

**DISEÑO DE UN MODULO PARA EL CONTROL DE ILUMINACIÓN DE UN  
HARDWARE DOMOTICO A TRAVÉS DE UN TELEFONO CELULAR EN UN  
AMBIENTE DISTRIBUIDO**

**ANGELA VIVIANA RAMÍREZ ACEVEDO  
JULIAN AUGUSTO BOJACA MEDINA**



**UNIVERSIDAD EL BOSQUE  
FACULTAD DE INGENIERIA DE SISTEMAS  
AREA DE TELECOMUNICACIONES  
BOGOTÁ, D.C. DICIEMBRE DE 2007**

**DISEÑO DE UN MODULO PARA EL CONTROL DE ILUMINACIÓN DE UN  
HARDWARE DOMOTICO A TRAVÉS DE UN TELEFONO CELULAR EN UN  
AMBIENTE DISTRIBUIDO**

**Línea de Investigación en Telecomunicaciones**

**Angela Viviana Ramírez Acevedo  
Julián Augusto Bojacá Medina**

**Trabajo de grado para optar por el título de Ingeniero de Sistemas**

**Carlos Fernando Varela Pérez  
Director de Tesis**

**Nilson Valencia  
Asesoría Metodológica**

**Carlos Angarita  
Asesor de Domótica**

**UNIVERSIDAD EL BOSQUE  
FACULTAD DE INGENIERIA DE SISTEMAS  
BOGOTA D.C.  
2007**

La Universidad El Bosque, no se hace responsable de los conceptos emitidos por los investigadores en su trabajo, sólo velará por el rigor científico, metodológico y ético del mismo en aras de la búsqueda de la verdad y la justicia.

## **AGRADECIMIENTOS**

Luego de más de un año de dedicación a este proyecto son muchas las personas a las cuales queremos agradecer. Fueron muchos días de esfuerzo, de investigación, de desarrollo y de pruebas; en muchas ocasiones pensamos que era imposible cumplir con los tiempos de entrega, incluso aún, cuando el mismo día de la sustentación todo falló. A pesar de esto, lo logramos, y queremos expresar infinita gratitud a las personas que mencionamos a continuación.

Primero que todo a Dios, por permitirnos comenzar y terminