tesis contribuciones en el área de los sistemas *Smart Homes* en los aspectos de automatización y software adaptable.

1.5. Objetivos Generales y Particulares

El objetivo general del presente trabajo es “Diseñar e implementar un administrador de servicios para *Smart Home* a nivel laboratorio, configurable a las particularidades de cada casa y escalable en los servicios, con control de los elementos mediante una plataforma inalámbrica basada en *ZigBee*”.

Los objetivos particulares son:

- Diseñar e implementar un software prototipo para el control y monitoreo de un sistema *Smart Home*.
- Diseñar una herramienta CASE para la escalabilidad de un sistema *Smart Home*.
- Diseñar una base de datos que permita almacenar todos los datos requeridos del sistema.
- Desarrollar programas para los controladores, sensores y actuadores que se localizan en el interior de la vivienda.
- Integrar una interfaz de control remoto del Administrador a través de internet.

1.6. Aportación o Contribución

1. Desarrollo completo de un nuevo sistema *Smart Home* dinámico, escalable y configurable a las necesidades del usuario.
2. Diseño de una herramienta CASE que permite la creación, modificación y eliminación de la interfaz gráfica principal del sistema *Smart Home*.
3. Lo anterior reflejará un proyecto original en su rubro, que contribuye en el estado del arte actual del desarrollo de sistemas *Smart Homes* usando la reutilización de código mediante la ingeniería de software asistida por computadora (CASE).

1.7. Organización del Documento

En el primer capítulo, se presentan los aspectos generales de este trabajo, una breve introducción sobre los sistemas *Smart Homes*, la problemática planteada en este trabajo de tesis, la justificación del desarrollo de esta investigación, los objetivos generales y particulares y por último la aportación de la investigación realizada.

En el segundo capítulo se presenta el estado del arte de los sistemas *Smart Homes*, lo que nos permite visualizar las diversas investigaciones realizadas en diferentes países.

En el tercer capítulo se presentan el diseño del software administrador del sistema *Smart Home*.

En el cuarto capítulo se presenta la implementación del hardware, se describen los dispositivos que se emplearon, la tecnología que se utilizó y la interacción entre todos los elementos del sistema.

En el quinto capítulo se presentan las pruebas realizadas al sistema *Smart Home* y los resultados que se obtuvieron.

En el sexto capítulo se presentan las conclusiones a las que se llegaron en base a los resultados obtenidos con el desarrollo de este trabajo de investigación. Además se hace mención de las mejoras y adiciones que se pueden efectuar a futuro para dar continuidad a los resultados de este trabajo de investigación.

Capítulo 2

Marco Teórico

Existen diversas investigaciones en el área de *Smart Homes* desarrolladas en varios países. Por lo que es de gran importancia realizar un estudio general de investigaciones recientes, con el propósito de efectuar una evaluación de las mismas para tener una perspectiva clara y concisa de los avances y problemas que existen en la actualidad. A continuación se presentan algunas de las principales investigaciones desarrolladas en los últimos años en el área de *Smart Home*.

2.1. Proyecto *Oxigen*

El artículo [13] describe un proyecto de colaboración entre las muchas actividades de investigación en el Laboratorio de Ciencias de la Computación (LCS) y el Laboratorio de Inteligencia Artificial (AIL) del Instituto de Tecnología de Massachusetts (MIT). La visión del proyecto *Oxigen* es proporcionar una gran cantidad de cómputo y comunicación al alcance de los seres humanos a través del uso de interfaces naturales de percepción del habla y la visión de los gestos humanos.

El proyecto *Oxigen* tiene el triple enfoque de dividir el espacio en tres grandes categorías (véase Fig. 2): un dispositivo de mano H21, una red avanzada N21, y un ambiente de muchos sensores E21.

El enfoque del proyecto consiste en recopilar información a través del altavoz de múltiples maneras, fusionar esta información con el sistema de información visual para reconocer una serie de gestos. La colaboración entre varios dispositivos de vigilancia simplifica muchas tareas de reconocimiento. Por ejemplo, un conjunto de micrófonos junto con una serie de cámaras se

pueden utilizar para realizar el procesamiento de la voz de todas las conversaciones que suceden en una reunión, aun cuando varias personas hablan al mismo tiempo.

Los retos en el proyecto *Oxygen* son de tipo Generalizado (debe estar en todas partes del hogar), Incrustados (las personas no deben percibirlo), Nómadas (debe permitir a los usuarios y los cálculos moverse libremente), Adaptable (debe proporcionar la flexibilidad y la espontaneidad en respuesta a los cambios en las necesidades del usuario y las condiciones de funcionamiento), Duradero (debe estar disponible todo el tiempo).

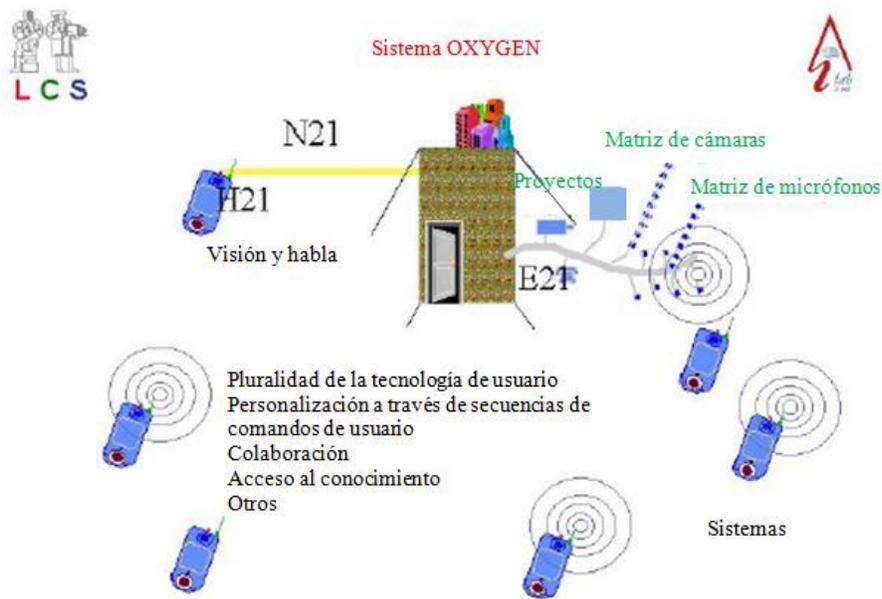


Figura 2. Una visión general de la infraestructura de *Oxígeno*.

2.2. Interfaz de Usuario de Sistema Sensible al Contexto para el Control de *Smart Home*

El artículo [14] se describe un dispositivo de mano, fácil de utilizar con el mando a distancia para controlar todos los aparatos electrodomésticos. La interfaz de usuario está diseñado lo más simple posible, sin perder la usabilidad. Con el mando a distancia, el usuario es capaz de controlar todos los aparatos centralizados en el medio ambiente, con el uso de tecnologías de RF (Radio Frecuencia) e infrarrojo (véase Fig. 3).

El artículo menciona que, en la actualidad, por lo general hay un propio control remoto para manipular todos los aparatos de infrarrojos (por ejemplo TV, DVD, VCR, amplificador,...). Cada fabricante tiene su propio estilo en sus interfaces de usuario y son diferentes a los demás

productos de otros fabricantes. Por lo tanto, si una persona puede usar un mando a distancia determinado adecuadamente, no significa necesariamente que pueda utilizar todos los mandos a distancia posible. Incluso si los aparatos son del mismo fabricante de la interfaz de usuario, los controladores remotos pueden variar mucho.

Este tipo de sistema de control es muy robusto y adaptable, donde es posible configurar el sistema en entornos muy diferentes. La interfaz de usuario de este prototipo se ha mantenido muy simple y fácil de usar, el prototipo es bastante pequeña y económico [14].

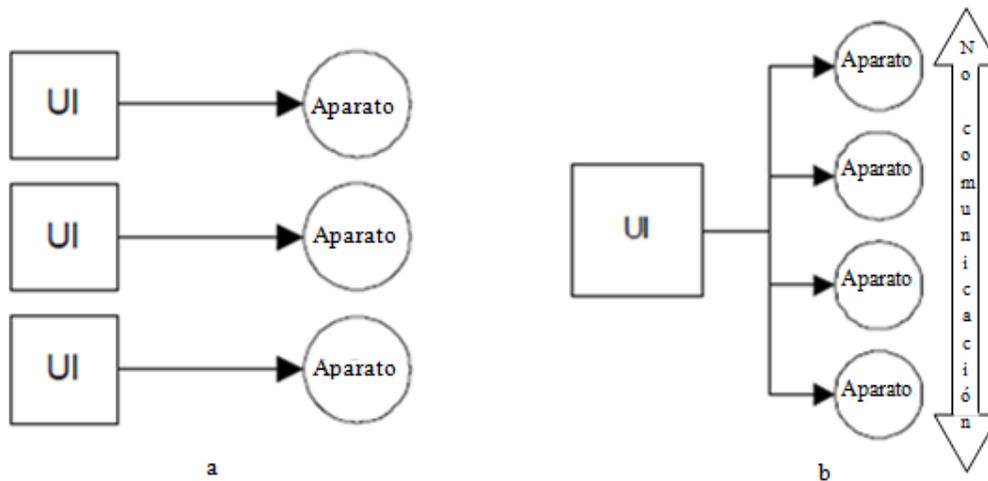


Figura 3. a) Forma tradicional de control Remoto b) Control remoto con un control remoto universal.

2.3. Reconocimiento Facial en Tiempo Real para Aplicaciones de *Smart Home*

El artículo [15] propone un sistema de reconocimiento facial en tiempo real para aplicaciones de consumo/embebidos. El sistema es embebido dentro de una conexión en el hogar y permite a los servicios inteligentes la identificación automática de los usuarios. Esto permite tener servicios personalizados, por ejemplo, un reconocimiento facial basado en televisión inteligente para seleccionar los programas favoritos del usuario (véase Fig. 4). El proceso se compone de cuatro pasos: 1) detección de rostros de vídeo en directo, 2) extracción de características faciales, 3) normalización de la cara y, 4) reconocimiento.

El sistema ofrece un buen rendimiento con respecto a los diferentes ambientes y fondo complejo, y es computacionalmente costeable. El sistema de reconocimiento facial se ha integrado en una

aplicación de una red en el hogar, donde el sistema identifica a los usuarios con una tasa de reconocimiento total de 95% y elige los programas de televisión sobre una base.

A diferencia de los sistemas profesionales de seguridad, este sistema no requiere de requisitos especiales, tales como la iluminación especial. Esto proporciona interfaces de usuario más naturales, por lo que las técnicas de reconocimiento facial encajan en la vida cotidiana.

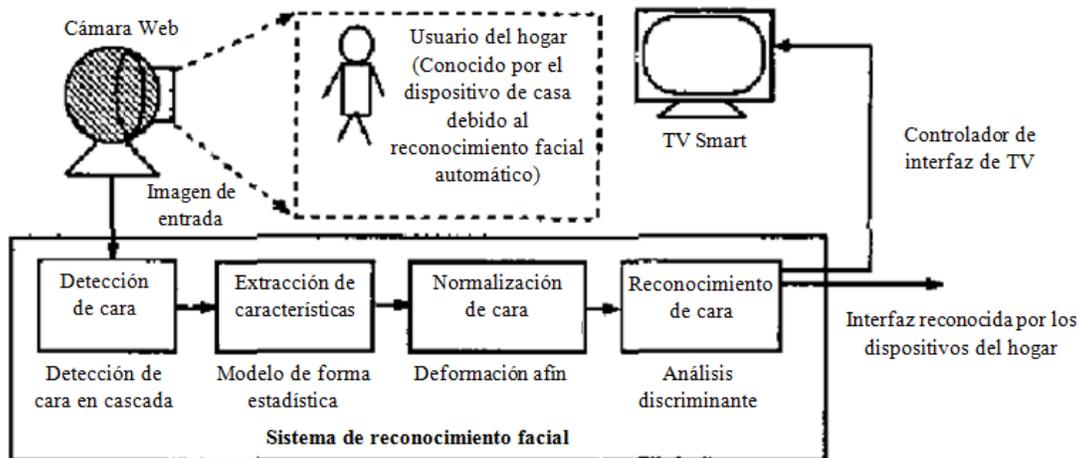


Figura 4. Detección de la cara por los detectores en cascada.

2.4. Construcción de un Control Remoto de Supervisión de un Sistema en Red para Aplicaciones de *Smart Home*

El artículo [16] propone unificar la comunicación entre diferentes dispositivos con los electrodomésticos dentro de la red *Smart Home* (véase Fig. 5). Además se unifican los protocolos de comunicación de SAANet (*Smart Appliance Alliance Net*) el cual es un protocolo de comunicación aplicada en electrodomésticos en Taiwan y el IEEE 802.15.4 para el control y monitoreo de todos los aparatos dentro del hogar. El protocolo *ZigBee* se encarga de las capas del modelo OSI (*Open Systems Interconnected*). El protocolo SAANet se necesita para satisfacer las necesidades de reconocimiento de datos. El SAANet se usa entre la comunicación de *ZigBee* y la computadora, utilizando una interfaz UART (*Universal Asynchronous Receiver/Transmitter*). Por lo tanto, el uso de SAANet y *ZigBee* cumple el requisito de simplificar el proceso de conexión entre los dispositivos finales.

Hasta ahora, la integración de SAANet y ZigBee ha sido aplicada con éxito para lograr un sistema de control y supervisión doméstico de la temperatura y humedad, y control de aparato. Así mismo, se proyectan más aplicaciones de *Smart Homes* para el futuro próximo.

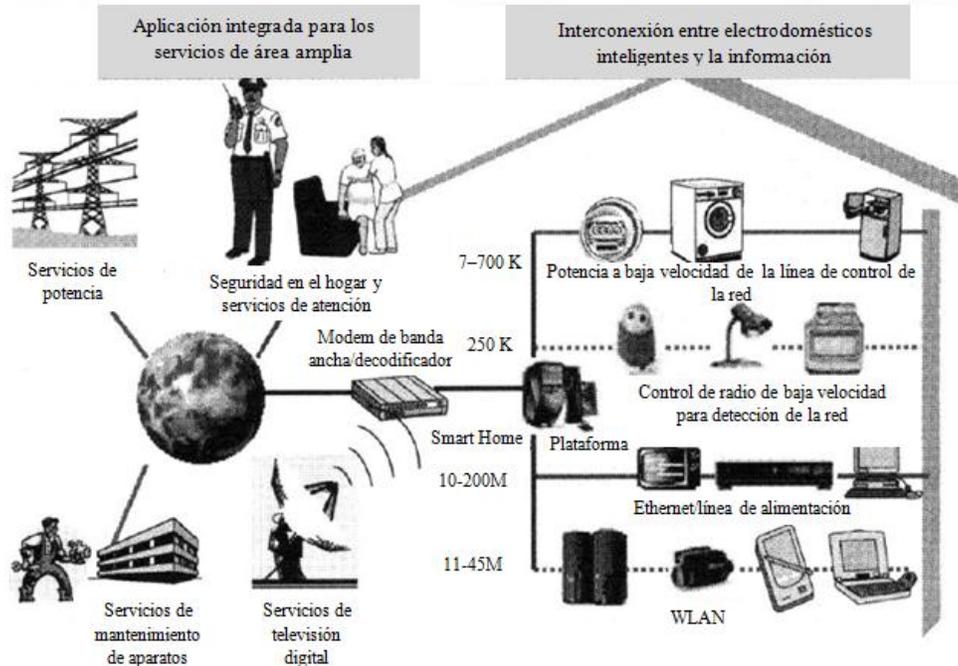


Figura 5. Sistema de aplicación de control de *Smart Home*.

2.5. Laboratorio *Crescent*: Laboratorio *Smart Home* para Estudiantes

El artículo [17] describe el laboratorio *Crescent*, el cual es un *Smart Home* que permite a los estudiantes en especial a los de licenciatura en ciencias de la computación construir sus conocimientos mediante la experiencia.

El objetivo principal de la creación de este laboratorio es desarrollar nuevos componentes que integren el laboratorio con el fin de satisfacer las necesidades reales de los clientes.

Actualmente se está trabajando en construir sistemas multi-agentes inteligentes que integren funciones específicas con el uso de la interfaz adecuada, de tal manera que con el tiempo los nuevos estándares, protocolos y los componentes se pueden añadir.

Los estudiantes y profesores han desarrollado una serie de prototipos de sistemas como la "*Smart Kitchen*", "*Smart Entertainment Controller*", "*Virtual Human Interface*" y "*Environment simulators*".

Smart Kitchen: Una herramienta para la planificación de menús, recetas y gestión de inventario que tiene en cuenta las preferencias y limitaciones del usuario.

Smart Entertainment Controller: Este sistema controla un equipo audio/visual y la iluminación. El usuario puede grabar y guardar las macros para que los ajustes preferidos del dispositivo puedan ser fácilmente registrados.

Virtual Human Interface: Uno de los propósitos generales de la interfaz del sistema es el reconocimiento del habla o la introducción de texto basado en la coincidencia de patrones simples y salidas de voz o comandos del subsistema.

Environment simulators: Dos simuladores, *SimHouse* y *ResiSim*, se han desarrollado para el dominio/análisis de requisitos y pruebas de software. *ResiSim* es un simulador visual 3-D de un ambiente familiar que permite a los desarrolladores probar el sistema con una combinación de dispositivos físicos y virtuales. *SimHouse* ofrece sensores complejos, actuadores, y simulación de control de bajo nivel que pueden utilizarse para realizar pruebas más avanzadas de control y nuevos conceptos de integración.

2.6. Health Smart Home - Hacia una Herramienta Auxiliar para la Evaluación Automática de la Dependencia de los Ancianos

En el artículo [18] se desarrolla un sistema de sensores no invasivos que permite evaluar la salud de los adultos mayores e indica si hay alguna anomalía (véase Fig. 6). Se interesa en respetar la privacidad de los usuarios y desarrollar un sistema que sus funciones sean de una manera pasiva sin necesidad de la participación de los ocupantes y tiene por objeto garantizar una vida segura y autónoma para los adultos mayores.

El artículo menciona que la salud puede ser evaluada por el grado de rendimiento de los ancianos, en las actividades de la vida diaria. Basándose en trabajos anteriores para evaluar el comportamiento de los ocupantes como el análisis de los resultados estadísticos sobre la distancia recorrida por el ocupante, la distribución de la duración de la estancia en cada habitación y los criterios para clasificar a un día como "regular" o "inusual". Se aplicó un método para detectar las desviaciones de comportamiento basado en la tasa de ocupación de las habitaciones por el ocupante.

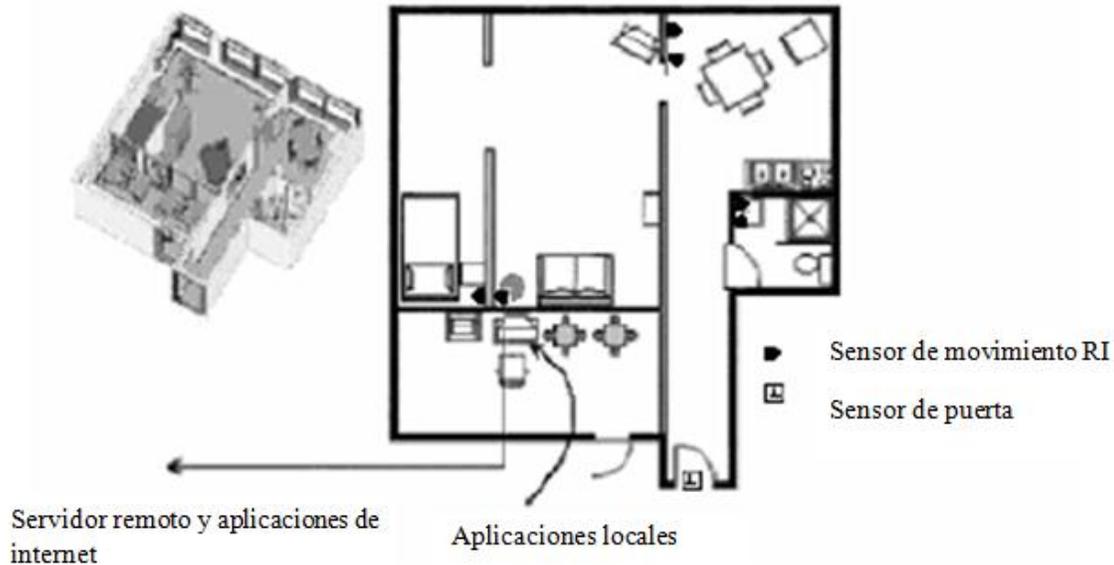


Figura 6. Red de sensores.

2.7. Detección de Anomalías Usando Minería de Datos en un Ambiente *Smart Home*

El objetivo principal del artículo [19] es realizar una investigación de diseños de algoritmos de aprendizaje automático. La hipótesis inicial consiste en que los algoritmos de aprendizaje automático pueden ser diseñados para aprender de forma automática los modelos de comportamiento residente en un *Smart Home*, y estos resultados pueden ser utilizados para realizar vigilancia de la salud y detección de anomalías de forma automática.

Estos algoritmos obtienen datos de las redes de sensores instalados en el hogar, analizan los datos obtenidos y construyen modelos de las actividades previstas. Con el modelo realizado pueden detectar anomalías en las actividades cotidianas de las personas. Los algoritmos se validan utilizando datos sintéticos y reales recogidos de las actividades de las personas en un entorno *Smart Home*.

Los resultados de los experimentos apoyan la hipótesis de que un modelo puede aprender de los datos observados en un entorno *Smart Home* y utilizarlo para informar de las anomalías producidas.

2.8. Seguimiento de las Actividades del Usuario en Entornos de *Smart Home*

El artículo [20] propone un sistema de sensores instalados en el techo de una vivienda, los cuales permiten monitorear las actividades del usuario usando la técnica RSSI (Indicador de intensidad de Señal Recibida) a través del diseño de una aplicación web que registra las actividades del sistema de sensores (véase Fig. 7). Esta aplicación analiza los datos obtenidos y emite una alerta enviando un correo electrónico si se presenta alguna anomalía en estos datos. Con la obtención de los datos anteriores permite determinar el estilo de vida de una persona, activa o sedentaria. En este trabajo se desplegaron redes de sensores inalámbricos (WSN) en un dormitorio para determinar el comportamiento de una persona según sus actividades físicas. Además las WSN se utilizaron para identificar las sillas ocupadas en una habitación, así como la movilidad de una persona. También se desplegaron sensores de humedad para saber si existe variación de la misma y determinar si hay algún cuerpo presente en la habitación. Los autores concluyen que en un futuro próximo, la lógica difusa y otras técnicas de aprendizaje automático se utilizarán para el descubrimiento de conocimiento y minería de datos; donde se proporcionarán los servicios sensibles al contexto y aplicaciones de usuario final.

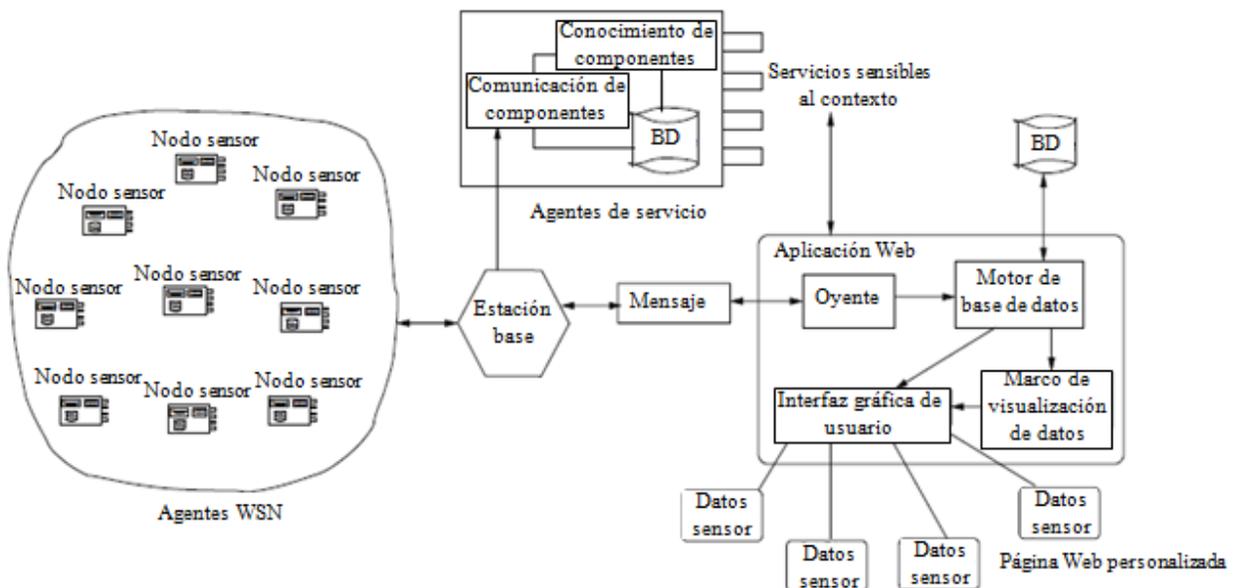


Figura 7. Arquitectura basada en agentes para aplicaciones *Smart Home*.

2.9. Diseño y Evaluación de un Sistema de Sensores Inalámbricos en el Cuerpo para Vigilancia de Salud en *Smart Home*

El artículo [21] desarrolla un sistema de sensores inalámbricos colocados en el cuerpo de una persona (véase Fig. 8), la cual permite monitorear los signos vitales fisiológicos sin salir de casa. El sistema creado es una unidad montada en la cintura, que consta de acelerómetros de tipo triaxial, los cuales utilizan un electrocardiograma para monitorear el ritmo cardíaco, sensores para monitorear la actividad muscular, sensores para monitorear la actividad eléctrica del cerebro, y un sensor de presión sanguínea para controlar la presión arterial.

Todos los datos de las muestras se transmiten usando un transmisor inalámbrico IEEE 802.15.4 hacia una computadora la cual registra todas las actividades y estas mismas son enviadas a los médicos supervisores para que verifiquen el estado de salud del paciente. Esta red de sensores se enfrenta a varios retos como el peso y tamaño de la unidad montada en la cintura, la necesidad de que la energía de la señal electromagnética de la red sea lo suficientemente baja para no perjudicar a la persona, la memoria de procesamiento de la señal, el consumo de la batería para el funcionamiento, alta fiabilidad de los datos, la seguridad de los datos.

Los autores evaluaron la factibilidad de implementar el estándar IEEE 802.15.4 en el sistema, los resultados mostraron que la transmisión 802.15.4 se ve afectada por las señales inalámbricas cercanas que ocupan la misma banda de frecuencia. Para reducir el impacto de esas señales, los transmisores-receptores permiten la opción de seleccionar un canal de operación adecuado con menos interferencias.

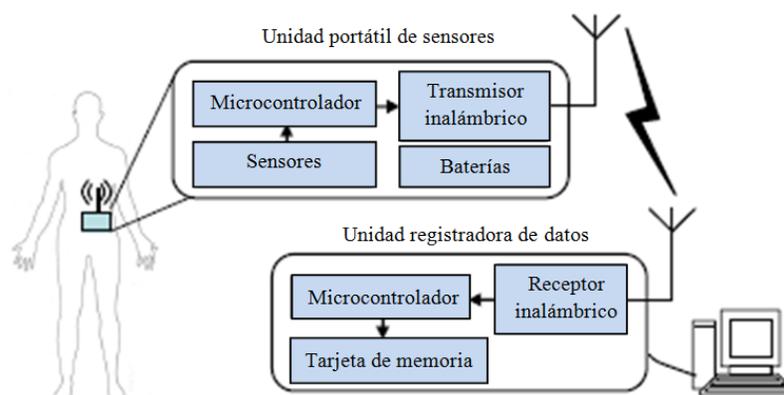


Figura 8. Un sistema de sensores inalámbricos para vigilancia de la salud.

2.10. Diseño de un Objeto Virtual que Representa la Interacción Hombre-Máquina para el Control de la Reproducción de Música en *Smart Home*

El artículo [22] parte de la hipótesis que la automatización excesiva del hogar, a veces puede perder el sentido de la realidad de los electrodomésticos o aparatos que opera el individuo.

Se menciona que los objetos virtuales en el hogar no son mobiliarios verdaderos, pero tienen las características y capacidades de los electrodomésticos o aparatos reales para proporcionar los servicios en el hogar. Es decir, los objetos virtuales conservan las características originales de funcionamiento del aparato, sin perder la utilización de alta tecnología y automatización del hogar sobre el mismo.

En este trabajo se diseñó una interfaz virtual amigable para controlar el servicio de reproducción de música en el hogar (véase Fig. 9). La interfaz objeto se llama Virtual-CD. Los residentes pueden tener un Virtual-CD, y hacer algunos gestos y movimientos que operan en la entidad, y entonces la música se reproducirá. El residente puede cantar una melodía de la época, decir el nombre de la canción, y reconocerá comandos vocales, gestos y movimientos, y preferencias de vivienda de la casa y preparar la canción.

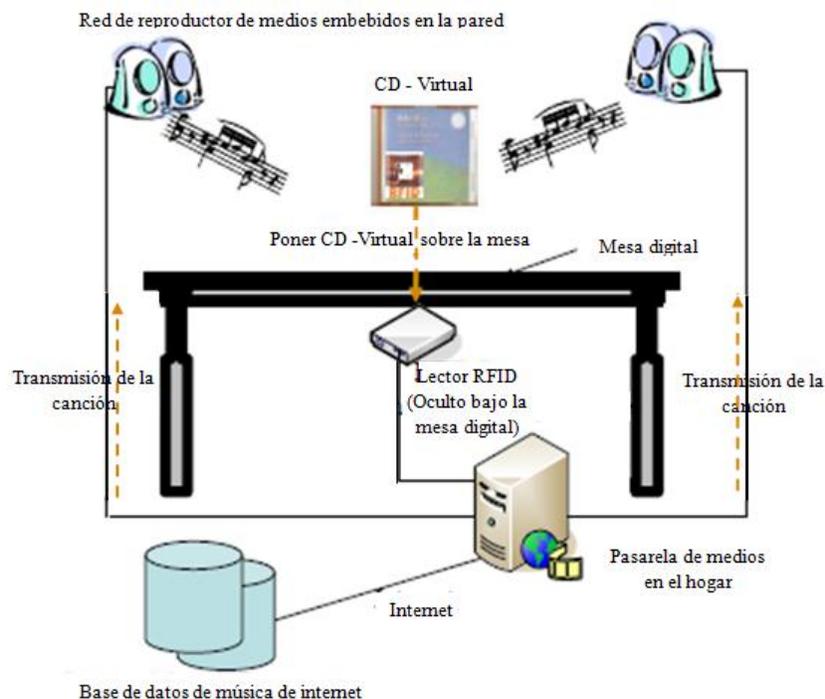


Figura 9. Formato tradicional de CD de audio y su correspondiente Virtual-CD.

2.11. Análisis e Implementación de un Sistema de Control para *Smart Home*

El artículo [23] presenta la implementación de un controlador para *Smart Home* (véase Fig. 10). El sistema consta de tres subsistemas: el controlador integrado, convertidor de señal y el dispositivo terminal. La arquitectura seleccionada permitir al usuario administrar el controlador en cualquier lugar y en cualquier momento en una terminal diferente.

El sistema soluciona el problema de incompatibilidad de los protocolos de cada aparato haciendo una envoltura de comunicación para la red de comunicación. Por otra parte también se ha diseñado un algoritmo de auto dirección de envío para configurar la dirección de los actuadores.

Para que el producto sea bien aceptado por el cliente, muchos factores se deben considerar, la facilidad de uso, el sistema debe tener una interfaz gráfica para el usuario (GUI) fácil de usar, la diversidad de los dispositivos terminales y por último que el sistema se comunique con los diferentes tipos de sensores y actuadores compatibles.

Se ha desarrollado una versión de demostración del sistema para *Smart Home*. El sistema puede controlar la iluminación y aire acondicionado en el hogar, el sensor de alarma y el control de TV.

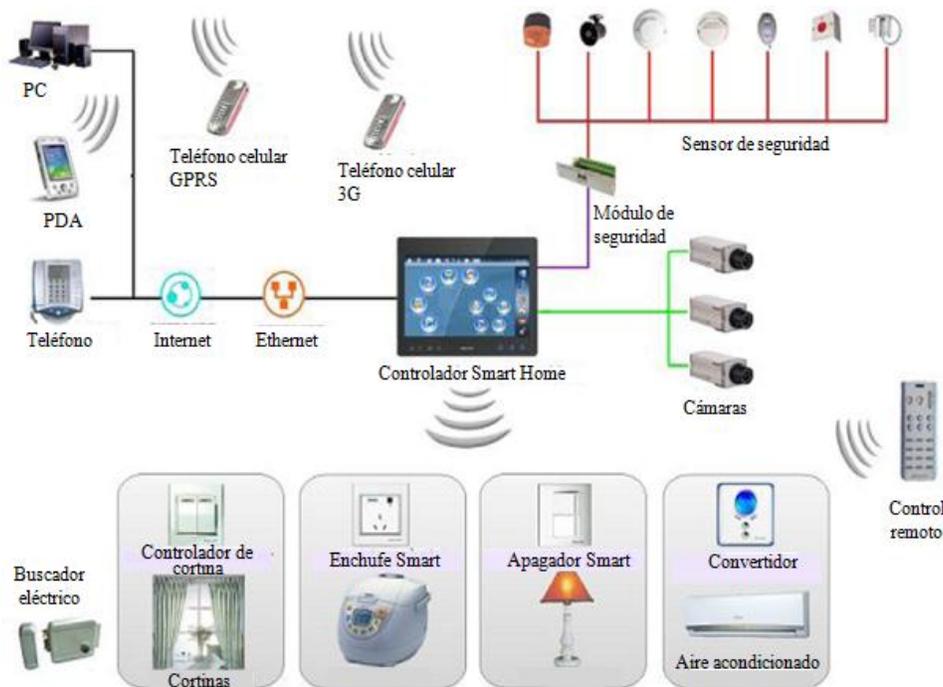


Figura 10. Topología del sistema de control para *Smart Home*.

2.12. Reconocimiento y Rastreo de Actividades en un *Smart Environment*

El artículo [24] propone un sistema que sea capaz de aprender los hábitos de los habitantes de la casa y reaccionar en función de ellos. Este sistema cuenta con una red de sensores que detectan la actividad humana la cual es registrada en una base de datos de una computadora, en ella se encuentra instalado un software que analiza las actividades registradas y emite una respuesta a través de algún actuador del sistema.

El trabajo introduce un método automatizado para el seguimiento de las actividades más frecuentes que se producen naturalmente en la rutina de una persona (véase Fig. 11). Con esta capacidad del sistema se puede monitorear la salud y detectar cambios en los patrones y el estilo de vida de la persona. El trabajo describe el enfoque de seguimiento y validación de los algoritmos mediante los datos recogidos en los entornos físicos inteligentes.

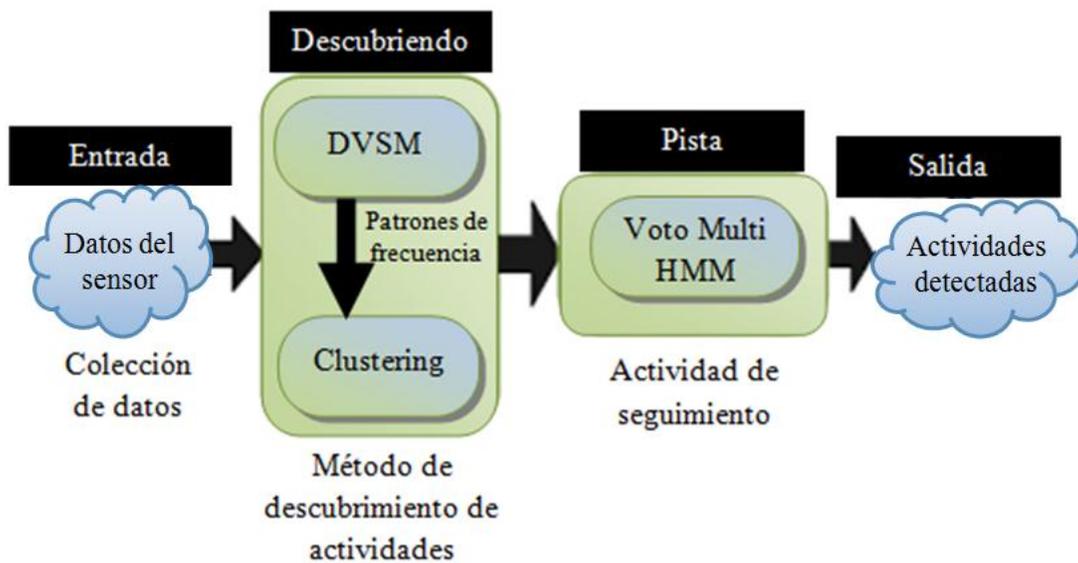


Figura 11. Componentes principales del sistema propuesto para el descubrimiento y seguimiento de las actividades en [24].

2.13. Aplicación de RF4CE Basado en la Arquitectura Inalámbrica Automática de Configuración para *Smart Home* Ubicua

El artículo [25] propone una arquitectura basada en RF4CE [por sus siglas en inglés *Radio Frequency for Consumer Electronics*], el cual es un estándar fiable, que mejora el estándar IEEE 802.15.4, proporcionando una capa de red simple y perfiles de aplicación estándar que se puede utilizar para crear una solución de interoperabilidad de múltiples proveedores para su uso en el hogar.

El uso de la interoperabilidad entre diversos productos electrónicos de consumo, por ejemplo, TV, *set-top box* (dispositivo electrónico encargado de recepción y decodificación de una señal), reproductores de DVD, portátiles y dispositivos de TI (Tecnología de la Información), tales como terminales móviles y *netbooks* es cada vez mayor. Por lo que los servicios de convergencia entre la CE (*Consumer Electronics*) y los dispositivos informáticos portátiles son indispensables, sobre todo porque tienen diferentes interfaces inalámbricas. El servicio de convergencia es imposible sin una fácil configuración de conexión de red entre los dispositivos digitales e inteligentes, por ejemplo, usuarios que comparten contenidos transparentes.

Las características de esta arquitectura son (véase Fig. 12): 1) No hay línea de vista, 2) capacidad bidireccional, 3) comunicación más rápida y confiable, 4) menor consumo de energía, 5) interoperabilidad entre productos de distintos fabricantes.

Los escenarios experimentales demuestran que la arquitectura propuesta permite la construcción de un control inteligente y proporciona eficiencia en la red doméstica inalámbrica.

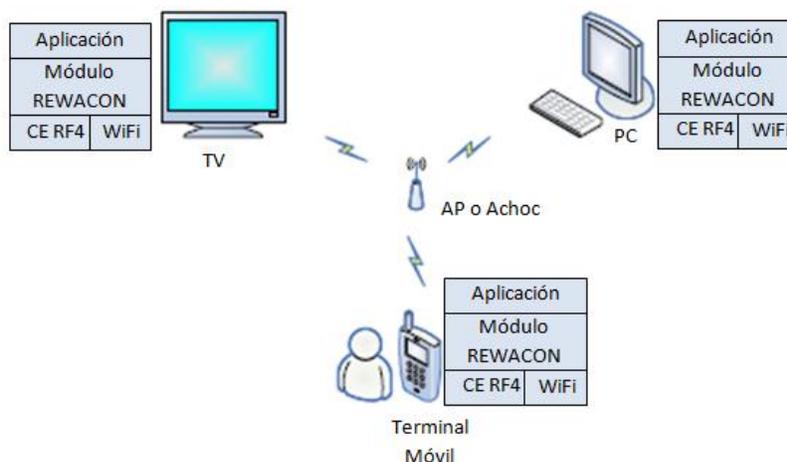


Figura 12. Arquitectura general del sistema propuesto en [25].

2.14. Resumen del Capítulo

Las investigaciones realizadas en el área de *Smart Homes* en los últimos años proporcionan alternativas para los diversos problemas existentes. Las investigaciones estudiadas en esta tesis, desarrollan proyectos de diversos temas los cuales se pueden englobar en cuatro áreas:

- Interfaz de comunicación, monitoreo y control
- Salud
- Inteligencia Artificial
- Reconocimiento de patrones de voz y video

En el área de Interfaz de comunicación se abordan los problemas de interoperabilidad entre las diferentes interfaces de comunicación, proponiéndose alternativas de unificación de protocolos de los diferentes dispositivos utilizados en el hogar. También se aborda el problema de configuración y escalabilidad del sistema desde un enfoque de los protocolos de comunicación.

En el área de la salud, se han desarrollado aplicaciones de seguimiento y asistencia de salud ayudando a las personas con discapacidades físicas o mentales, siendo las personas de la tercera edad las más beneficiadas en estos proyectos.

En el área de la inteligencia artificial, se han desarrollado algoritmos evolutivos que permiten que el sistema sea capaz de aprender en base a los datos obtenidos del sistema. Además de poder reconocer a las personas analizando sus actividades (costumbres).

En el área de reconocimiento de patrones de voz y video, se han desarrollado algoritmos capaces de reconocer las personas por medio de los gestos del rostro y del habla, dando como resultado un servicio personalizado hacia cada individuo del hogar.

Capítulo 3

Diseño del Software

3.1. Introducción

Para el desarrollo del sistema *Smart Home* propuesto en esta tesis se deben considerar varios aspectos con el objetivo de obtener los mejores resultados. Los aspectos a considerar son los siguientes:

- Lenguaje de programación
- Entorno de desarrollo
- Motor de la base de datos
- Tecnología inalámbrica
- Dispositivos electrónicos

Existe una gran variedad de opciones de cada aspecto mencionado, sin embargo realizando un análisis de las opciones se determinaron las óptimas para el proyecto. En este capítulo se explican todos los aspectos que se consideraron para el software del sistema y en el capítulo 4 se explica los aspectos relevantes del hardware del sistema.

3.2. Arquitectura del Sistema

La arquitectura de un sistema *Smart Home* especifica cómo están gobernados los diferentes elementos del sistema. Existen dos tipos de arquitecturas básicas en los sistemas *Smart Homes*: la arquitectura centralizada y la arquitectura distribuida.

La arquitectura centralizada cuenta con una topología de red tipo estrella, toda transmisión pasa por el administrador, esta evalúa y emite una respuesta si es necesario. No es posible la comunicación entre elementos.

Por otro lado, la arquitectura distribuida cuenta con una topología de red tipo bus, de manera que todos los elementos están conectados entre sí, su comunicación puede ser de manera centralizada o distribuida, la comunicación es centralizada si toda transmisión pasa por el administrador y es distribuida si los elementos pueden comunicarse entre ellos sin necesidad que toda transmisión pase por el administrador.

En el sistema desarrollado se utilizó la arquitectura centralizada, debido a que el mando recae solo en un elemento que es el sistema administrador del sistema, permitiendo el control absoluto de todos los elementos del sistema. Además con la arquitectura centralizada es más fácil agregar elementos al sistema, solo se debe actualizar el sistema administrador, y no es necesario que en los otros elementos se realice alguna modificación. El sistema administrador es el que toma las decisiones de todo el sistema *Smart Home*, por lo que es necesario el intercambio de información entre este y los elementos que integran todo el sistema. Los elementos del sistema se aprecian en la Fig. 13.

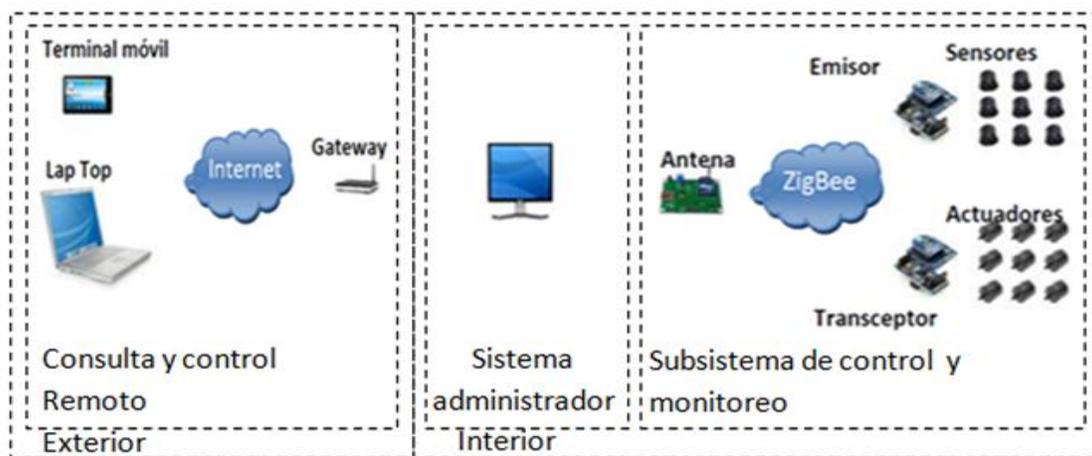


Figura 13. Elementos del sistema propuesto.

El sistema se divide en tres partes, los cuales son:

- Subsistema de control y monitoreo: En esta parte se encuentran todos los sensores y actuadores, el controlador que permite manipular las señales entrada/salida de los sensores y actuadores, los módulos de comunicación de Radio Frecuencia (RF) y una tarjeta electrónica que sirve como interfaz para la comunicación serial.

- Sistema administrador: Es el elemento más importante del sistema. Consta de una computadora personal en la que se instaló el software desarrollado para monitorear y controlar el sistema *Smart Home*.
- Consulta y control remoto: Está integrada por el *Gateway*, el protocolo de internet y el elemento que consta de una conexión a internet, este puede ser una terminal móvil o una computadora personal.

Todo el sistema está conectado de forma inalámbrica, en la parte interior por medio del protocolo IEEE 802,15.4 conocido como *ZigBee* [21] y en el exterior por medio del protocolo de internet. Toda transmisión de información tiene que procesarse en el sistema administrador y esta se encargará de responder a la transmisión recibida.

En este capítulo se aborda el diseño del sistema administrador y se menciona las principales aportaciones de esta tesis. En el capítulo 4 se explica la parte de hardware del sistema que involucra las partes subsistema de control, monitoreo, consulta y control remoto.

3.3. Análisis del Software

Para poder desarrollar el sistema administrador se requiere aplicar ingeniería de software. La ingeniería de software nos proporciona métodos para la elaboración correcta de un sistema de software. Estos métodos indican cómo desarrollar técnicamente el sistema de software deseado, considerando todos los aspectos de diseño, para garantizar un funcionamiento óptimo.

Los métodos de la ingeniería de software abarcan diferentes tareas como el análisis (ingeniería de requerimientos), diseño, desarrollo, pruebas y mantenimientos [26-28].

La primera tarea que debe efectuarse para desarrollar un sistema de software es un análisis aplicando la ingeniería de requerimientos. Los requerimientos de un sistema son la descripción de los servicios proporcionados por el sistema y sus restricciones operativas, donde la ingeniería de requerimientos pretende definir dichos requerimientos [27]. La ingeniería de requerimientos incluye dos actividades principales: la obtención de requerimientos, que da como resultado una especificación del sistema y el análisis, que da como resultado un modelo de análisis que los desarrolladores pueden interpretar sin ambigüedades [27]. De manera resumida, el proceso de la ingeniería de requerimientos es: descubrir, analizar, documentar y verificar los servicios y

restricciones [28]. En los siguientes subtemas se aborda cada proceso involucrado en el desarrollo del software.

3.4. Metodología del Modelado

Al iniciar la ingeniería de requerimientos es importante definir el enfoque o metodología a utilizar para el modelado. Existen dos metodologías (véase Fig. 14), el análisis estructurado y el análisis orientado a objetos [26][29][30].

- Metodología estructurada: la orientación de esta metodología se dirige hacia los procesos que intervienen en el sistema a desarrollar, es decir, cada función a realizar por el sistema se descompone en pequeños módulos individuales.
- Metodología orientada a objetos: a diferencia de la metodología estructurada, ésta no comprende los procesos como funciones sino que arma módulos basados en componentes, es decir, cada componente es independiente del otro. Esto nos permite que el código sea reutilizable. Es más fácil de mantener porque los cambios están localizados en cada uno de estos componentes.

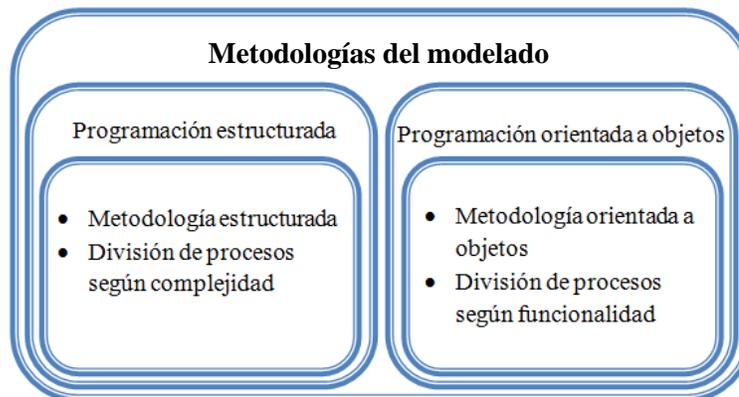


Figura 14. Metodología del modelado propuesto.

Para el desarrollo del sistema administrador se aplicará la metodología orientada a objetos debido a las ventajas que ofrece con respecto a la metodología estructurada, como la reutilización de código, la facilidad de depurar el sistema, el encapsulamiento, entre otros.

3.5. Método del Modelado del Software

Para la realización del sistema administrador se definió una metodología orientada a objetos, de esta manera el modelado de software es orientado a objetos. Existen diferentes métodos para

modelar sistemas de software con enfoque orientado a objetos, las más conocidas son las siguientes:

- Método de *Booch*
- Método de *Rumbaugh*
- Método de *Jacobson*
- Método de *Coad y Yourdon*
- Método de *Wirfs-Brock*

Los métodos anteriores son los que fueron utilizados según [26-28][39] hasta que en 1997 *Grady Booch, James Rumbaugh e Ivar Jacobson* se unieron para desarrollar un nuevo método basándose en sus métodos anteriores, hasta llegar a la creación del “UML” (por sus siglas en inglés Unified Modeling Language). En la actualidad este lenguaje es el que se adopta para modelar sistemas orientados a objetos [31], y es el que se elige para el desarrollo del sistema administrador de esta tesis.

3.6. Modelo de Datos

Un sistema de software necesita registrar todas las modificaciones que el usuario realiza en el sistema con el fin de tener un respaldo del estado del sistema, para resguardar la integridad del mismo. Esto permite que el sistema sea robusto ante fallos del sistema como un cierre inesperado, o simplemente al cerrar el sistema e iniciarlo de nuevo, no pierda su estado último anterior. Por lo tanto es indispensable realizar un análisis de los sistemas de bases de datos. Una base de datos es un conjunto de datos relacionados entre sí [32]. Bajo la estructura de una base de datos se encuentra el modelo de datos, el cual representa una colección de herramientas conceptuales para describir los datos, las relaciones, la semántica y las restricciones de consistencia [32].

Los modelos de datos más usados según [32] son:

- Modelo relacional
- Modelo de datos orientado a objetos

Otros modelos precedentes a los anteriores son el modelo de red y el modelo de datos jerárquicos. Estos modelos en la actualidad se usan muy poco [32] por lo que lo descartamos su

utilización en este proyecto. Otro modelo es el de Entidad-Relación que es muy parecido al modelo Relacional, pero este último, por su mejor comprensión, se utiliza más [32].

En primera instancia se podría definir la aplicación del modelo de datos orientado a objetos, pero recabando información existen pocos sistemas de software que interactúan con bases de datos orientado a objetos. Además existen desventajas muy fuertes en los modelos de datos orientado a objetos que no se pueden ignorar según [32] entre las que destacan:

- Carencia de un modelo de datos universal
- Carencia de experiencia
- Carencia de estándares

Como conclusión, se decide no utilizar este modelo para el proyecto a desarrollar. En cambio los sistemas relacionales se basan en un fundamento formal o teoría, denominada el modelo relacional de datos. Este último modelo es el más utilizado y aceptado a nivel internacional por lo que se ha optado en utilizarlo para el modelo de datos del sistema a desarrollar.

3.7. Requerimientos del Sistema

El modelado de requerimientos orientado a objetos abarca las siguientes vistas que nos ayudan a comprender las especificaciones del sistema a desarrollar:

- Casos de uso
- Diagrama de canal
- Paquete de análisis
- Diagrama de agregación

3.7.1. Vista de Casos de Uso

Un caso de uso es una descripción del comportamiento de un sistema. Esta descripción está basada desde el punto de vista del usuario y se plasma en un formato de casos de uso. Para conceptualizar el sistema se explica de manera general el alcance y situación actual del sistema y se desarrolla un diagrama general de casos de uso de UML siguiendo los formatos de casos de uso.

Alcance

Se realizará e implementará un Sistema Administrador de *Smart Homes* que permita el monitoreo y control de diversos servicios dentro del hogar, con la particularidad que el sistema pueda ser ajustable (configurable y escalable en los servicios) a las necesidades del usuario. Además de permitir la consulta y control del sistema por medio de una conexión remota.

El sistema administrador va a gestionar los siguientes servicios:

- Control de luces
- Control de persianas
- Monitoreo de presencia
- Monitoreo de humo

El sistema administrador será:

- Ajustable (permitirá la reconfiguración de los servicios)
- Escalable (permitirá modificar el tamaño del sistema agregando/quitando módulos del sistema)

El sistema administrador va a permitir:

- Programación de tareas (encendido/apagado de luces - abrir/cerrar las persianas)
- Alertar al usuario (mensajes en la interfaz del sistema)
- Conexión remota para la consulta y control del sistema

Situación actual

Los sistemas *Smart Homes* actuales permiten gestionar una gran variedad de servicios dentro del hogar, sin embargo no han volteado la mirada hacia la automatización de la etapa de mantenimiento que permitirá en un futuro ajustar el sistema a las necesidades del usuario sin necesidad de programar ninguna línea de código en el sistema.

3.7.2. Diagrama de Casos de Uso

Los diagramas de casos de uso son una forma de representación de casos de uso que brinda una conceptualización general de la descripción del sistema [32]. La Fig. 15 muestra los casos de uso mencionados anteriormente en el alcance del sistema.

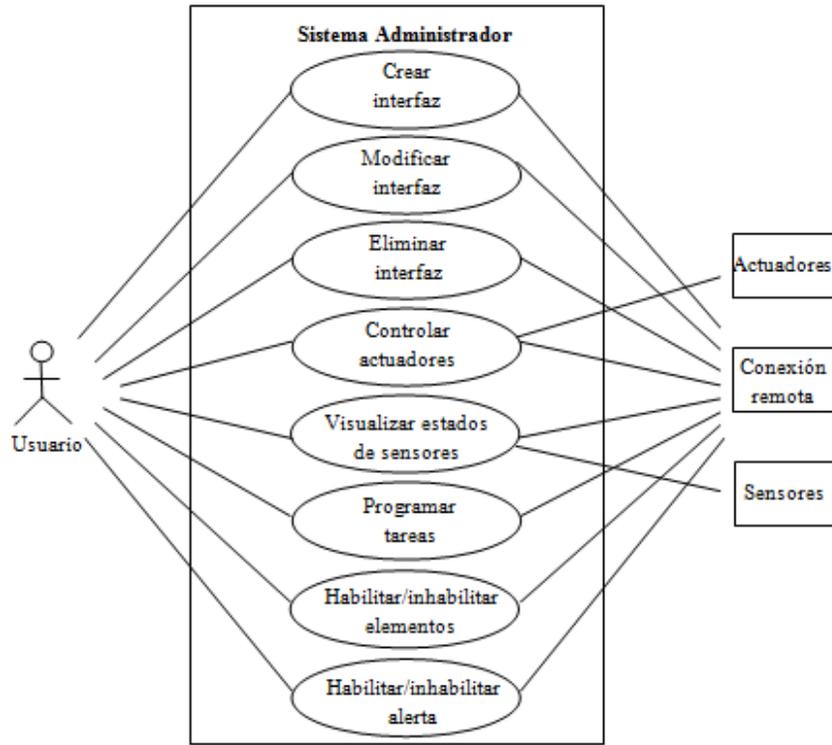


Figura 15. Casos de uso del sistema.

3.7.3. Formatos de Casos de Uso

Es la representación formal de los casos de uso [26], se especifica a detalle el proceso de las tareas más importantes del sistema en las Tablas 1-6.

Tabla 1. Formato de caso de uso No. 1.

Caso de uso:	Creación de un módulo de un servicio
Actor principal:	Usuario
Objetivo de contexto:	Tener un nuevo módulo de un servicio
Precondiciones:	Cada servicio permite la creación de seis módulos como máximo.
Disparador:	El usuario decide crear un nuevo módulo de servicio
Escenario:	<ol style="list-style-type: none"> 1 El usuario pulsa el botón agregar del menú principal 2 El sistema despliega el formulario de agregar 3 El usuario elige el servicio 4 El sistema muestra la interfaz del servicio elegido 5 El usuario ingresa los datos requeridos del sistema y pulsa el botón aceptar 6 El sistema crea nueva interfaz del servicio elegido
Excepciones:	<ol style="list-style-type: none"> 1 El número de módulos creados llegó al máximo 2 El nombre del módulo ya existe 3 El usuario cancela la creación del módulo
Prioridad:	Moderada
Cuando estará disponible:	En un lapso menor de un segundo
Frecuencia de uso:	Baja
Canal para el actor:	A través de la interfaz del sistema administrador
Actores secundarios:	Ninguno

Tabla 2. Formato de caso de uso No. 2.

Caso de uso:	Modificación de un módulo existente de un servicio
Actor principal:	Usuario
Objetivo de contexto:	Realizar cambios en un módulo existente de un servicio
Precondiciones:	Que exista un módulo creado
Disparador:	El usuario decide modificar un módulo de servicio
Escenario:	<ol style="list-style-type: none"> 1 El usuario pulsa el botón modificar del menú principal 2 El sistema despliega el formulario de modificar 3 El usuario elige el servicio 4 El sistema muestra la interfaz del servicio elegido 5 El usuario elige el módulo a modificar 6 El sistema muestra sus características del módulo elegido 7 El usuario modifica las características del módulo y pulsa el botón aceptar 8 El sistema realiza los cambios al módulo elegido
Excepciones:	<ol style="list-style-type: none"> 1 El usuario cancela la modificación del módulo
Prioridad:	Moderada
Cuando estará disponible:	En un lapso menor de un segundo
Frecuencia de uso:	Baja
Canal para el actor:	A través de la interfaz del sistema administrador
Actores secundarios:	Ninguno

Tabla 3. Formato de caso de uso No. 3.

Caso de uso:	Eliminación de un módulo existente de un servicio
Actor principal:	Usuario
Objetivo de contexto:	Eliminar un módulo existente de un servicio
Precondiciones:	Que exista un módulo creado
Disparador:	El usuario decide modificar un módulo de servicio
Escenario:	<ol style="list-style-type: none"> 1 El usuario pulsa el botón quitar del menú principal 2 El sistema despliega el formulario de quitar 3 El usuario elige el servicio 4 El sistema muestra la interfaz del servicio elegido 5 El usuario elige el módulo a quitar y pulsa el botón de aceptar 6 El sistema despliega un mensaje de verificación 7 El usuario pulsa el botón de aceptar 8 El sistema elimina el módulo elegido
Excepciones:	El usuario cancela la eliminación del módulo
Prioridad:	Moderada
Cuando estará disponible:	En un lapso menor de un segundo
Frecuencia de uso:	Baja
Canal para el actor:	A través de la interfaz del sistema administrador
Actores secundarios:	Ninguno

Tabla 4. Formato de caso de uso No. 4.

Caso de uso:	Control de un servicio
Actor principal:	Usuario
Objetivo de contexto:	Controlar un actuador del sistema
Precondiciones:	Que el actuador en el sistema esté integrado y dado de alta. Se puede controlar hasta cinco luces y cinco persianas de cada módulo del sistema.
Disparador:	El usuario decide controlar un elemento del sistema.
Escenario:	<ol style="list-style-type: none"> 1 El usuario pulsa el botón del servicio a controlar que está en el menú principal 2 El sistema despliega los módulos creados del servicio elegido 3 El usuario elige un módulo del servicio 4 El sistema muestra la interfaz del módulo elegido 5 El usuario manipula los controles de la interfaz del módulo para el control del elemento 6 El sistema envía la instrucción al elemento seleccionado 7 El sistema actualiza la base de datos
Excepciones:	El usuario no manipula los controles de la interfaz del módulo
Prioridad:	Moderada
Cuando estará disponible:	En un lapso menor de un segundo
Frecuencia de uso:	Alta
Canal para el actor:	A través de la interfaz del sistema administrador
Actores secundarios:	Ninguno

Tabla 5. Formato de caso de uso No. 5.

Caso de uso:	Monitoreo de un servicio
Actor principal:	Usuario
Objetivo de contexto:	Monitorear el estado de un sensor del sistema
Precondiciones:	Que esté integrado y dado de alta el sensor en el sistema. Se puede monitorear hasta cinco sensores de presencias y cinco sensores de humo de cada módulo del sistema.
Disparador:	El usuario decide monitorear un elemento del sistema.
Escenario:	<ol style="list-style-type: none"> 1 El usuario pulsa el botón del servicio a monitorear que está en el menú principal 2 El sistema despliega los módulos creados del servicio elegido 3 El usuario elige un módulo del servicio 4 El sistema muestra la interfaz del módulo elegido 5 El usuario visualiza los estados de los sensores del módulo elegido 6 El usuario elige si activa/desactiva la alarma 7 El sistema actualiza la base de datos
Excepciones:	El usuario activa/desactiva la alarma de la interfaz del módulo
Prioridad:	Moderada
Cuando estará disponible:	En un lapso menor de un segundo
Frecuencia de uso:	Alta
Canal para el actor:	A través de la interfaz del sistema administrador
Actores secundarios:	Ninguno

Tabla 6. Formato de caso de uso No. 6.

Caso de uso:	Programación de tarea
Actor principal:	Usuario
Objetivo de contexto:	Programar una actividad para su realización.
Precondiciones:	Que el actuador en el sistema esté integrado y dado de alta. Se puede programar las luces y persianas del sistema.
Disparador:	El usuario decide programar un elemento del sistema.
Escenario:	<ol style="list-style-type: none"> 1 El usuario pulsa el botón programar del módulo en servicio 2 El sistema despliega el formulario programar 3 El usuario ingresa los datos requeridos del sistema y pulsa el botón aceptar 4 El sistema muestra la actividad programada en la interfaz del módulo 5 El sistema registra la tarea programada en la base de datos 6 El sistema realiza la tarea programada
Excepciones:	<ol style="list-style-type: none"> 1 El usuario inserta fecha no válida 2 El usuario cancela la programación de tarea
Prioridad:	Moderada
Cuando estará disponible:	En un lapso menor de un segundo
Frecuencia de uso:	Baja
Canal para el actor:	A través de la interfaz del sistema administrador
Actores secundarios:	Ninguno

Al término de los formatos de casos de uso, se requiere analizar cada uno de ellos, por lo que en la siguiente sección se desarrollan los diagramas de canal que permiten visualizar los requerimientos desde otra óptica.

3.7.4. Diagrama de Canal

Estos diagramas permiten analizar los casos de uso visualizando el flujo de interacción dentro de un escenario específico, además de poder visualizar las responsabilidades de las clases que interactúan en la escena [26]. A continuación en las Figs. 16-18 se presentan todos los diagramas de canal del sistema correspondientes a ajustes del sistema, servicios del sistema y programación de tareas.

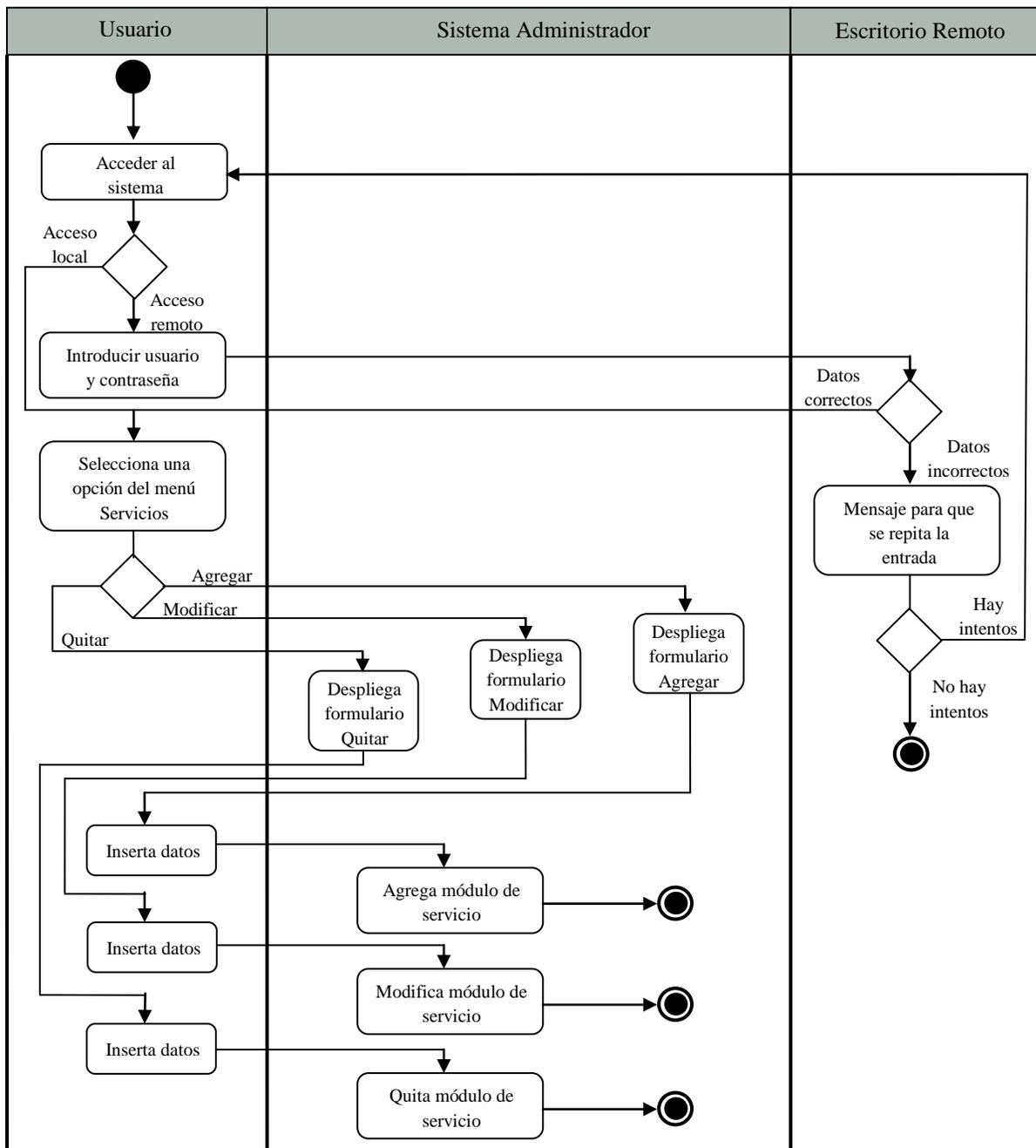


Figura 16. Diagrama de canal para la función de ajuste del sistema.

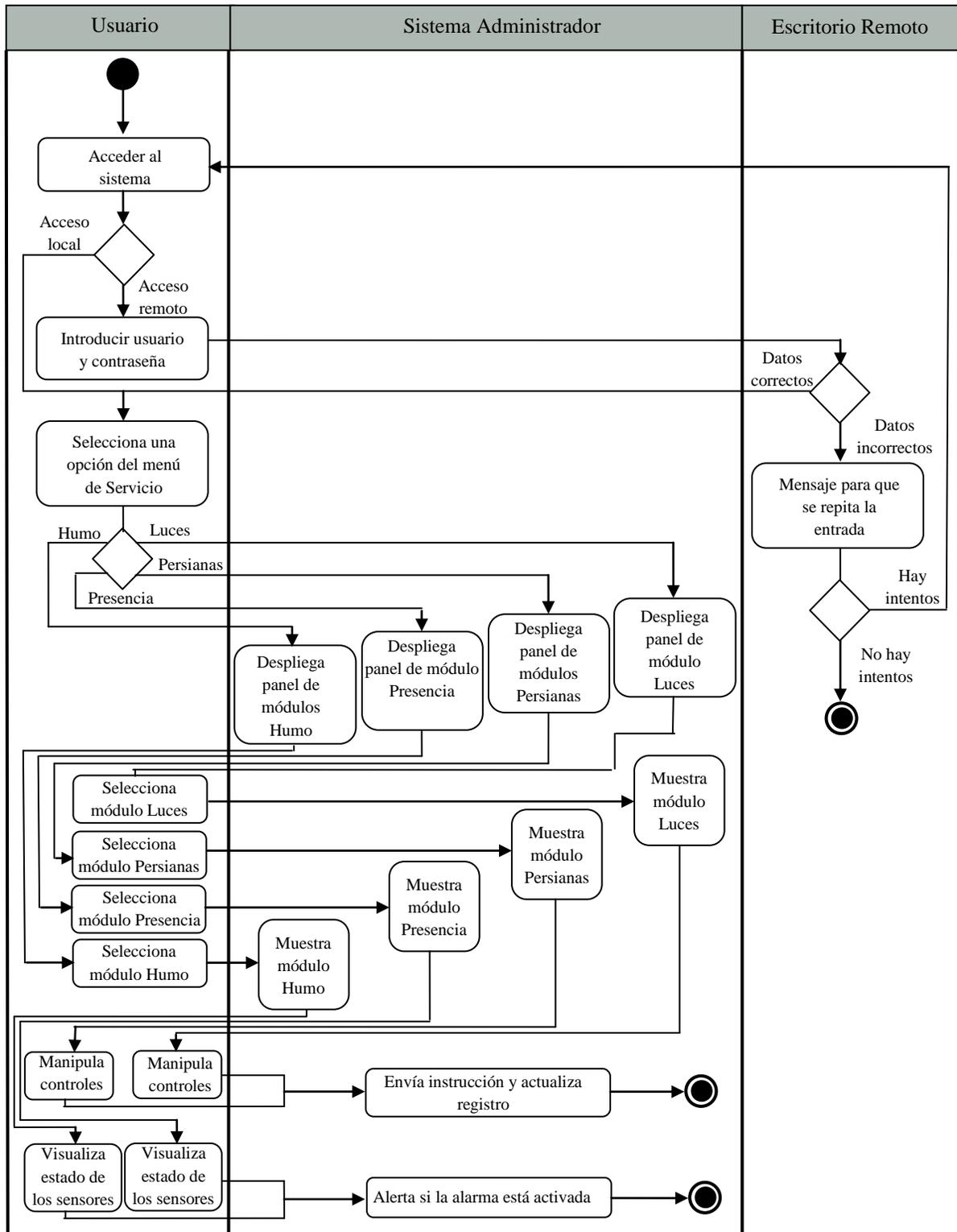


Figura 17. Diagrama de canal para la función servicios del sistema.

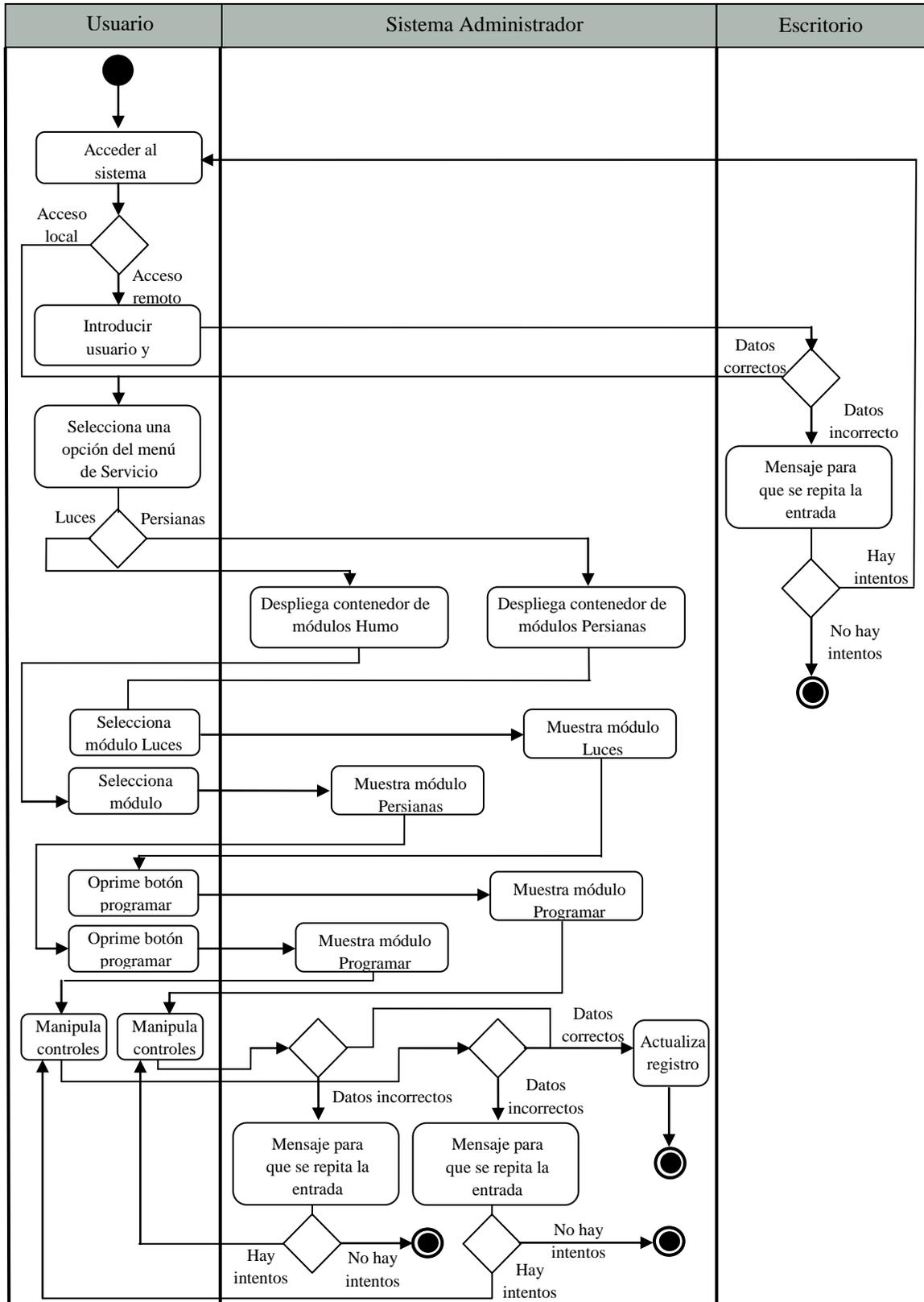


Figura 18. Diagrama de canal para la función programar tareas.

3.7.5. Paquete de Análisis

Cualquier sistema grande se debe dividir en unidades más pequeñas, de modo que se limita la información del sistema en grupos pequeños [26]. Esto permite al analista poder comprender mejor el sistema desde un punto de alto nivel [26], dado que dentro de un paquete se encuentra una gran variedad de funciones, procesos, clases, etc. que están encapsulados para mayor comprensión.

En el sistema desarrollado dividimos en tres paquetes el sistema completo como se puede visualizar en la Fig. 19, lo que nos permite comprender mejor el sistema y percibir las partes que lo componen. En el subtema de diseño se aborda cada parte y se explica cada elemento que compone el paquete.

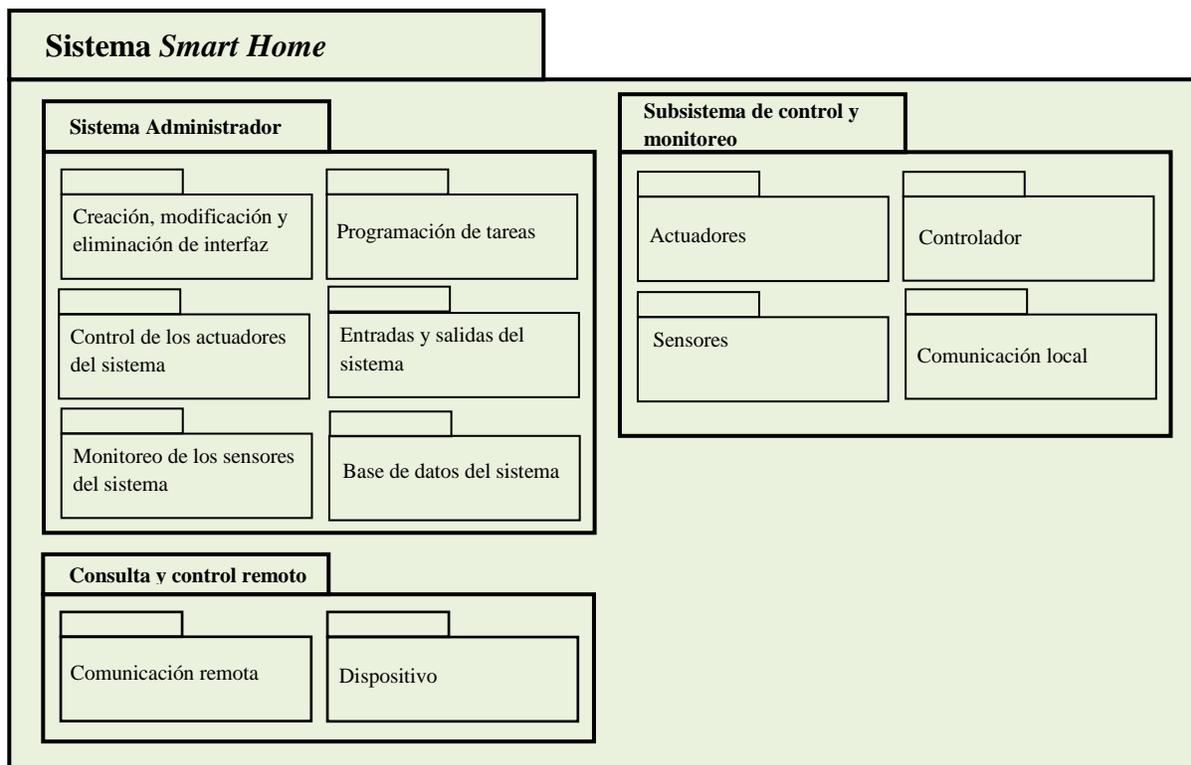


Figura 19. Paquetes de análisis del sistema.

3.7.6. Diagrama de Agregación

Es un diagrama en la que se representa un conjunto de componentes que integran otro componente superior o más grande [31]. Este diagrama permite visualizar todos los componentes que forman parte de un componente superior y la relación entre ellos. En la Fig. 20, se visualizan todos los componentes del software del sistema y la relación que existe entre ellos. Además en la Fig. 21, se visualizan los componentes del hardware del sistema y su relación.

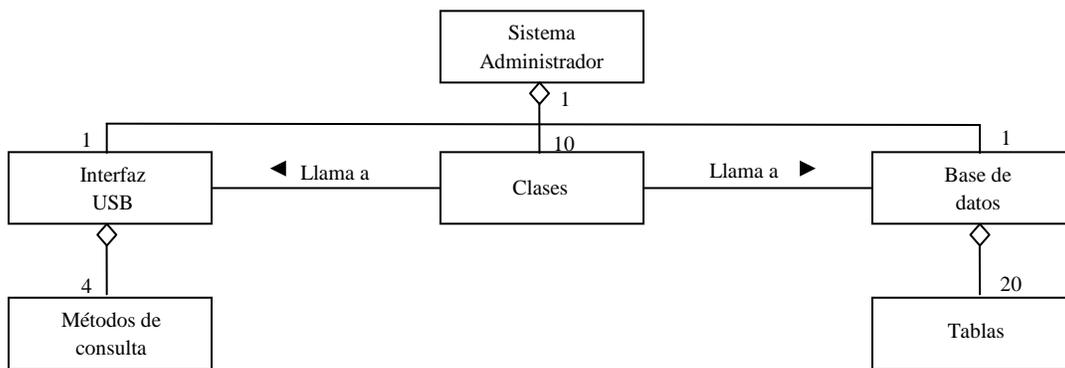


Figura 20. Diagrama de agregación del software.

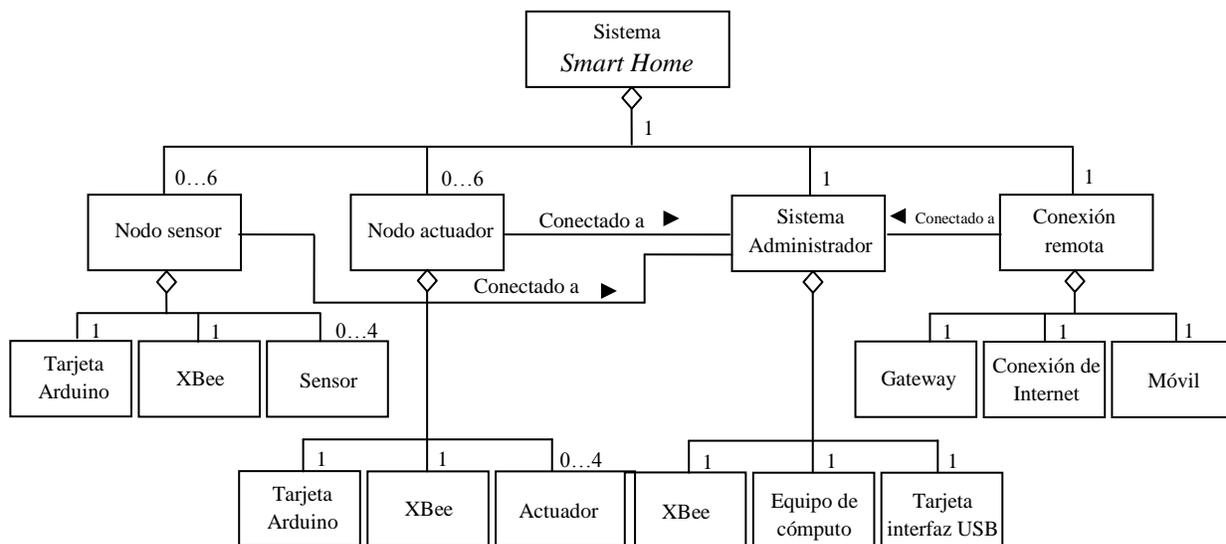


Figura 21. Diagrama de agregación del hardware del sistema.

3.8. Diseño

En esta sección se efectúa el diseño del sistema basándose en los diagramas desarrollados en el tema anterior, los cuales reflejan los requerimientos que el sistema debe cumplir.

En los requerimientos del sistema sobresale un aspecto importante, la parte de ajustes del sistema, lo que permite configurar y escalar el sistema. La configuración del sistema significa tener la opción de modificar los módulos de servicio para que la interfaz sea personalizada para el usuario, y la escalabilidad significa poder modificar el tamaño del sistema sin necesidad de llamar a un experto para realizar tal actividad. Todo lo anterior involucra uno de los campos de investigación más importante de la tecnología orientada a objetos, el desarrollo de un herramienta CASE (Ingeniería de Software Asistida por Computadora) [33]. Esta parte del sistema es la aportación más importante de esta tesis, por lo que se le destina el siguiente subtema especialmente para abordar el tema y en los subtemas siguientes se desarrolla los diagramas de diseño del sistema.

3.9. CASE

Esencialmente, CASE es una herramienta que ayuda al ingeniero de software a desarrollar y mantener el software. Por definición, el término CASE representa las herramientas individuales para ayudar al desarrollador de software o administrador de proyecto durante una o más fases del desarrollo de software (o mantenimiento) [30].

Según [26], son herramientas de construcción de prototipos de interfaz que permiten una rápida creación en pantalla de interfaces de usuario sofisticadas, que se ajustan al estándar de interfaz que se haya adoptado para el software.

Las herramientas CASE constituyen uno de los principales factores para aumentar la productividad de software [33]. Es por ello que en las últimas dos décadas se ha visto un creciente interés en la tecnología CASE [34]. Esta tecnología ha despertado el interés de distintas áreas de la ciencia de la computación como el de la inteligencia artificial, la cual ofrece un enfoque prometedor en el subcampo de Procesamiento del Lenguaje Natural para ayudar a los ingenieros de software en la fase de análisis [34]. Sin embargo, en la actualidad las herramientas CASE solamente se especializan en alguna etapa del desarrollo de software, dejando a un lado el deseo de tener una herramienta que ayude a desarrollar un software en todas sus etapas [34].

Al desarrollar software es indispensable contar con entornos de desarrollo y herramientas como las CASE para facilitar el desarrollo de la misma [35]. En el ciclo de vida del software cada vez toma más fuerza la generación automática de los diferentes esquemas conceptuales y de código, dado que se reduce el tiempo de desarrollo y, por lo tanto, los costos en los que se incurre durante el proceso [36]. También se minimizan los errores que pueda cometer el analista debido a la automatización del proceso de desarrollo [30][36]. También se busca una mayor interacción con el cliente, de tal modo que se pueda tener una validación en tiempo real [36].

Debido a lo anterior en este proyecto se desarrolla una herramienta CASE que automatice la etapa de mantenimiento del sistema de software planteado en esta tesis, la etapa de mantenimiento involucra los ajustes que el usuario desee realizar, siendo esta la reconfiguración y la escalabilidad del sistema.

3.9.1. Clasificación de CASE

Las clasificaciones de las herramientas CASE nos ayudan a comprender su papel en el proceso de software. Existen varios tipos de clasificaciones, cada una de ellas nos proporciona una perspectiva diferente, algunas son [28]:

- Perspectiva funcional: se clasifica de acuerdo con su función específica.
- Perspectiva de proceso: se clasifica de acuerdo con las actividades del proceso que ayudan.
- Perspectiva de integración: se clasifica de acuerdo con la forma en la que están organizadas en unidades integradas que proporcionan ayuda a una o más actividades del proceso.

En este trabajo se prefiere mostrar la perspectiva funcional, debido a que es más específico en la clasificación (véase Tabla 7) [28].

Tabla 7. Clasificación de las herramientas CASE.

Herramientas de planificación	Herramientas PERT, herramientas de estimación, hojas de cálculo
Herramientas de edición	Editores de texto, editores de diagramas, procesadores de texto
Herramientas de gestión del cambio	Herramientas de rastreo de requerimientos,

	sistemas de control de cambios
Herramientas de gestión de la configuración	Sistemas de gestión de las versiones, herramientas de construcción de sistemas
Herramientas de construcción de prototipos	Lenguajes de muy alto nivel, generadores de interfaz de usuario
Herramientas de apoyo al método	Editores de diseño, diccionario de datos, generadores de código
Herramientas de procesamiento de lenguaje	Compiladores, intérpretes
Herramientas de análisis de programa	Generadores de referencias cruzadas, analizadores estáticos, analizadores dinámicos
Herramientas de pruebas	Generadores de pruebas de datos, comparadores de archivos
Herramientas de depuración	Sistema de depuración interactiva
Herramientas de documentación	Programas de diseño de páginas, editores de imágenes
Herramientas de reingeniería	Sistemas de referencias cruzadas, sistemas de reestructuración de programas

En base al requerimiento de ajustes del sistema donde se pide que el sistema pueda variar el número de módulos de los servicios y además que se puedan reconfigurar esos módulos, se puede visualizar que esta se encuentra en la clasificación de herramientas de construcción de prototipos debido a que se requiere que el sistema genere una interfaz de controles que permita manipular los servicios del sistema. Por ende la herramienta desarrollada en este proyecto es una herramienta de construcción de prototipos.

3.9.2. Componentes de CASE

En general, las herramientas de tipo CASE incluyen alguno de los siguientes componentes: herramientas para diagramación, un depósito de información, generador de interfaces, generador de códigos y herramientas de administración [37].

- Herramientas para diagramación: Dan soporte al análisis y documentación de los requerimientos de una aplicación.

- Depósito de información: La captura, análisis, procesamiento y distribución de todos los sistemas de información es asistida por un depósito de información centralizado o diccionario de datos.
- Generador de interfaces: Ofrecen la capacidad para preparar imitaciones y prototipos para las interfaces con los usuarios. Por lo general, soportan la rápida creación de menús para el sistema y de presentación en pantalla.
- Generadores de código: Automatizan la preparación de software. Éstos incorporan métodos que permiten convertir las especificaciones del sistema en código ejecutable.
- Herramientas de administración: Ayuda a los gerentes de desarrollo a calendarizar las actividades de análisis y diseño así como la asignación de recursos a las diferentes actividades del proyecto.

La herramienta CASE que se desarrolla en esta tesis, contiene el depósito de información que es la base de datos que registra todo ajuste que se realice en el sistema y el generador de interfaces que son las clases que integran el sistema administrador.

Por otro lado se debe definir qué tipo de interfaz generará la herramienta CASE, de acuerdo con [38] existen dos tipos:

- RAD (entornos de desarrollo rápido) y herramientas de autor: Son herramientas de diseño para construir directamente la interfaz de usuario.
- Herramientas basadas en modelos: Son herramientas basadas en modelos para intentar aumentar el nivel de abstracción. Se apoyan en generadores de códigos parciales o totales.

Para el desarrollo de este trabajo, se optó por utilizar el primer tipo debido a que tiene un menor nivel de abstracción y la construcción de interfaces de usuario es muy rápida, se construye la interfaz de usuario mediante el paradigma WYSIWYG: eligiendo los componentes de la interfaz, ajustando sus propiedades y programando el enlace con la lógica adecuada [38].

Una vez analizada la herramienta CASE desarrollada en esta tesis, se procede a realizar los diagramas de diseño que explican el sistema.

3.10. Diagramas de Diseño

Los diagramas de diseño nos brindan la conceptualización del sistema administrador desde distintas ópticas, con el fin de visualizar los procesos más importantes de esta. En los diagramas se trata de exponer los aspectos más significativos del sistema administrador, evitando mostrar contenidos innecesarios o etapas de menor importancia. Las vistas que nos ayudan a desarrollar el sistema administrador son las siguientes:

- Diagrama de estado
- Diagrama de secuencia
- Diagrama de colaboración
- Diagrama de actividades
- Modelo relacional

3.10.1. Diagrama de Estado

Un diagrama de estado describe el comportamiento dinámico de los objetos, en un cierto plazo, modelando los ciclos de vida de los objetos de cada clase involucrado [39]. Cada objeto se trata como una entidad aislada que se comunica con el resto del mundo detectando eventos y respondiendo a ellos. En las Figs. 22 y 23 se visualizan los diagramas de estados de menú CASE y menú servicio, respectivamente.

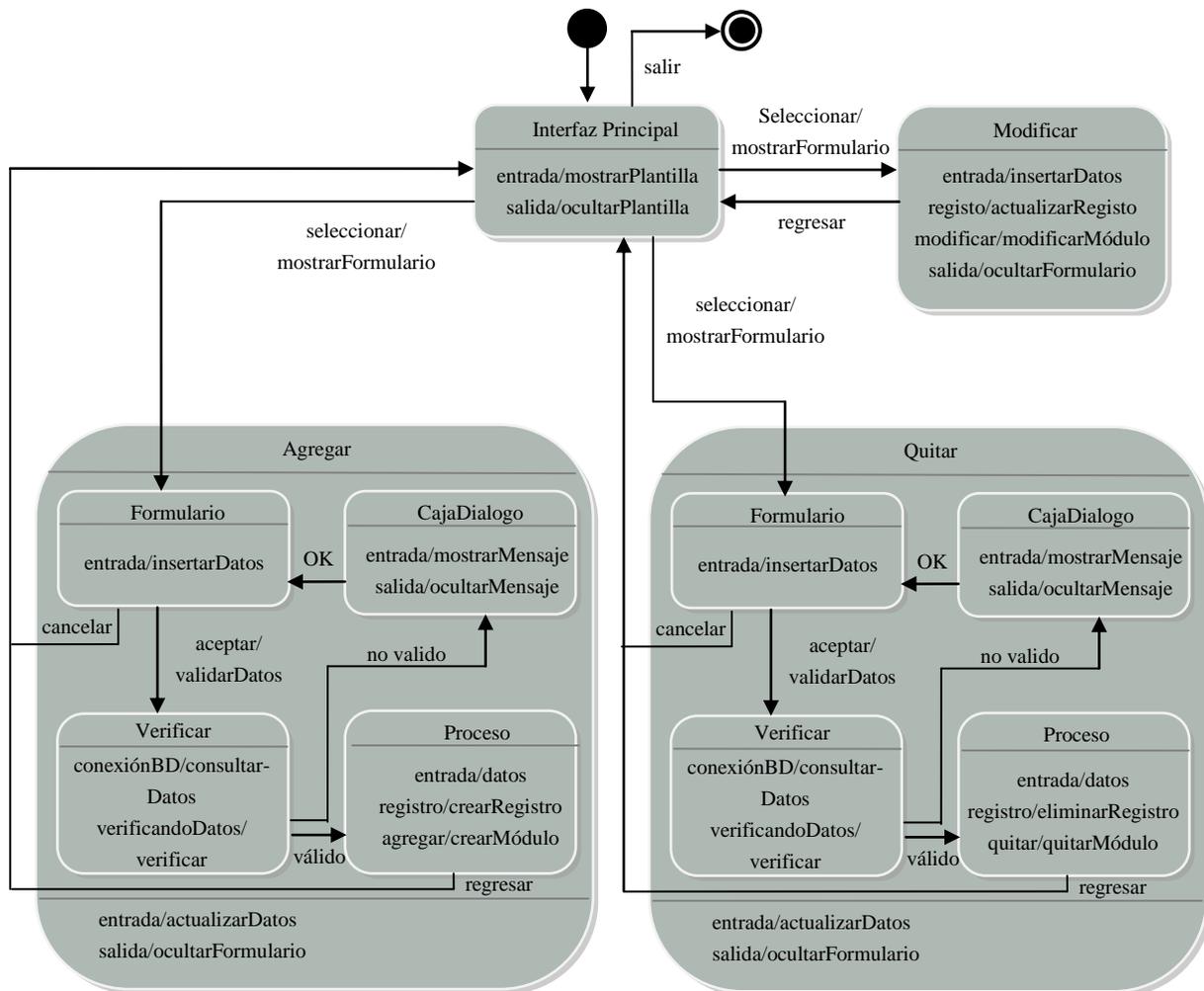


Figura 22. Diagrama de estado del menú CASE.

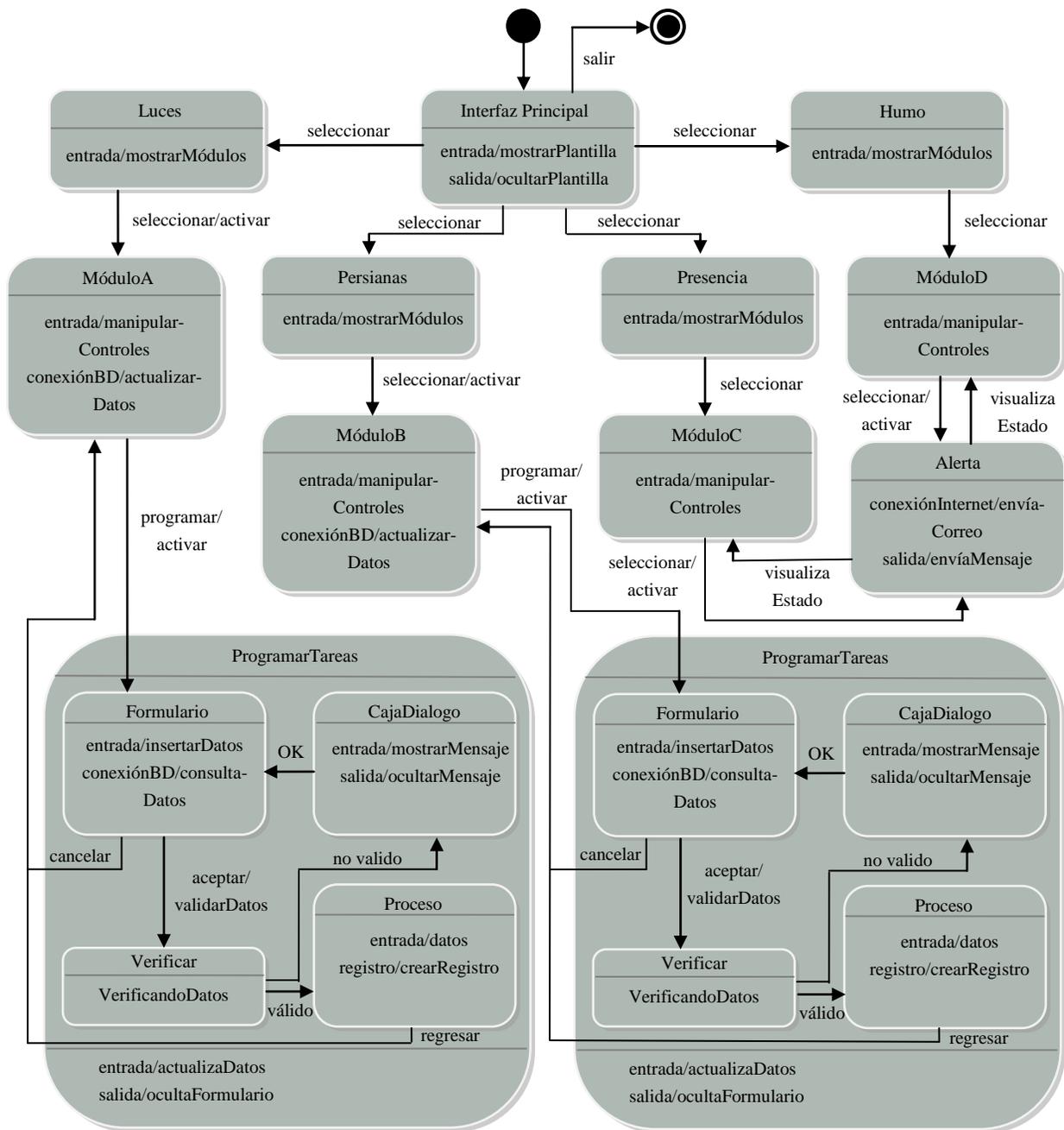


Figura 23. Diagrama de estado del menú servicio.

3.10.2. Diagrama de Secuencia

Los diagramas de secuencia son los modelos dinámicos más habituales desarrollado por los diseñadores de sistemas [40]. Los diagramas de secuencia se usan para formalizar el comportamiento del sistema y para visualizar la comunicación entre objetos o clases y permiten observar los eventos realizados en una orden de tiempo de ejecución. Así mismo un diagrama de secuencia representa una interacción como un gráfico bidimensional. La dimensión vertical es el eje de tiempo, que avanza hacia abajo y la dimensión horizontal muestra los roles de clasificador que representan objetos individuales o clases en la colaboración del diagrama [39]. En las Figs. 24-29 se visualizan los diagramas de secuencia del sistema administrador.

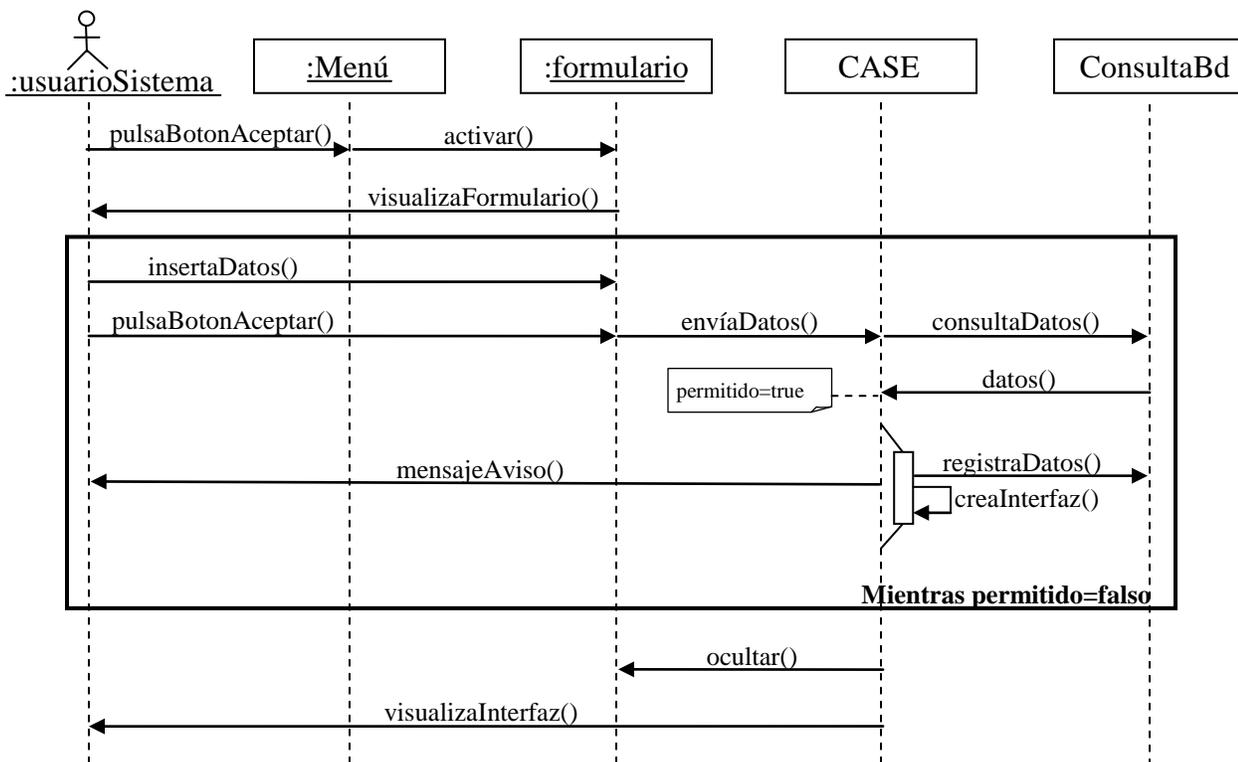


Figura 24. Diagrama de secuencia para el escenario de crear interfaz.

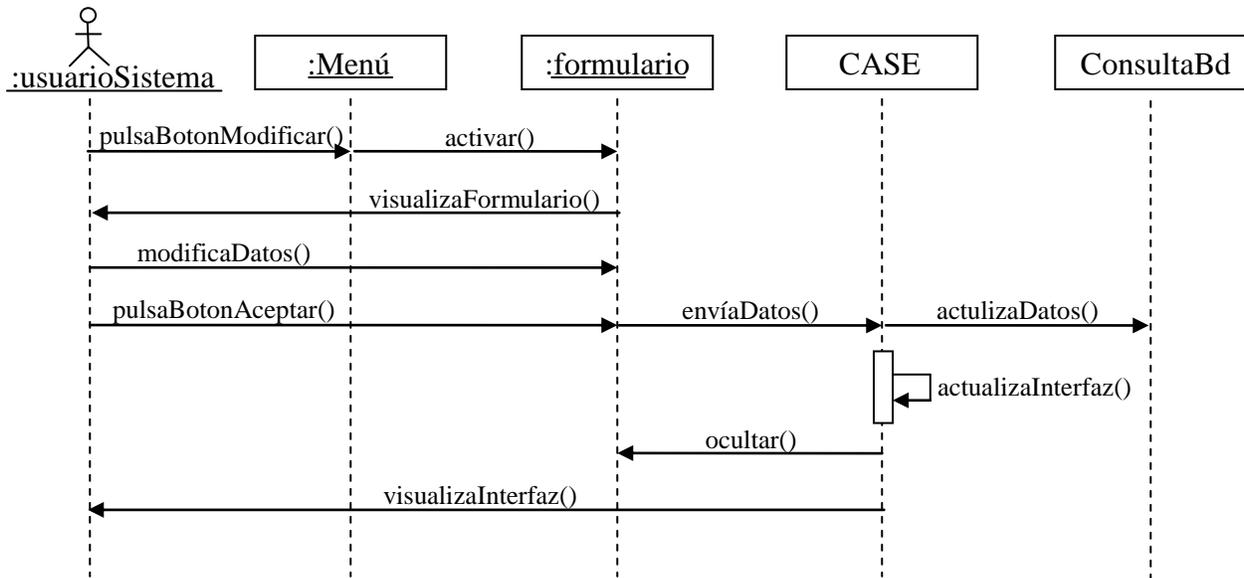


Figura 25. Diagrama de secuencia para el escenario de modificar interfaz.

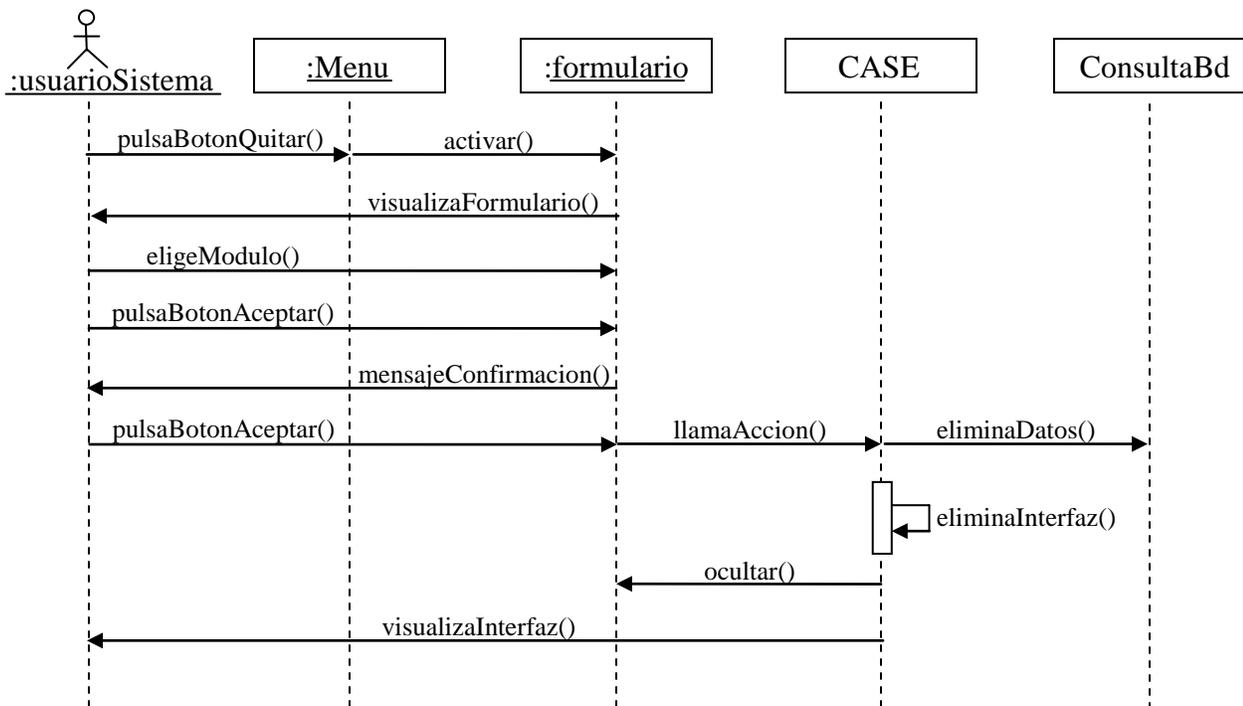


Figura 26. Diagrama de secuencia para el escenario de quitar interfaz.

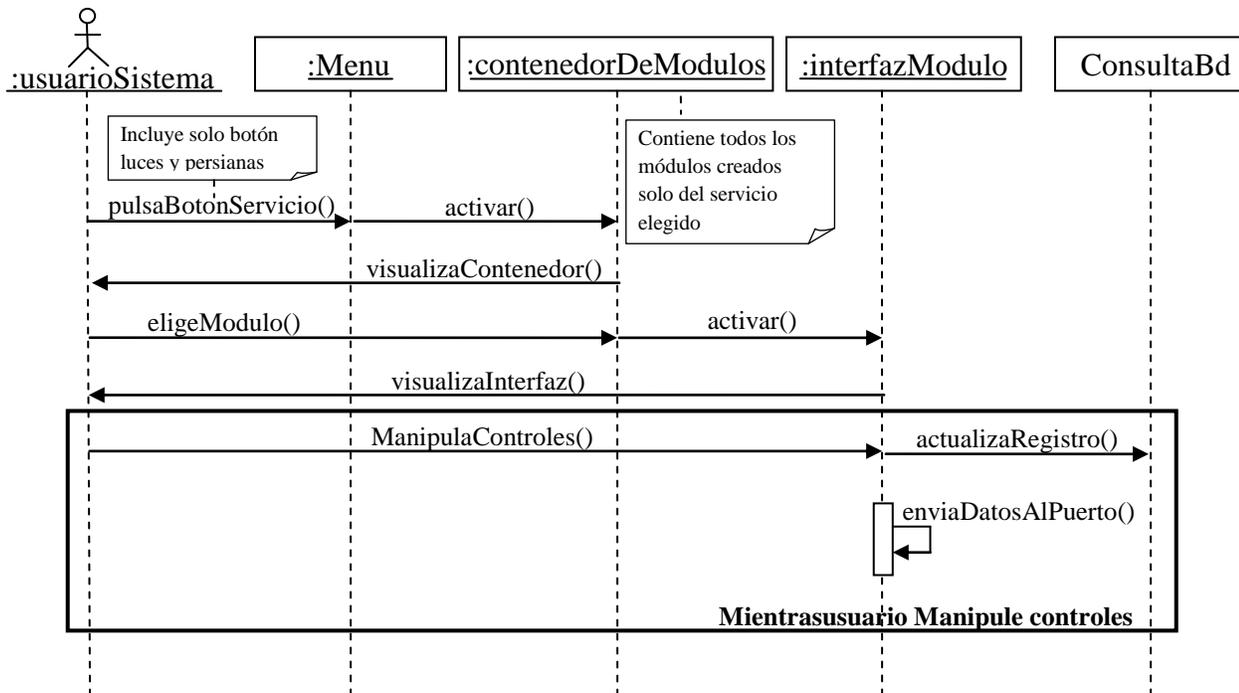


Figura 27. Diagrama de secuencia para el escenario de control de actuadores.

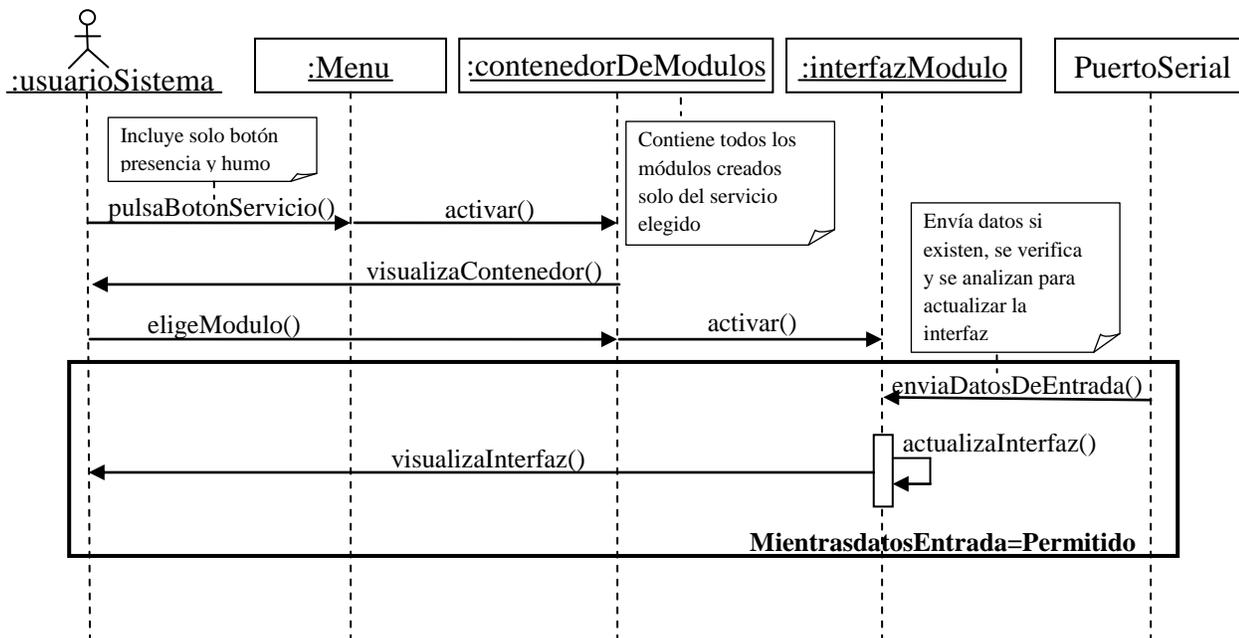


Figura 28. Diagrama de secuencia para el escenario de monitoreo de sensores.

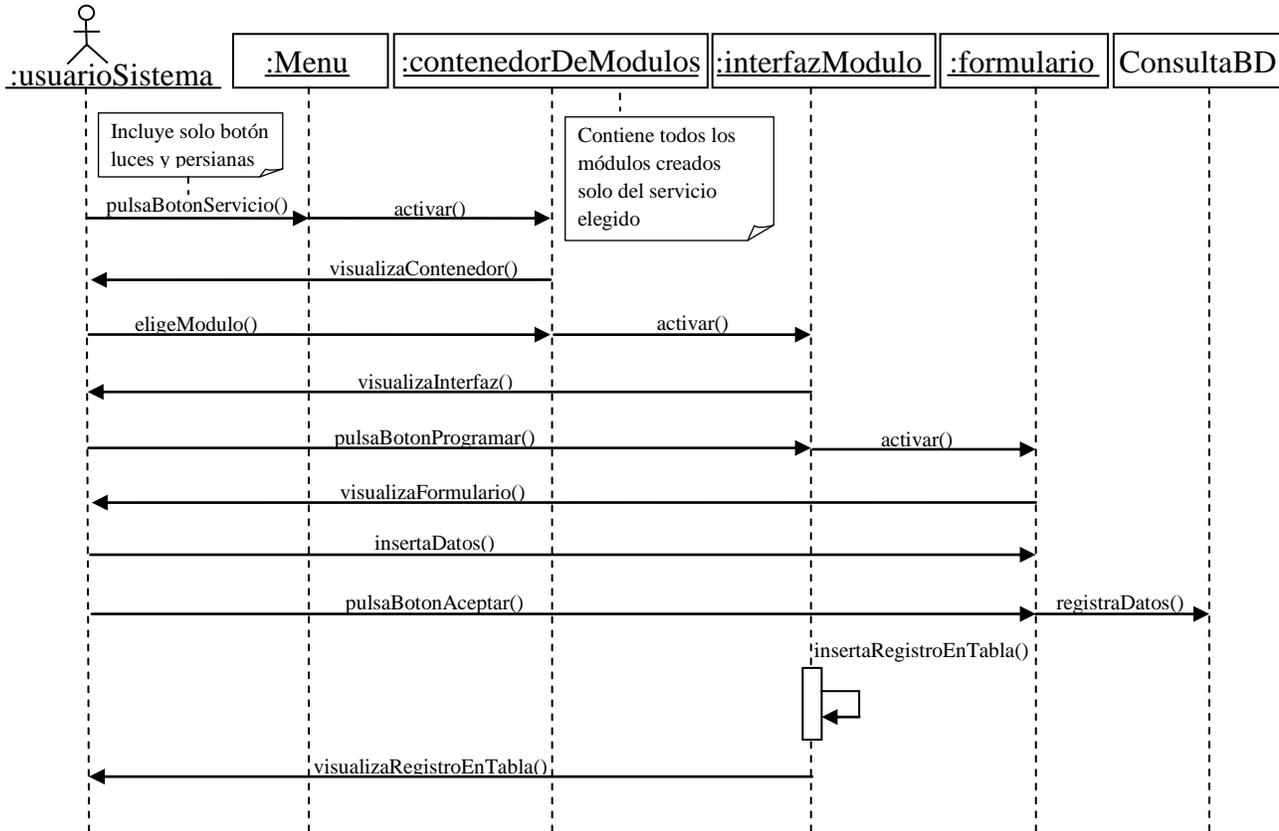


Figura 29. Diagrama de secuencia para el escenario de programación de tareas.

3.10.3. Diagrama de Actividades

El diagrama de actividades proporciona una representación gráfica del flujo de interacción dentro de un escenario específico [26]. Un diagrama de actividades contiene estados de actividad. Un estado de actividad representa la ejecución de una sentencia en un procedimiento, o el funcionamiento de una actividad en un flujo de trabajo. En la Fig. 30 se representa la actividad de monitoreo del puerto COM4 para la recepción de paquetes de los nodos de sensores del sistema y en la Fig. 31 se representa la actividad de ejecución de tareas programadas.

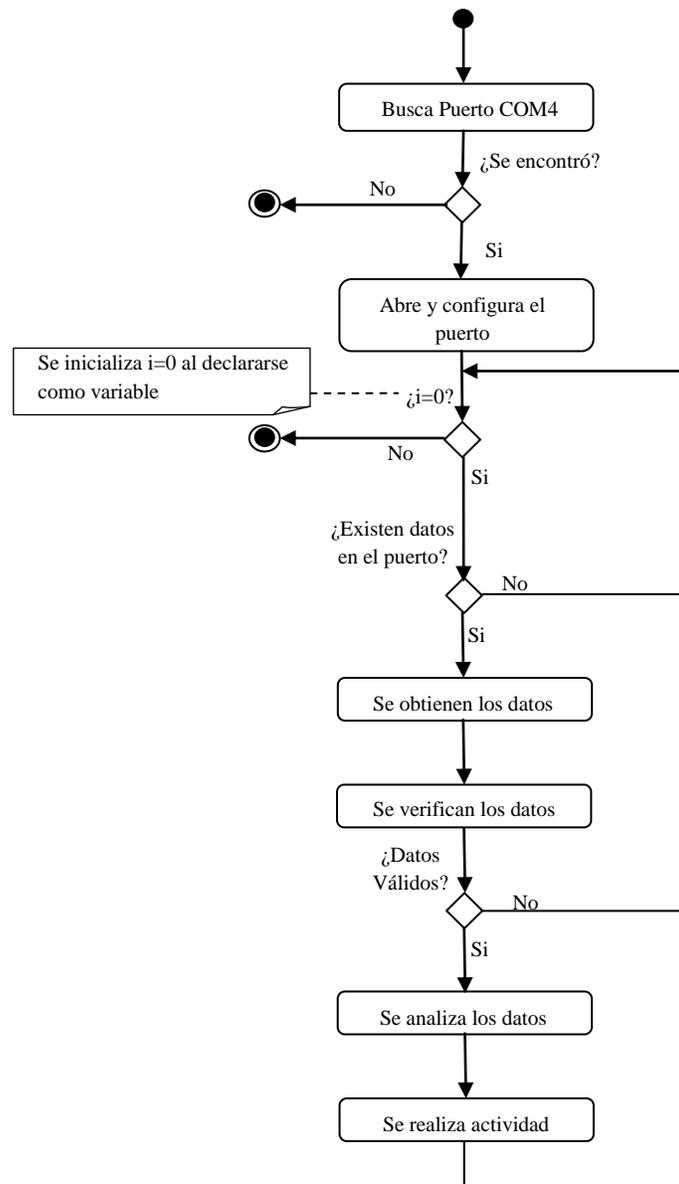


Figura 30. Diagrama de actividad para el monitoreo del puerto COM4.

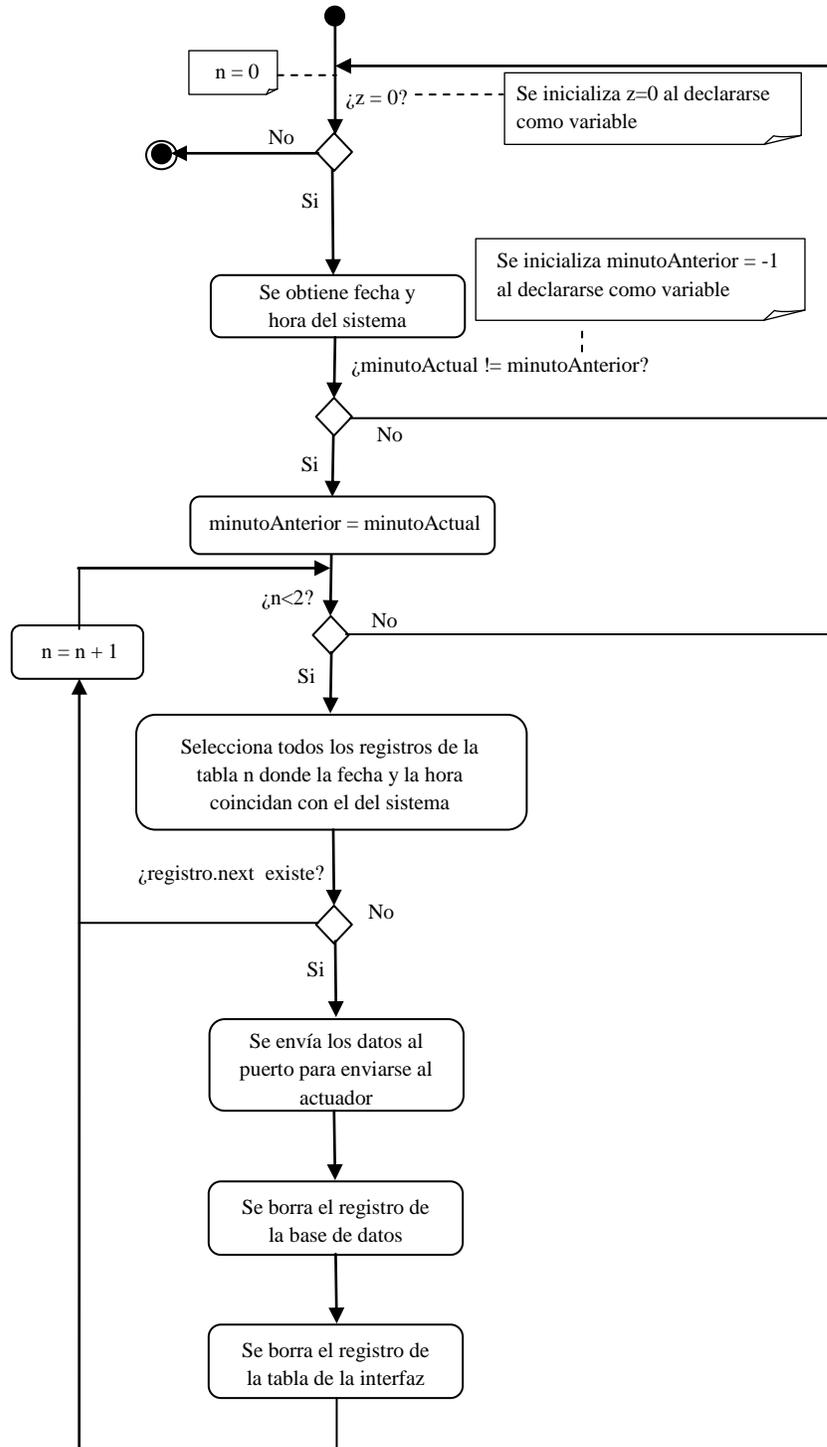


Figura 31. Diagrama de actividad para realizar las tareas de programadas.

3.10.4. Modelo de la Base de Datos Relacional

Hoy en día el modelo relacional es el principal modelo de datos para aplicaciones comerciales de procesamiento de datos [32]. Ha conseguido esa posición debido a su simplicidad, lo cual facilita el trabajo del programador en comparación con otros modelos [32]. Una base de datos relacional consiste en un conjunto de tablas, a cada una de las cuales se le asigna un nombre exclusivo. Cada fila de la tabla representa una relación entre un conjunto de valores [32]. De manera informal se puede decir que cada tabla es un conjunto de entidades y cada fila es una entidad. Cada tabla está relacionada con otras tablas por medio de la cardinalidad que le corresponda. Detrás del modelo de datos relacional existe un fundamento teórico el cual se puede consultar en [32] para mayor profundidad en el tema. En esta sección nos limitamos a mostrar el modelo de la base de datos relacional del sistema administrador creado el cual se muestra en las Fig. 33 con su descripción.

Para el sistema administrador es indispensable tener, en cada instante de tiempo, el estado de cada actuador del sistema para su buen funcionamiento. Si el estado del actuador en el sistema no se guarda, esto puede acarrear problemas, como el de no realizar la tarea que desea el usuario al querer apagar las luces de la habitación o cerrar alguna persiana de la casa. Por otro lado para que la herramienta CASE del sistema pueda funcionar requiere guardar todas las modificaciones que se realizaron en la interfaz como la creación de un módulo y sus controles (objetos) que contiene o la eliminación de un módulo. Si no se guardan las modificaciones realizadas al reiniciar el sistema se perderán todas las modificaciones hechas anteriormente y se empezaría de nuevo con la creación de la interfaz. Además para la programación de tareas es necesaria una base de datos.

Para evitar los problemas anteriores se desarrolla una base de datos relacional la cual guarda los estados de los actuadores y las modificaciones realizadas en la interfaz.

Los estados de los actuadores se guardan en las tablas estado_panel_0 y estado_panel_1, la primera tabla mencionada guarda los estados de las luces y la segunda la de persianas.

Los controles (objetos) contenidos en cada módulo se registran en las tablas características_panel_n, donde n=0,1,2,3.

Los módulos creados de cada servicio están registrados en las tablas paneles_creadas_n, donde n=0,1,2,3.

Las tareas programadas se guardan en la tabla programar_0 y programar_1, la primera tabla mencionada guarda las tareas programadas del servicio luces y la segunda del servicio persianas. No se pueden programar tareas para los sensores por lo que no existe tabla para ello.

El diseño de la base de datos se visualiza en las Figs. 32-33.

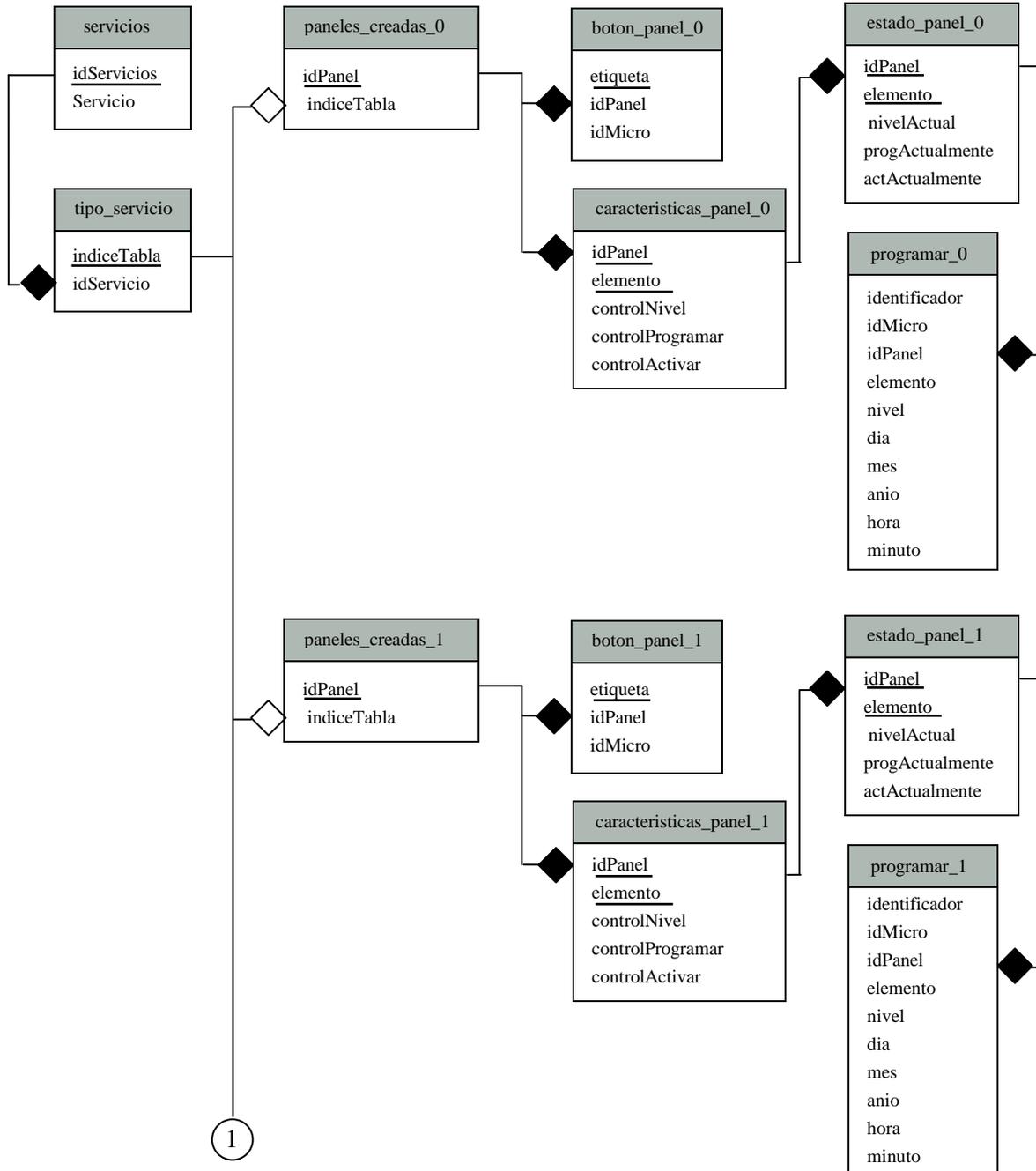


Figura 32. Diseño de la base de datos del sistema parte (a).

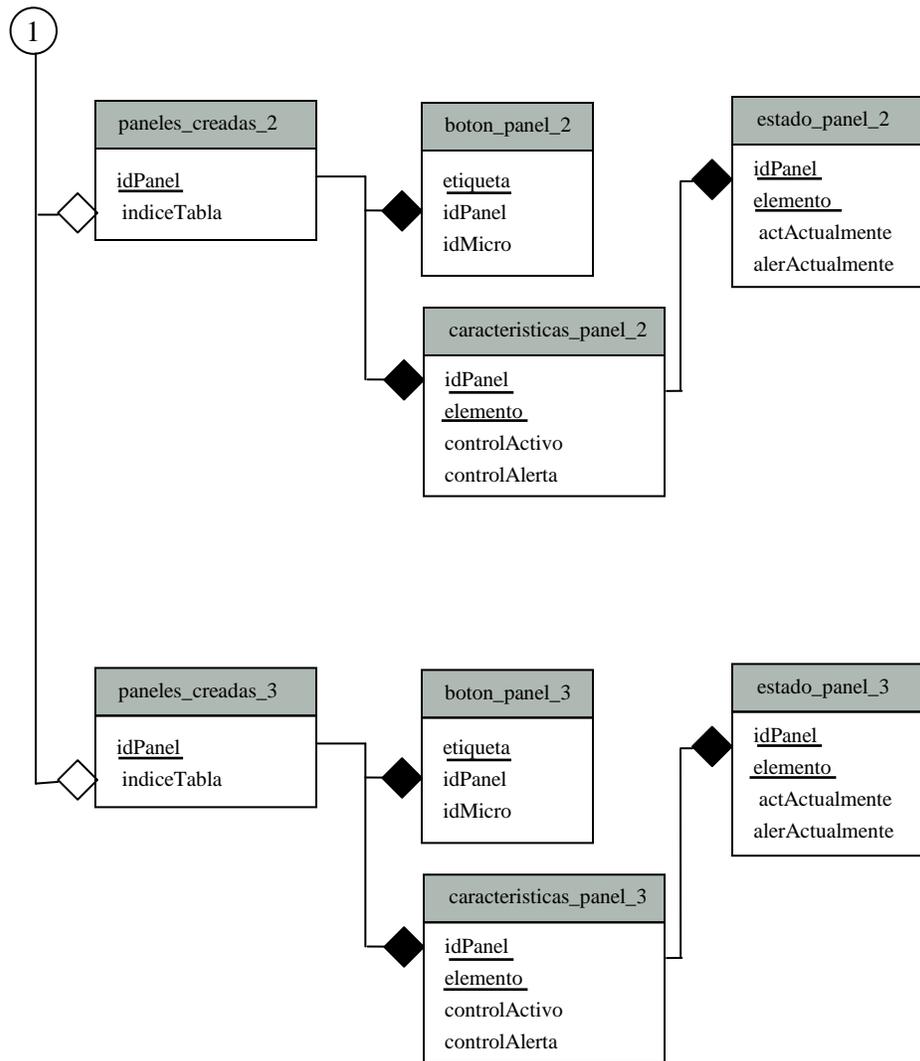


Figura 33. Diseño de la base de datos del sistema parte (b).

3.11. Resumen del Capítulo

En este capítulo se analizaron los requerimientos del sistema y se diseñaron los diagramas UML que son la base para el desarrollo del sistema administrado. En el capítulo siguiente se analiza el hardware del sistema para su implementación y con ello concluir la integración de las partes del sistema y pasar a la etapa de pruebas.

Capítulo 4

Implementación en Hardware

En el capítulo anterior se mencionó que el sistema *Smart Home* propuesto en esta tesis está conformado por tres partes: sistema administrador, subsistema de control y monitoreo y consulta y control remoto. En el Capítulo 3 se abordó la parte del sistema administrador, por lo tanto, en este capítulo se abordará las partes subsistema de control y monitoreo, consulta y control remoto que son parte del hardware del sistema.

4.1. Subsistema de Control y Monitoreo

El subsistema de control y monitoreo es la parte que se instala dentro de la casa, en lugares donde se desea tener servicio de control o monitoreo que brinda el sistema. El subsistema lo integran la parte de sensores, actuadores, controladores y la red inalámbrica.

La función que se encarga de realizar este subsistema, es la de integrar todas las partes instaladas en la casa y proporcionar la comunicación entre ellas y el sistema administrador de manera eficiente. Los elementos que se instalan dentro de la casa están conectados a un controlador (Arduino [41]) el cual permite realizar dos procesos: primero, la conversión de las señales analógicas que emiten los sensores a valores digitales, los cuales son transmitidos al sistema administrador para evaluación, y el segundo es, la conversión de los valores binarios emitidos por el Sistema Administrador a una señal digital PWM (Modulación de Ancho de Pulso) que se envía a los actuadores para controlar su intensidad de iluminación o su ángulo de posición. En los temas siguientes se explica el funcionamiento de cada elemento del subsistema.

4.1.1. Sensores del Sistema

El sistema requiere monitorear la presencia de personas y la detección de humo dentro del hogar. Para realizar el monitoreo se adquirieron los sensores analógicos Sharp 2Y0A02 y MQ2 para el monitoreo de presencia y humo, respectivamente.

El sensor Sharp 2Y0A02 (Fig. 34) tiene las características mencionadas en la Tabla 8:



Figura 34. Imagen del sensor Sharp 2Y0A02.

Tabla 8. Características del Sharp 2YoA02.

Parámetro	Rango
Infrarrojo del visor	20 cm (~ 8 ") a 150 cm (~ 60")
Diferencia de voltaje de salida	2.05V a 20 cm - 0.4V a 150 cm
Interfaz	Analógica
Consumo de energía	33 mA
Tiempo de respuesta	39 ms
Alimentación	5 V

El sensor MQ2 (Fig. 35) tiene las características mocionadas en la Tabla 9.



Figura 35. Imagen del sensor MQ2.

Tabla 9. Características del MQ2.

Parámetro	Rango
Detección de partículas	300 a 10000 ppm
Diferencia de voltaje de salida	2.5 V a 300 ppm a 4 V a 10000 ppm
Interfaz	Analógica
Consumo de energía	150 mA
Tiempo de respuesta	≤ 10s
Alimentación	5 V

Los sensores emiten una variación de voltaje al detectar presencia o partículas de algún gas. El valor de la señal analógica que emite el sensor depende, en el caso del sensor Sharp 2Y0A02, de la distancia del objeto que interfiere la señal infrarroja que emite el sensor (véase Tabla 8); en el caso del sensor MQ2, de la concentración de partículas de CO₂ (véase Tabla 9). La señal emitida por los sensores es enviada a la tarjeta Arduino donde la señal analógica es convertida a señal digital y enviada al Sistema Administrador. Esta parte se explica en la sección 4.1.3.

4.1.2. Actuadores del Sistema

El sistema requiere controlar las luces y persianas de un hogar. El control de las persianas se realiza mediante motores que permiten posicionar las persianas a un ángulo deseado, el control de las luces se puede realizar mediante PWM de una señal o de una fuente de energía, esta técnica permite modificar el ciclo de trabajo de una señal periódica, con el objetivo de transmitir información o para controlar la cantidad de energía que se envía a una carga. En este caso el objetivo es controlar la energía enviada permitiendo así el control de la intensidad de luz.

Los actuadores que se utilizaron en el sistema para el control de las luces y persianas fueron servomotores Futaba 3003 y diodos emisores de luz (LEDS) respectivamente.

Los servomotores (Fig. 36) funcionan mediante señales PWM, cada modelo de servomotor tiene sus características. Las características de los servomotores Futaba 3003 se mencionan en la Tabla 10.

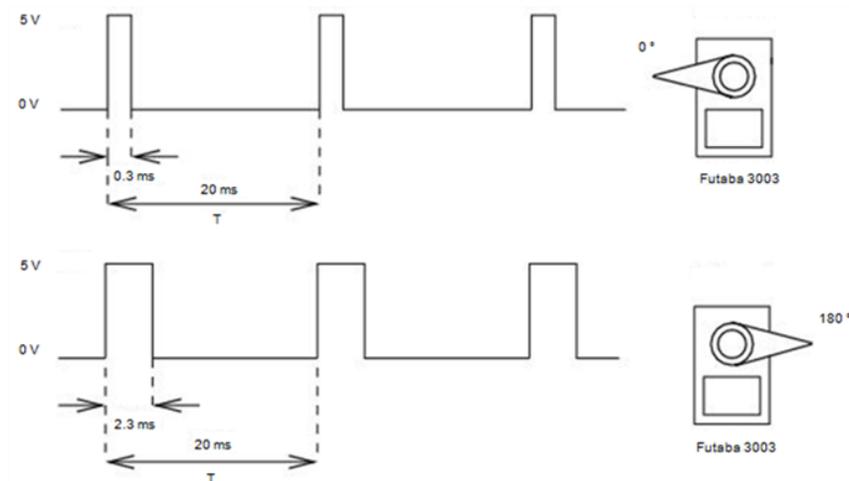


Figura 36. Imagen del servomotor Futaba 3003.

Tabla 10. Características de Futaba 3003.

Parámetro	Rango
Velocidad:	0.23 seg/60 grados
Par de salida:	3.2 Kg-cm (0.314 N.m)
Dimensiones:	40.4 x 19.8 x 36 mm
Peso:	37.2 gr
Frec. PWM:	50Hz (20ms)
Rango giro:	180 grados

Para posicionar el servomotor Futaba 3003 se requiere aplicar una señal periódica, de 50 Hz (20 ms de periodo). La anchura del pulso determina la posición del servomotor. Si la anchura es de 0.3 ms, el servomotor se sitúa en un ángulo de 0° y si la anchura es de 2.3 ms se sitúa en un ángulo de 180°. Cualquier otra anchura entre 0.3 y 2.3 sitúa el servomotor en una posición comprendida entre 0° y 180° (véase Fig. 37).

**Figura 37.** Señal PWM para Futaba 3003.

Para el control de las luces del sistema se utilizaron un conjunto de LEDs que simulan las luces del hogar (véase Fig. 38). Como se mencionó anteriormente, el control de la intensidad de la luz se realiza mediante señales PWM que permiten controlar la cantidad de energía que se envía a una carga. El proceso es similar al control de un servomotor, si el ancho de pulso es 0 ms, el LED permanecerá apagado, mientras que si el ancho de pulso es 3 ms, el LED permanecerá encendido con intensidad alta de luz. La alimentación de energía es de 3 V.

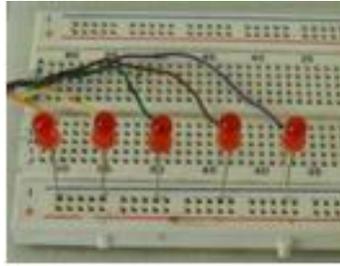


Figura 38. Matriz de Leds.

4.1.3. Controlador

Para el monitoreo y control de los sensores y actuadores es necesario utilizar un controlador que permita realizar esta tarea. Esto se debe a que se requiere un proceso de análisis de los datos y una conversión de señal analógico-digital como lo muestra la Fig. 39.

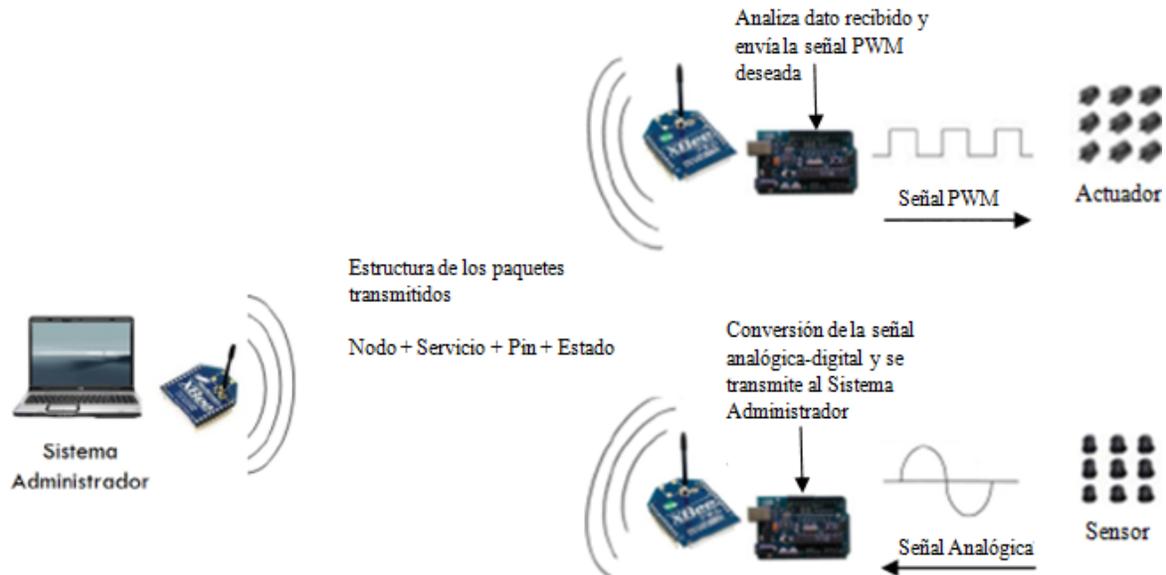


Figura 39. Proceso de comunicación para realizar el monitoreo y control.

El controlador está conectado a los sensores o actuadores, lo que permite la manipulación de las señales necesarias para la realización de las tareas deseadas del sistema. En la actualidad existe una gran diversidad de controladores que están integrados en tarjetas electrónicas que cuentan con puertos de entrada/salidas para el desarrollo de proyectos. En este proyecto se trabajó con la tarjeta Arduino UNO basada en un Microcontrolador ATmega 328, con puertos de entrada y salida, analógico y digital, y una interfaz de comunicación USB (véase la Fig. 40). El entorno de programación de la tarjeta Arduino nos permite operar de forma cómoda y sencilla las señales

emitidas por el sistema. En la tarjeta Arduino se realiza uno de dos procesos, esto depende de la tarea asignada. El primero es la conversión de la señal analógica emitida por los sensores a un valor digital, la cual se envía al Sistema Administrador. La segunda es la conversión de los valores binarios emitida por el Sistema Administrador a una señal digital PWM que se envía a los actuadores para controlar su intensidad de iluminación o su ángulo de posición.



Figura 40. Tarjeta Arduino UNO.

Cada tarjeta Arduino manipula 4 elementos, en el caso de los sensores, estos se conectan a un puerto de entrada analógica y los actuadores en los puertos digitales. Además cada tarjeta Arduino se programa para una tarea específica, la de manipular sensores o actuadores.

4.1.4. Red Inalámbrica en el Interior del Sistema

Para crear la red del sistema se necesita definir su topología, es decir, la distribución física de los elementos respecto al medio de comunicación. Existen muchos tipos distintos de topologías: estrella, anillo, árbol, malla, línea o bus, entre otros. La que se utilizó para la arquitectura del sistema es la topología estrella, la cual representa un sistema centralizado. Esta topología nos da ventajas pero también existen desventajas. Las ventajas de esta topología son:

- Facilidad para añadir nuevos elementos (escalabilidad del sistema)
- Un fallo de un elemento no afecta al resto del sistema (independencia de los elementos)

Las desventajas de esta topología son:

- Un fallo en el controlador principal afecta a todo el sistema (proceso centralizado)
- Se produce un cuello de botella en el controlador principal (al existir intenso intercambio de información o muchos elementos)

Al analizar las ventajas y desventajas que implica utilizar una topología estrella, se observa que facilita la escalabilidad del sistema, debido a que solo se tiene que avisar al sistema central o administrador que un nuevo elemento se integra al sistema. También se observa que un fallo en un elemento del sistema no afecta al resto, por lo regular en un sistema *Smart Home* se modifican los elementos, como el cambio de baterías, etc., y esta actividad no afectará al sistema. Por otra parte, las desventajas son que ante la existencia de un fallo en el Sistema Administrador, el sistema deja de funcionar, esto constituye un riesgo que puede aceptarse con el respaldo de realizar diferentes pruebas que den una buena confiabilidad del Sistema Administrador. En el caso de la desventaja que se pueda producir un cuello de botella en el Sistema Administrador, es muy difícil que se presente en el sistema desarrollado debido a que el intercambio de información es mínimo y los elementos a manipular son relativamente pocos.

También se debe definir el medio físico a emplear en la red, según [2], los medios físicos que se utilizan en las redes de los sistemas *Smart Home* son: la línea eléctrica, el cable par trenzado y la de RF.

Algunos usuarios de *Smart Home* han utilizado la línea eléctrica por ser dispositivos fáciles de configurar, rápidos de desarrollar y baratas. Utilizan el código X10 en sistemas propietarios. El principal problema con este sistema se relaciona con la interferencia y los cortes de energía [2].

La utilización del cable par trenzado para transmitir datos a los dispositivos es la más utilizada actualmente según [2], por la seguridad y confiabilidad que esta brinda. Esta red es independiente de cualquier otra red instalada en el hogar. El inconveniente es la incomodidad de los cables, el costo y la poca movilidad que se tiene con la red y los elementos que la integran.

La adopción de la comunicación por medio de RF cada vez es mayor, por lo que últimamente el desarrollo de redes de este tipo en los sistemas *Smart Home* es de interés para los investigadores [2]. La mayoría de los fabricantes de tecnología para *Smart Home* tienen una gama de productos de RF, y cada vez son más seguros y confiables para su uso [2].

En este proyecto es de interés implementar una red inalámbrica utilizando tecnología de RF debido a la ventaja de bajo costo y movilidad que brindan las redes inalámbricas.

Existen muchas tecnologías inalámbricas para interiores, en la Tabla 11 se muestran las características de algunas de ellas.

Tabla 11. Características de redes para interiores.

Características	<i>Wifi</i>	<i>Bluetooth</i> (Básico)	<i>Bluetooth</i> (EDR)	<i>ZigBee</i>	<i>UWB</i>
Norma	802.11n	802.15	802.15	802.15.4	802.15.3a
Intervalo de Frecuencia	2.4 GHz a 2.497 GHz 5.15 GHz a 5.35 GHz 5.725 a 5.825 GHz	2400 MHz a 2483.5 MHz	2400 MHz a 2483.5 MHz	2.4 GHz a 2.4835 GHz (World) 902 MHz a 928 MHz (NorteAmérica) 868.3 MHz (Europa)	3.1 GHz a 10.6 GHz
Modulación	BPSK, QPSK, 16QAM, 64QAM	GFSK	Header: GFSK Data: $\pi/4$ DQPSK, 8DPSK	BPSK (868/915 MHz), OQPSK (MSK) (2.4 GHz)	QPSK
Acceso Múltiple	OFDM, CSMA/CA	FHSS	FHSS	CSMA/CA	TFI-OFDM
Duplex (<i>Uplink/Downlink</i>)	TDD (Time Division Duplex)	TDD	TDD	TDD	TDD
Ancho de Banda de Canal	20 MHz o 40 MHz	1 MHz	1 MHz	5 MHz	528 MHz
Número de Canales	14 2.4GHz 12 5 GHz	79	79	1 (868 MHz) 10 (915 MHz) 16 (2.4 GHz)	6 Grupos de Bandas
Máxima tasa de transferencia	<600 Mbps	1 Mbps	3 Mbps	20 kbps (868 MHz) 40 kbps (915 MHz) 250 kbps (2.4 GHz)	480 Mbps
Máxima Velocidad	10 Km/h	1 Km/h	1 Km/h	1 Km/h	1 Km/h
Máxima Distancia	100 m	10 m	10 m	10 m hasta 100 m	10 m

La elección de la tecnología inalámbrica a utilizar, se basó en el artículo [21] el cual presenta las características más sobresalientes de la tecnología *ZigBee*, como son la potencia de transmisión menor a 1 mW, lo que no afecta al usuario si está expuesta a ella, y la inmunidad a las interferencias de redes *Wifi* y a aparatos como el microondas, por lo que se decidió implementarla en el sistema.

Los dispositivos que se adquirieron para realizar la red inalámbrica local fueron los módulos *XBee* Serie 1, los cuales son fabricados por la compañía Digi (véase Fig. 41).



Figura 41. Módulo XBee Serie 1.

Estos dispositivos son configurables, lo que permite realizar diferentes tipos de conexiones, punto a punto, punto a multipunto, *broadcast*, cable virtual, además de otras configuraciones.

Para realizar una red con los módulos *XBee* se requiere principalmente un identificador de red y elegir un canal de transmisión (véase Fig. 42).

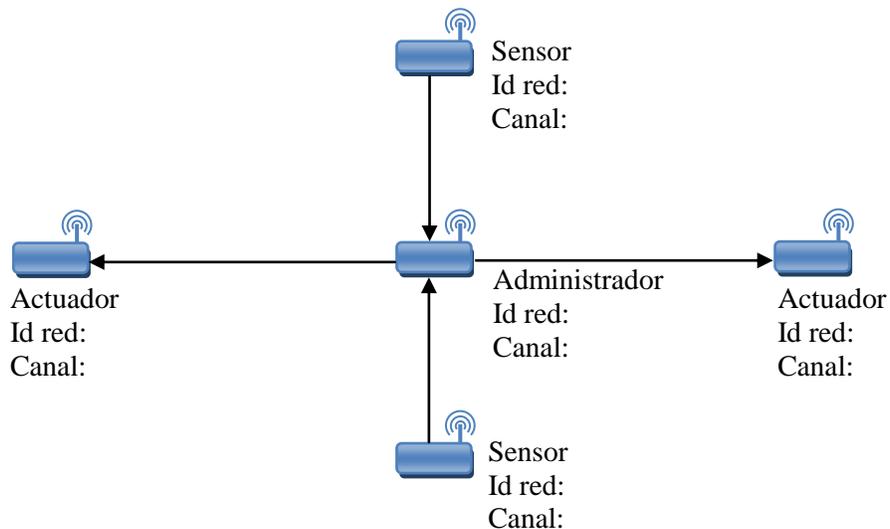


Figura 42. Red *XBee*.

4.2. Consulta y Control Remoto

La comunicación remota se realiza mediante escritorio remoto por medio del Protocolo de Internet (IP), esta conexión permite la manipulación del sistema vía remota, utilizando una terminal externa al sistema como una computadora o una terminal móvil.

La aplicación utilizada es el *Remote Desktop Connection* de *Windows 7* (véase la Fig. 43), consiste en una aplicación que trae integrado el sistema operativo, es fácil de usar, solo se requiere la IP de la PC que se desea manipular, en este caso es la IP de la computadora del

Sistema Administrador y el nombre de usuario de la misma. Esta aplicación nos permite manipular el Sistema Administrador como si se estuviera sentado frente a ella, dándole el servicio de conexión remota al sistema realizado.

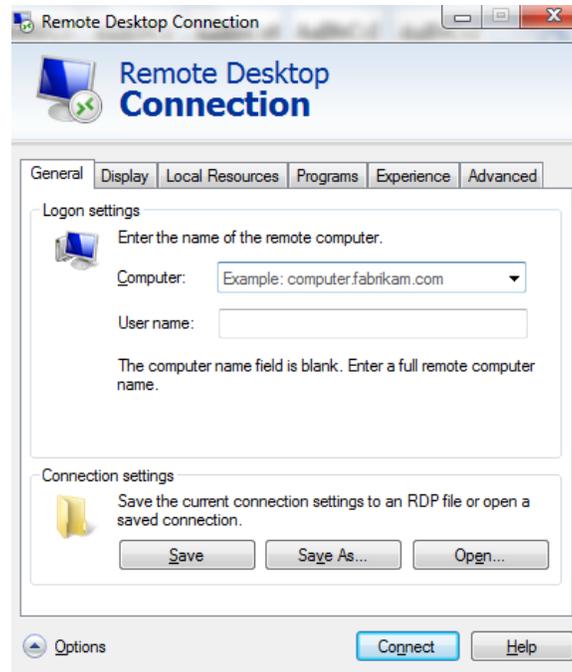


Figura 43. Aplicación para la conexión remota.

4.3. Resumen del Capítulo

En este capítulo se implementó el hardware del sistema, lo que concluye la integración del hardware y software del sistema *Smart Home* propuesto en esta tesis. En el capítulo 5 se inicia la etapa de pruebas del sistema para la evaluación de la misma a partir de los resultados.

Capítulo 5

Pruebas y Resultados

Después de la integración de las partes del sistema, se prosigue a iniciar la etapa de pruebas. En este capítulo se realizan las pruebas del sistema para evaluar su funcionalidad y confiabilidad a partir de los resultados. Con esta etapa se concluye el desarrollo del sistema *Smart Home* planteado en esta tesis, y en el Capítulo 6 se dan las conclusiones finales del sistema y se mencionan algunos trabajos futuros para la continuidad de esta investigación.

5.1. Pruebas del Sistema

Las pruebas realizadas al sistema se dividen en dos partes: la primera, se enfoca en la funcionalidad del Sistema Administrador, principalmente en la herramienta CASE la cual proporciona la escalabilidad y configurabilidad del sistema, la segunda, verifica la comunicación de los elementos del sistema y el monitoreo y control de los sensores y actuadores del sistema.

5.1.1. Pruebas y Resultados del Sistema Administrador

El Sistema Administrador es la parte principal del sistema *Smart Home*. La función del Sistema Administrador es realizar los procesos de verificación y análisis de los datos proporcionados por las partes que integran el sistema y la realización de las demandas del usuario, estas demandas involucran el monitoreo y control de todo el sistema, la configurabilidad y escalabilidad del mismo.

A continuación se describe cada una de las partes del Sistema Administrador, posteriormente se muestran las pruebas realizadas y resultados que esta emitió.

- **Interfaz principal:** Es la pantalla inicial del Sistema Administrador, en ella se muestran todos los menús, Panel de módulos y Panel de alertas (véase Fig. 44).
- **Menú de servicios:** Este menú contiene los servicios que brinda el sistema, el control de luces, control de persianas, monitoreo de presencia y monitoreo de humo. Está integrado por cuatro botones, donde al dar *clic* en uno de ellos se despliegan los módulos creados de ese servicio en el panel de módulos de ese servicio.
- **Menú CASE:** Este menú permite la escalabilidad y configurabilidad del sistema. Está integrada por tres botones, los cuales dan las opciones de agregar, modificar y quitar módulos. Al dar *clic* en uno de ellos se despliega un formulario en donde se insertan los datos requeridos para la creación, modificación o eliminación del módulo elegido.
- **Panel de módulos:** Cada servicio tiene su panel de módulos, en este se despliegan los módulos creados en ese servicio. El panel de módulos se visualiza al dar *clic* en un botón del menú de servicios.
- **Panel de alertas:** En este panel se visualizan todos los mensajes de alertas del sistema. Se podrá configurar si se desean alertas del sistema en cada elemento de cada módulo de los servicios de presencia y humo, con la finalidad de tener un servicio personalizado en cada módulo de servicio, el cual percibe los estados de cada nodo de presencia o humo. Se observarán mensajes de alertas cuando haya alguna presencia inusual o cuando se detecte presencia de humo en la casa, siempre y cuando esté configurado el sistema para realizar esta tarea.

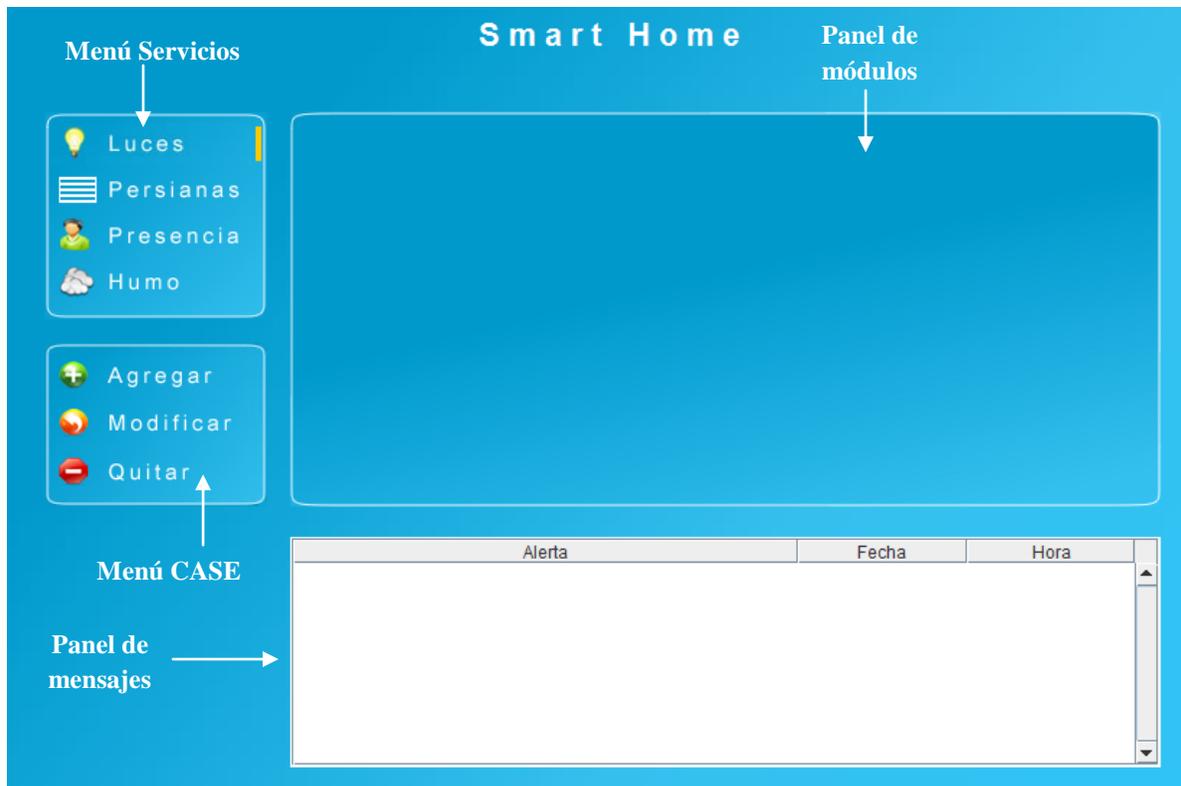


Figura 44. Interfaz principal diseñada.

Al iniciar el Sistema Administrador se muestra la interfaz principal de la Fig.44, el panel de módulos de cada servicio no contienen módulos, el usuario deberá crear los módulos requeridos en el sistema, para esto se hace uso del menú CASE.

5.1.2. Menú CASE

El menú CASE es la contribución principal del proyecto. Este menú permite agregar, modificar y quitar módulos, lo que le da al sistema la opción de escalabilidad y configurabilidad a las particularidades del usuario dando como resultado un sistema personalizable. Se realizaron pruebas en las tres opciones: Agregar Módulo, Modificar Módulo y Quitar Módulo, las cuales se explican a continuación.

5.1.3. Agregar Módulo

Para crear un módulo de servicio es necesario dar *clic* en el botón de agregar del menú CASE, este evento desplegará el formulario de la Fig. 45, en este se deben insertar los datos requeridos para la creación del módulo.

La parte de las características de cada elemento del módulo, involucra tres opciones: 1) La opción de nivel se refiere a si se desea tener un control en la intensidad de la luz o si se eligió

persianas un control de posición. En caso de elegir “No” solo tendrá control de apagado/encendido o en caso del servicio de persianas de abrir/cerrar. 2) La opción de programar se refiere si se desea tener la opción de programar el elemento indicado, esto permite visualizar una tabla donde podrá programarse las tareas, en esta tabla se ingresan los datos necesarios para realizar la tarea (elemento, nivel, fecha, hora), si se elige “No” esta opción no aparecerá. 3) La opción de activo se refiere a que si el elemento tendrá la opción de poderse activar o desactivar en algún momento, si se elige “No” esta opción no se visualizará.

Al dar *click* en aceptar, automáticamente el sistema crea la interfaz del módulo deseado y crea los registros de la misma en la base de datos. En la base de datos se guardan todo los datos que se insertaron en el formulario, con el fin que la interfaz permanezca igual ante fallos como falta de energía eléctrica, o cierres inesperados del sistema.

Para ejemplificar, se crea un módulo del servicio de luces (véase Fig. 45), el módulo se etiqueta como sala, el identificador (id) del Arduino con el que se va a comunicar es A, el número de elementos (luces) que tendrá el módulo son cinco, y las características que tendrán cada elemento se aprecia en la Fig. 45. Los resultados de la interfaz y los registros de la base de datos se muestran en las Figs. 46-50.

Agregar Módulo

Servicio

Luces Persianas Presencia Humo

Luces

Nombre Módulo: Id Módulo:

Cantidad Elementos:

Elemento	Nivel	Programar	Activo
1	<input type="text" value="Si"/>	<input type="text" value="Si"/>	<input type="text" value="Si"/>
2	<input type="text" value="Si"/>	<input type="text" value="Si"/>	<input type="text" value="Si"/>
3	<input type="text" value="Si"/>	<input type="text" value="Si"/>	<input type="text" value="Si"/>
4	<input type="text" value="Si"/>	<input type="text" value="Si"/>	<input type="text" value="Si"/>
5	<input type="text" value="Si"/>	<input type="text" value="Si"/>	<input type="text" value="Si"/>

Aceptar Cancelar

Figura 45. Formulario agregar.

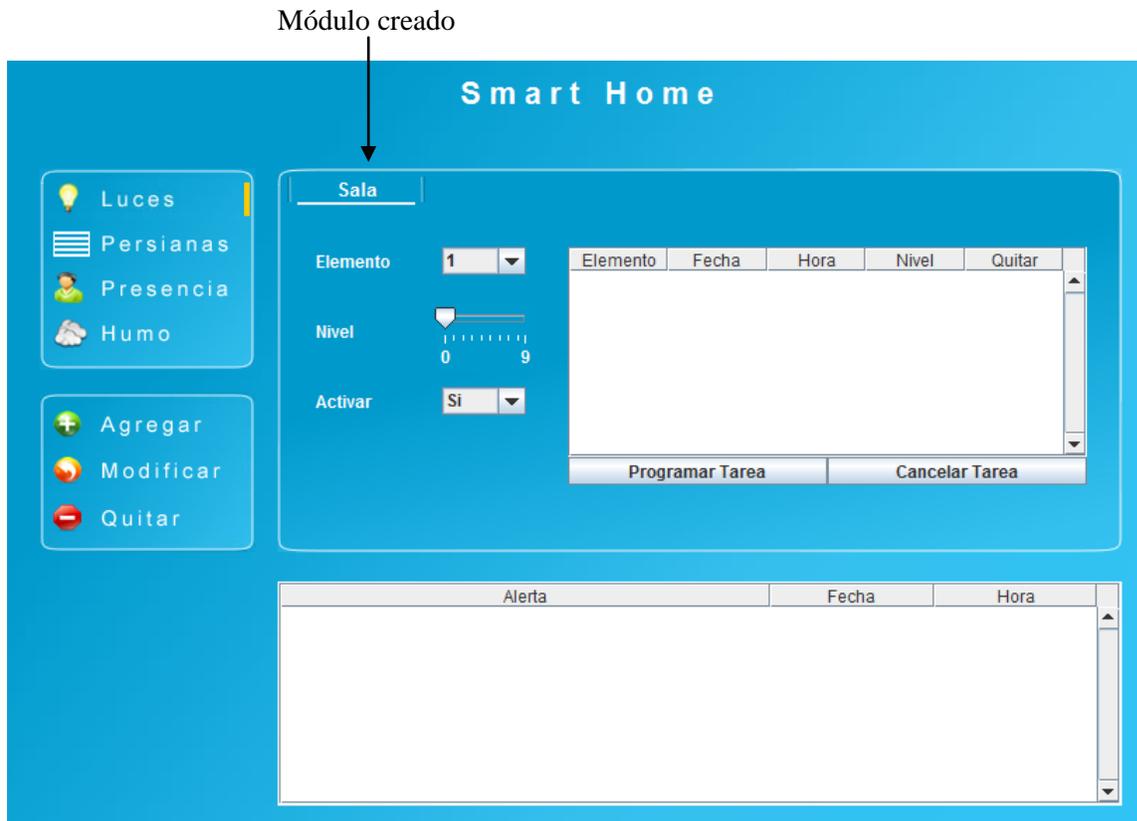


Figura 46. Creación de un módulo de servicio de luces.

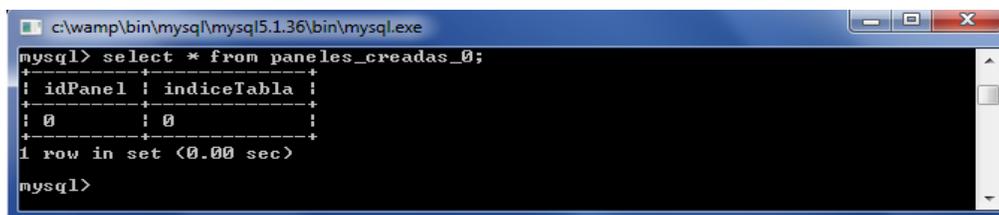


Figura 47. Registro del panel creado.

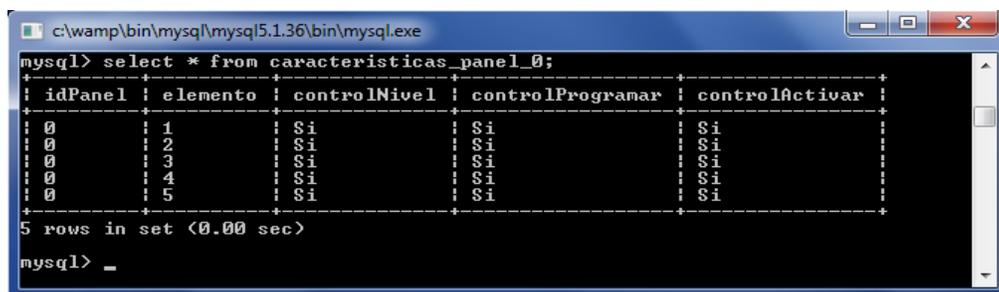
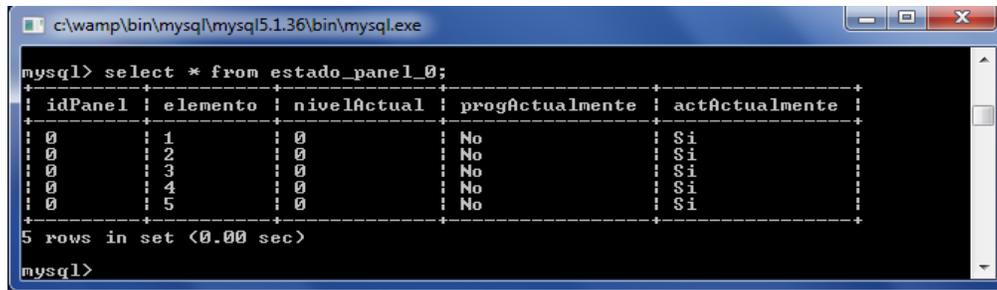


Figura 48. Registros de las características del módulo.



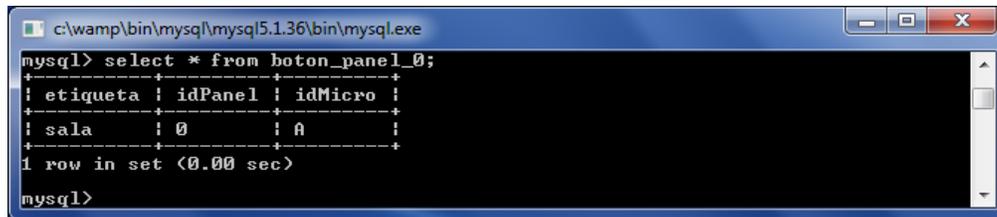
```

mysql> select * from estado_panel_0;
+----+-----+-----+-----+-----+
| idPanel | elemento | nivelActual | progActualmente | actActualmente |
+----+-----+-----+-----+-----+
| 0       | 1       | 0         | No              | Si             |
| 0       | 2       | 0         | No              | Si             |
| 0       | 3       | 0         | No              | Si             |
| 0       | 4       | 0         | No              | Si             |
| 0       | 5       | 0         | No              | Si             |
+----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql>

```

Figura 49. Registros de los estados de cada elemento del módulo.



```

mysql> select * from boton_panel_0;
+-----+-----+-----+
| etiqueta | idPanel | idMicro |
+-----+-----+-----+
| sala    | 0       | A       |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql>

```

Figura 50. Registro del botón del módulo.

Se realizaron pruebas de creación de módulos en todos los servicios disponibles, además de variar las características de cada módulo creado, con el fin de verificar la creación correcta de los módulos y los registros en la base de datos.

Los resultados de todas las pruebas fueron favorables en un 100%, no hubo inconvenientes al realizar cada prueba, por lo que se concluye que la funcionalidad de la opción de Agregar Módulo es confiable.

5.1.4. Modificar Módulo

Los registros de las características del módulo permanecen inalterados hasta que el usuario desee cambiarlos con la opción de modificar del menú CASE. Para modificar la interfaz de un módulo específico se da *clic* en el botón de modificar, el evento despliega el formulario de la Fig. 51, en esta se visualizan las características originales del módulo elegido, el usuario debe cambiar las características a las deseadas y dar *clic* en aceptar para la modificación del módulo.

La modificación del módulo se visualiza en la interfaz del módulo, además de realizarse una actualización de los registros de las características de ese módulo en la base de datos.

Modificar Módulo

Servicio

Luces Persianas Presencia Humo

Luces

Nombre Módulo: Sala

Elemento	Nivel	Programar	Activo	Quitar
1	No	No	Si	<input type="checkbox"/>
2	Si	Si	Si	<input type="checkbox"/>
3	Si	Si	Si	<input type="checkbox"/>
4	Si	Si	Si	<input type="checkbox"/>
5	Si	Si	Si	<input type="checkbox"/>

Agregar Quitar

Aceptar Cancelar

Figura 51. Formulario modificar.

Para ejemplificar, se modifican las características del módulo creado en la sección anterior, se modifican las características de nivel y programar del elemento uno y el resultado de la interfaz y de la base de datos son mostradas en las Figs. 52-53.

Se observa en la Fig. 52 que no se tiene la opción de intensidad y no aparece la tabla de registros de programación de tareas en la interfaz, y en la Fig. 53 se observa que el registro del elemento 1 se modificó en el campo controlNivel y controlProgramar que son las características de nivel y programar respectivamente.

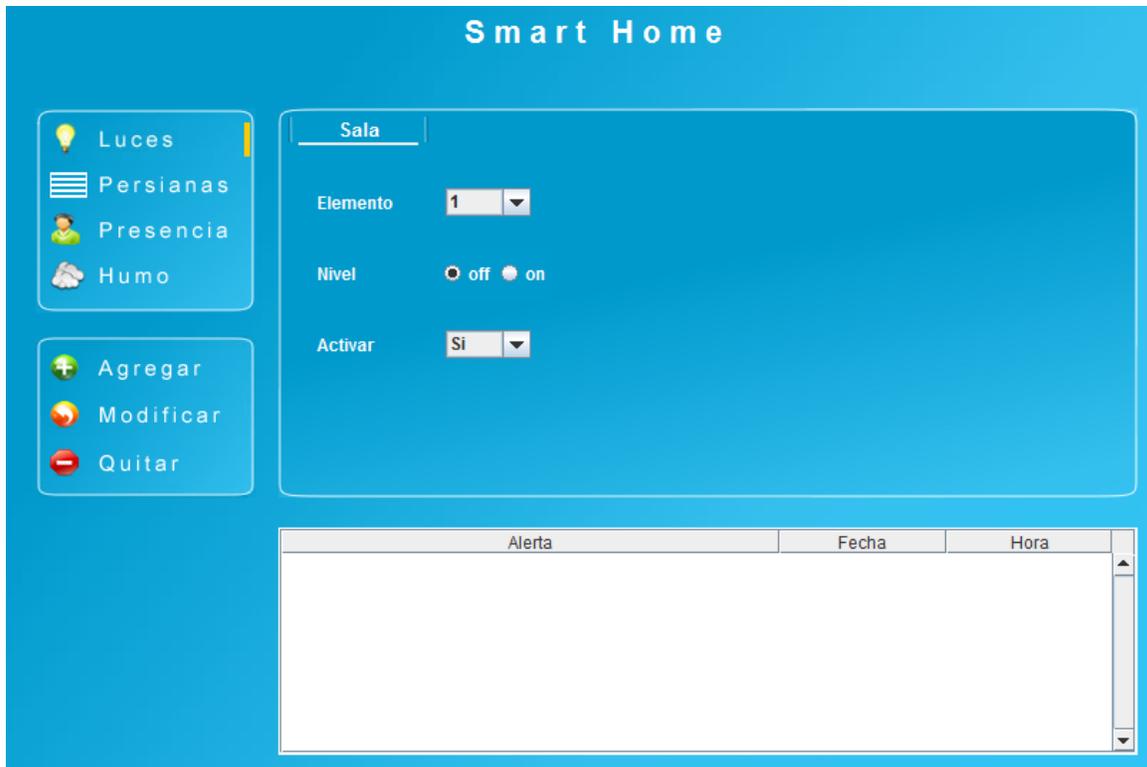


Figura 52. Modificación de la interfaz.

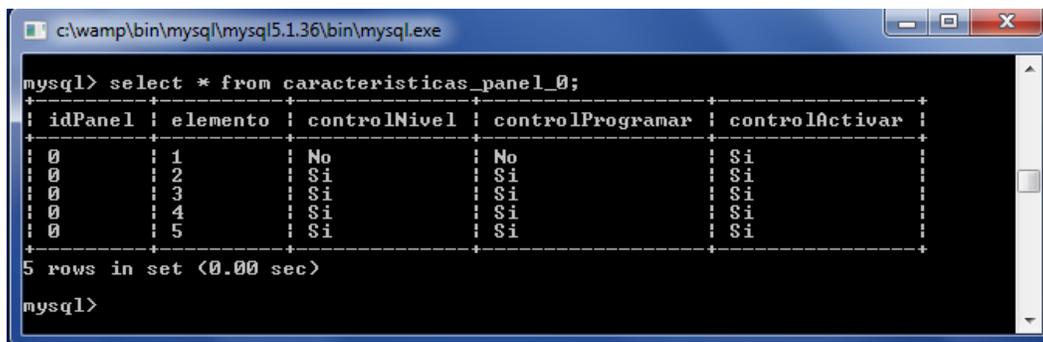


Figura 53. Registro modificado.

Se realizaron varias pruebas de modificación de módulos en todos los servicios disponibles, se cambiaron las características de cada módulo creado, se verificaron los cambios en la interfaz y las actualizaciones en los registros en la base de datos.

Los resultados de todas las pruebas fueron favorables en un 100%, no hubo inconvenientes al realizar cada prueba, por lo que se concluye que la funcionalidad de la opción de Modificar Módulo es confiable.

5.1.5. Quitar Módulo

Para eliminar un módulo específico se hace uso de la opción quitar del menú CASE, al dar *clic* en el botón quitar, este evento despliega el formulario de la Fig. 54. En este formulario se visualizan los módulos creados de cada servicio, el usuario debe elegir el módulo a eliminar y dar *clic* en eliminar para la eliminación del módulo.

Para ejemplificar, se elimina el módulo creado anteriormente, para esto se elige en el formulario el servicio y el módulo a eliminar (véase Fig. 54), posteriormente se dar *clic* en el botón eliminar, esta desplegará un mensaje de confirmación (véase Fig. 55) donde se debe elegir si, el proceso eliminará el módulo de la interfaz y todos los registros del módulo de la base de datos (véase Fig. 56).



Quitar Módulo

Servicio

Luces Persianas Presencia Humo

Luces

Nombre Módulo: Sala

Eliminar Cancelar

Figura 54. Formulario quitar.

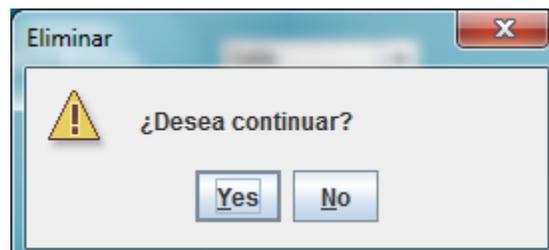
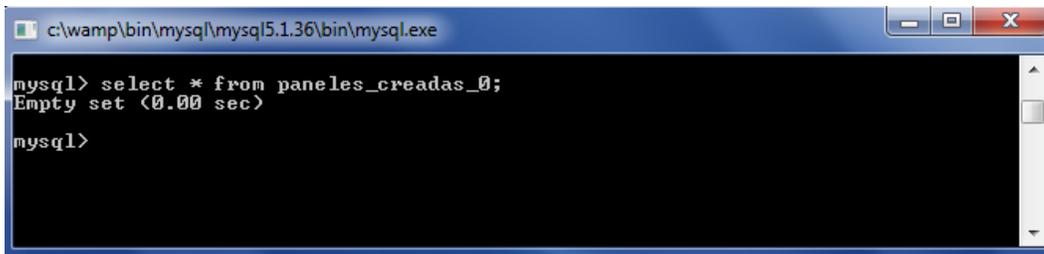


Figura 55. Mensaje de confirmación.



```
c:\wamp\bin\mysql\mysql5.1.36\bin\mysql.exe
mysql> select * from paneles_creadas_0;
Empty set (0.00 sec)
mysql>
```

Figura 56. Registro eliminado.

Se realizaron pruebas de eliminación de módulos en todos los servicios disponibles, se verificó que las tareas programadas de los módulos fueran eliminados junto con los registros de la base de datos del módulo eliminado.

Los resultados de todas las pruebas fueron favorables en un 100%, no hubo inconvenientes al realizar cada prueba, por lo que se concluye que la funcionalidad de la opción de Quitar Módulo es confiable.

5.1.6. Pruebas y Resultados de Monitoreo y Control

Se realizaron pruebas de ejecución del sistema completo, estas pruebas involucran el Sistema Administrador, la comunicación inalámbrica, el controlador, los sensores y los actuadores. Las pruebas realizadas son las siguientes:

- Control de luces
- Control de persianas
- Monitoreo de presencia
- Monitoreo de humo
- Programación de tareas

Cada prueba realizada con los resultados se explica en las siguientes secciones.

5.1.7. Control de Luces

Cada módulo puede controlar un máximo de cinco luces, el control se realiza mediante encendido/apagado o por medio de niveles de intensidad. Por lo tanto, se realizaron pruebas en diferente modalidad de control en todas las luces. Se controlaron cinco luces al mismo tiempo de manera eficiente, y se probó el control de estas en seis módulos diferentes.

Cada modificación de estado de las luces se registra en la base de datos como respaldo de futuros fallos en el sistema. Las actualizaciones de los registros de la base de datos se realizaron correctamente en todas las pruebas realizadas.

Los resultados de todas las pruebas fueron favorables en un 100%, no hubo pérdidas de paquetes, por lo que se concluye que la funcionalidad de control de luces es confiable.

5.1.8. Control de Persianas

Cada módulo puede controlar un máximo de cinco persianas, el control se realiza mediante encendido/apagado o por medio de niveles de posición. Por lo tanto, se realizaron pruebas en diferente modalidad de control en todas las persianas. Se controlaron cinco persianas al mismo tiempo de manera eficiente, y se probó el control de estas en seis módulos diferentes.

Cada modificación de estado de las persianas se registra en la base de datos como respaldo de futuros fallos en el sistema. Las actualizaciones de los registros de la base de datos se realizaron correctamente en todas las pruebas realizadas.

Los resultados de todas las pruebas fueron favorables en un 100%, no hubo pérdidas de paquetes, por lo que se concluye que la funcionalidad de control de persianas es confiable.

5.1.9. Monitoreo de Presencia

Cada módulo puede monitorear un máximo de cinco sensores de presencia, el monitoreo se visualiza en el Sistema Administrador mediante variación de niveles, estos niveles son el resultado de la obtención de las señales emitidos por el nodo que contiene los sensores de presencia. Se monitorearon cinco sensores de presencia al mismo tiempo de manera eficiente, y se probó el monitoreo en seis módulos diferentes.

Las actividades de los sensores no se registran en la base de datos.

Los resultados de todas las pruebas fueron favorables en un 100%, no hubo pérdidas de paquetes, por lo que se concluye que la funcionalidad de monitoreo de presencia es confiable.

5.1.10. Monitoreo de Humo

Cada módulo puede monitorear un máximo de cinco sensores de humo, el monitoreo se visualiza en el Sistema Administrador mediante variación de niveles, estos niveles son el resultado de la

obtención de las señales emitidas por el nodo que contiene los sensores de humo. Se monitorearon cinco sensores de humo al mismo tiempo de manera eficiente, y se verificó el monitoreo en seis módulos diferentes.

Las actividades de los sensores no se registran en la base de datos.

Los resultados de todas las pruebas fueron favorables en un 100%, no hubo pérdidas de paquetes, por lo que se concluye que la funcionalidad de monitoreo de humo es confiable.

5.1.11. Programación de Tareas

La programación de tareas es una opción para los servicios de luces y persianas. Al programar una tarea, se guarda el registro en la base de datos, este registro tiene los siguientes atributos:

identificador, idMicro, idPanel, elemento, nivel, dia, mes, año, hora, minuto lo que permite tener la información necesaria para realizar la tarea, idMicro, es el nodo donde se envía el paquete con la instrucción, idPanel, es el panel en el que se va actualizar el objeto de control (*slider*), elemento, luz o persiana en el que se realiza la tarea, (dia, mes, año) fecha y (hora, minuto) hora en el que realiza la tarea.

Se realizaron pruebas al programar veinte tareas en diferentes módulos de servicio. Las tareas programadas fueron realizadas de manera correcta.

Los resultados de todas las pruebas fueron favorables en un 100%, no hubo pérdidas de paquetes, por lo que se concluye que la funcionalidad de Programación de Tareas es confiable.

5.2. Aplicación de Instalación del Sistema

Al término de las pruebas y con los resultados favorables, se prosigue a realizar una aplicación ejecutable que permita la instalación del sistema. El ejecutable está integrado por un conjunto de aplicaciones que facilitan la instalación (véase Fig. 57).

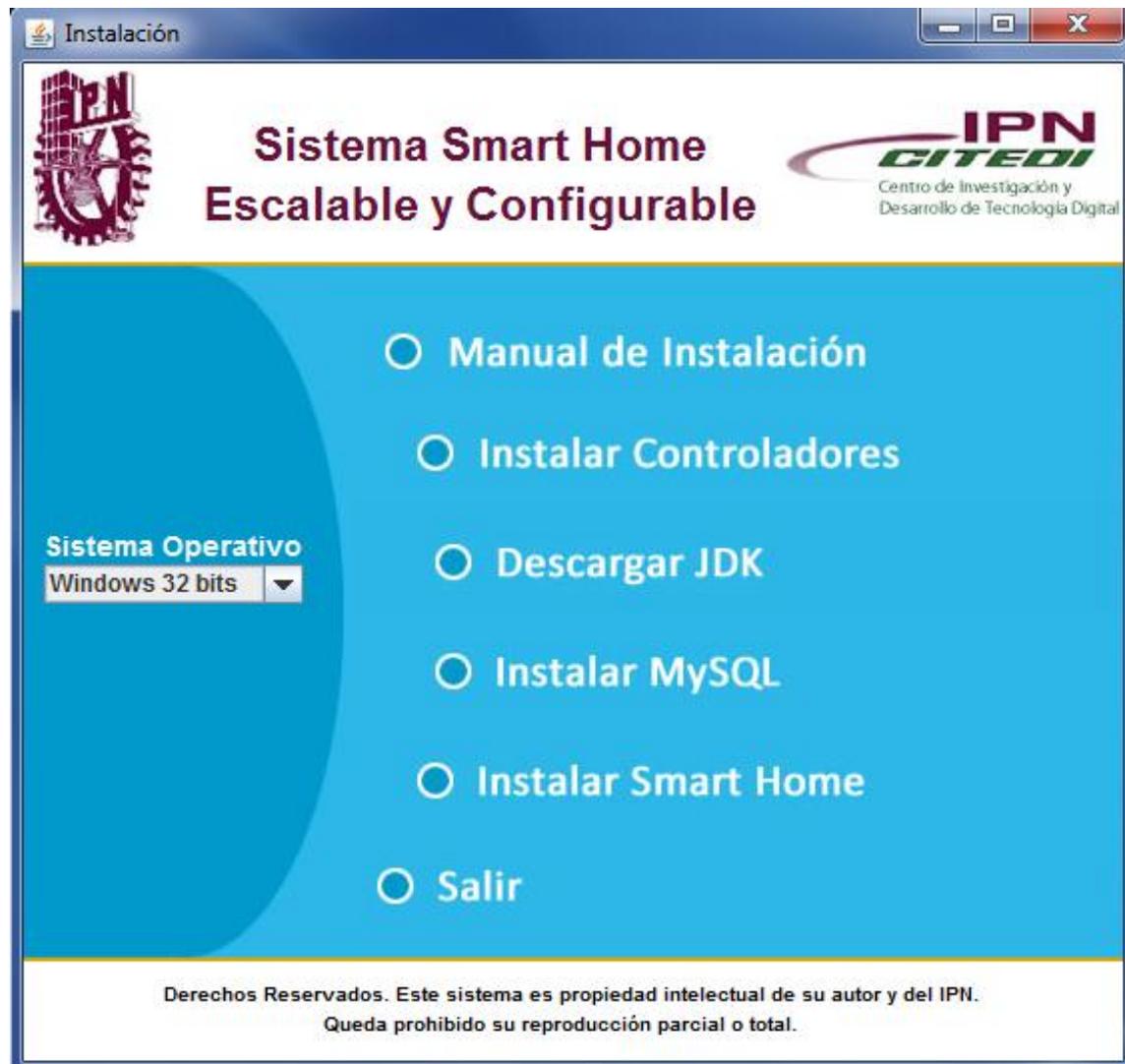


Figura 57. Interfaz de instalación.

Las opciones del menú son las siguientes:

- Manual de instalación. En este apartado se encuentra un manual de ayuda para instalar el sistema.
- Sistema Operativo. Esta opción permite elegir el sistema operativo en ejecución, esto permite instalar la versión adecuada del sistema *Smart Home*.
- Instalar Controladores. Este apartado permite la instalación de los controladores necesarios para el funcionamiento del sistema a instalar.
- Descargar JDK. Este apartado vincula a la página ORACLE para la descarga de JDK (*Java Development Kit*) que permite la ejecución de aplicaciones Java.

- Instalar *MySQL*. Este apartado permite la instalación de *MySQL*, que es el motor de la base de datos.
- Instalar *Smart Home*. Este apartado permite la instalación del sistema *Smart Home*.
- Salir. Este apartado permite salir de la instalación.

5.3. Resumen del Capítulo

Después de realizar todas las pruebas pertinentes al sistema completo, se obtuvieron resultados que demuestran que el sistema *Smart Home* desarrollado en esta tesis funciona de manera óptima, eficiente y confiable.

En el siguiente capítulo se expresan las conclusiones finales de este proyecto de tesis basándose en las experiencias, estudios y resultados obtenidos en el transcurso del desarrollo de este proyecto.

Capítulo 6

Conclusiones y Trabajos Futuros

6.1. Conclusiones Generales

El área de investigación *Smart Home* está teniendo un auge importante en la actualidad, debido a la diversidad de áreas de la ciencia que esta involucra. Los sistemas *Smart Homes* no deben considerarse solo como una tecnología, sino como una necesidad de la sociedad, debido a que solucionan diversos problemas importantes, como la vigilancia de personas de la tercera edad, monitoreo de salud, reconocimiento de situaciones de emergencia, entre otros.

Uno de los objetivos de los sistemas *Smart Homes* es realizar una convergencia entre varias tecnologías de software, hardware y comunicaciones, esto ha llevado a que varias investigaciones desarrollen nuevas tecnologías como alternativas que permitan la compatibilidad entre productos.

Por otro lado, el sistema propuesto en este trabajo de tesis plantea una alternativa de sistemas *Smart Homes* reconfigurables y escalables a partir de los requerimientos del usuario, con el objetivo de que los investigadores se interesen en el desarrollo de herramientas que permiten la automatización de etapas del desarrollo de un sistema. De esta manera se evitan posibles errores en el desarrollo de la codificación, dando como resultado más tiempo para el desarrollo de las etapas de análisis y diseño de la misma.

El sistema desarrollado está diseñado para poder expandir el número de servicios en un futuro, esto garantiza una larga vida del software; además, se pretende que el sistema desarrollado sea un prototipo que sirva para investigaciones futuras, dejando el código abierto para mejoras en ella.

Los elementos de hardware que se requieren para este sistema son de bajo costo, lo que garantiza la factibilidad en la implementación en viviendas y en el desarrollo de investigación.

Los resultados obtenidos del sistema *Smart Home* desarrollado demuestran la robustez de la escalabilidad y reconfiguración de la misma, reflejando aportaciones importantes en el área de *Smart Home*. Por lo tanto, se concluye que el sistema desarrollado en este trabajo de tesis maximiza los beneficios de los sistemas *Smart Homes* y reduce los costos de mantenimiento de los mismos.

6.2. Comentarios y Recomendaciones para Trabajos Futuros

Al finalizar el sistema *Smart Home* y realizar las pruebas finales estas arrojaron resultados favorables, el sistema es estable y confiable, dando expectativas excelentes para el autor de esta tesis, sin embargo, debido al corto tiempo de la investigación, esta abarcó aspectos estratégicos en el desarrollo del sistema, por lo que existen trabajos que pueden continuar con esta investigación. El sistema *Smart Home* desarrollado está diseñado para poder realizar modificaciones en ella, algunos trabajos futuros que pueden considerarse para este proyecto son:

Nuevos servicios: El sistema cuenta con la gestión de cuatro servicios (control de luces y persianas, monitoreo de presencia y humo) y el código de la misma es accesible para su modificación. Se pueden incrementar estos servicios con otros tales como el monitoreo de consumo de energía eléctrica/agua con el objetivo del ahorro de la misma, autenticación de usuarios mediante mediciones biométricas.

Análisis de los datos de sensores: El sistema no registra las actividades de los sensores, por lo que se debe rediseñar la base de datos del sistema para poder registrar las actividades de los sensores y posteriormente realizar el análisis de estos datos.

Seguridad del sistema: El sistema está diseñado para su funcionamiento en un entorno conocido, sin embargo, para que el sistema pueda ser implementado en una vivienda, esta debe de incrementar su seguridad, como por ejemplo la encriptación de los datos que se envían entre los elementos del sistema y el administrador o la identificación de los usuarios para asignar prioridades para el uso del sistema.

Diseño del hardware: El sistema desarrollado es un prototipo por lo que no puede ser implementado en una vivienda, para esto, se requiere elaborar un diseño de hardware más sofisticado, acorde a los actuadores que se instalan en una casa, por ejemplo un diseño de etapa de potencia para el control de luces y motores de dimensiones reales, o el diseño específico del circuito del microcontrolador de los nodos del sistema para reducir costos de las tarjetas electrónicas utilizadas en este proyecto.

7. REFERENCIAS BIBLIOGRÁFICAS

- [1] R. Lutolf. *Smart Home Concept and the Integration of Energy Meters Into a Home Based System*. IEEE Seventh International Conference on Metering Apparatus and Tariffs for Electricity Supply. 1992. 17-19 Nov 1992. Páginas 277 – 278.
- [2] L. Jiang, D.-Y. Liu, B. Yang. *SmartHome Research*. IEEE International Conference on Machine Learning and Cybernetics, 2004. 26-29 Aug. 2004. Páginas 659 - 663 vol.2.
- [3] J. H. Park, C-H. Hsu, H.R. Arabnia. *Intelligent Systems and Smart Homes*. Springer Volume 11, Numero 5 / November 2009.
- [4] V. Riquebourg, D. Menga, D. Durand, B. Marhic, D. Laurent, C. Logé. *The Smart Home Concept: Our Immediate Future*. IEEE International Conference on E-Learning in Industrial Electronics 2006. 18-20 Diciembre 2006. Páginas 23 – 28. Hammamet.
- [5] R. Babbitt. *Information Privacy Management in Smart Home Environments: Modeling, Verification, and Implementation*. IEEE COMPSAC '06. 30th Annual International Computer Software and Applications. 17-21 Sept. 2006. Páginas 344 – 346. Chicago, IL.
- [6] D. J. Cook, K. S. Das. *How Smart are Our Environments? An Updated Look at the State of the Art*. Ubiquitous computing and mobile. Volumen 3 Número 2, marzo, 2007.
- [7] P. Nektarios, M. Apostolos, E. Dimitrios, I. Karras, I. Liverezas. *A Connected Home Platform and Development Framework for Smart Home Control Applications*. IEEE INDIN 2009, 7th International Conference on Industrial Informatics. 23-26 June 2009. Páginas 402 – 409. Cardiff, Pais de Gales.
- [8] A. Venkatesh, *Smart Home Concepts: Current Trends*. Revista I.T. in the Home, Center for Research on Information Technology and Organizations, UC Irvine. 07-01-2003.
- [9] R. Babbitt. *Information Privacy Management in Smart Home Environments: Modeling, Verification, and Implementation*. IEEE COMPSAC '06. 30th Annual International.on Computer Software and Applications. 17-21 Sept. 2006. Páginas 344 – 346. Chicago, IL.
- [10] A. Serna, H. Pigot, V. Rialle. *Modeling the progression of Alzheimer's disease for cognitive assistance in Smart Homes*. Springer. User Modeling and User-Adapted Interaction Journal. Volume 17 Issue 4, Septiembre 2007.

-
- [11] S. K. Min, M. K. Kyung, C. K. Hee. *A Questionnaire Study for the Design of Smart Home for the Elderly*. IEEE HEALTHCOM 2006. 8th International Conference on e-Health Networking, Applications and Services. 17-19 Agosto. 2006. Páginas 265 - 268.
- [12] D. J. Cook, A. Crandall, G. Singla, B. Thomas. *Detection of Social Interaction in Smart Spaces*. Cybernetics and Systems. Volumen 41, número 2, 2010.
- [13] R. Larry. *Project Oxygen: Pervasive, Human-Centric Computing – An Initial Experience*. Springer CAiSE '01 13th International Conference on Advanced Information Systems Engineering. 2001.
- [14] M. Ritala, T. Tieranta, J. Vanhala. *Context Aware User Interface System for Smart Home Control*. HOIT Conference, Irvine, California. Estados Unidos. 2003.
- [15] F. Zuo, H. N. Peter. *Real-time Face Recognition for Smart Home Applications*. IEEE International Conference on Consumer Electronics (ICCE 2005). Volumen 51. Páginas 183-190. Las Vegas, Estados Unidos.
- [16] P. T. Yu, W. H. Jun, T. L. Cheng, Y. C. Chun. *Building a Remote Supervisory Control Network System for Smart Home Applications*. IEEE SMC '06. IEEE International Conference on Systems, Man and Cybernetics. 8-11 Oct. 2006. Páginas 1826 – 1830. Taipei.
- [17] L. Burnell, A. Sanchez, J. Priest, C. Hannon. *The Crescent Lab: A Smart Home Lab for Students*. IEEE. ENC '06 Seventh Mexican International Conference on Computer Science. Sept. 2006. Páginas 55 – 61. San Luis Potosi, México.
- [18] V. Jakkula, D. J. Cook. *Anomaly Detection Using Temporal Data Mining in a Smart Home Environment*. Methods of Information in Medicine (2008) Volumen: 47, Edición 1, Páginas 70-75.
- [19] L. X. Hoa Binh, M. Di Mascolo, A. Gouin, N. Noury. *Health Smart Home - Towards an Assistant Tool for Automatic Assessment of the Dependence of Elders*. IEEE EMBS 2007. 29th Annual International Conference on Engineering in Medicine and Biology Society. 22-26 Aug. 2007. Páginas 3806 – 3809. Lyon, Francia.
- [20] H. Sajid, Z. E. Senol, H. P. Jong. *Monitoring user activities in Smart Home environments*. Springer Volumen 11, Numero 5. Páginas 539-549. Noviembre 2009.
- [21] C. Chen, C. Pomalaza Ráez. *Design and Evaluation of a Wireless Body Sensor System for Smart Home Health Monitoring*. IEEE GLOBECOM 2009 Conference on Global

- Telecommunications. 30 Noviembre 2009 – 4 Diciembre 2009. Páginas 1 – 6. Honolulu, Hawai.
- [22] J. M. Hsu, I. R. Chang. *Design a Virtual Object Representing Human-Machine Interaction for Music Playback Control in Smart Home*. IEEE CISIS '09. International Conference on Complex, Intelligent and Software Intensive Systems. 16-19 Marzo 2009. Páginas 614 – 619. Fukuoka, Japón.
- [23] W. Zhenxing, W. Shutao, S. Linxiang, L. Zhongyuan. *The Analysis & Implementation Of Smart Home Control System*. IEEE ICIME '09. International Conference on Information Management and Engineering. 3-5 Abril 2009. Páginas 546 – 549. Kuala Lumpur, Malasia.
- [24] P. Rashidi, D. J. Cook, L. B. Holder, M. S. Edgecombe. *Discovering Activities to Recognize and Track in a Smart Environment*. IEEE. Transactions on Knowledge and Data Engineering. April 2011. Volumen 23 Edición 4. Páginas 527 – 539.
- [25] S. Taeshik, P. Yongsuk. *Implementation of RF4CE-based Wireless Auto Configuration Architecture for Ubiquitous Smart Home*. IEEE 2010 International Conference on Complex, Intelligent and Software Intensive Systems (CISIS). 15-18 Feb. 2010. Páginas 779 – 783. Cracovia, Polonia.
- [26] R. S. Pressman. *Ingeniería de Software, un Enfoque Práctico*. 2ª Edición. Ed. Mc Graw Hill. México. 2010.
- [27] B. Bruegge, A. H. Dutoit. *Ingeniería de Software Orientado a Objetos*. 1ª Edición. Prentice Hall. México. 2002.
- [28] I. Sommerville. *Ingeniería del Software*. 2ª Edición. Pearson Addison Wesley. España. 2005.
- [29] Departamento de Sistemas Informáticos y Computación. *Introducción a Herramientas CASE y System Architect*. Seminario Metodología y Tecnología de la Programación. Universidad Politécnica de Valencia. España. 2005.
- [30] Facultad de Informática. *Introducción a Herramientas CASE*. Seminario Metodología y Tecnología de la Programación. Universidad Politécnica de Valencia. España. 2004.
- [31] J. Schmuller. *Aprendiendo UML en 24 Horas*. 2ª Edición. Prentice Hall. México. 2003.
- [32] A. S. Silberschatz, H. F. Korth. Sundarshan. *Fundamentos de Bases de Datos*. 2ª Edición. Mc Graw Hill. España . 2002.

- [33] G. Choi, T. Kim. *UML CASE Tool Supporting OLE Compound Document*. IEEE Software Engineering Conference, 2004. 11th Asia-Pacific. 30 Nov.-3 Dic. 2004. Páginas 250 - 256.
- [34] H. M. Harmain, R. Gaizauskas. *CM-Builder: A Natural Language-Based CASE Tool for Object-Oriented Analysis*. Springer Automated Software Engineering. Volume 10. Edición 2, Abril 2003.
- [35] K. Maruyama, S. Yamamoto. *A CASE Tool Platform Using an XML Representation of Java Source Code*. IEEE International Workshop on. Source Code Analysis and Manipulation. Sept. 2004. Páginas 158 – 167. Chicago, IL.
- [36] M. Zapata, J. Chaverra. *Una Mirada Conceptual a la Generación Automática de Código*. Revista EIA, núm. 13, julio, 2010, páginas. 143-154. Escuela de Ingeniería de Antioquia. Colombia.
- [37] J. Kendall. K. Kendall. *Análisis y diseño de sistemas*. 7ª Edición. Ed. Pearson Educación. 2005. México.
- [38] P. J. Molina Moreno. *Especificación de interfaz de usuario: De los requisitos a la generación automática*. Tesis de Doctorado en Informática Universidad Politécnica de Valencia, 2003. Valencia, España.
- [39] J. Rumbaugh, I. Jacobson, G. Booch. *El lenguaje unificado de modelado, Manual de referencia*. 1ª Edición. Ed. Pearson Addison Wesley, 2000. Madrid, España.
- [40] R. C. Martin. *UML para programadores Java*. 1ª Edición. Ed. Pearson Prentice Hall. 2004. Madrid, España.
- [41] Arduino. En línea, disponible en: <http://www.arduino.cc/es/>. Acceso: Noviembre de 2011.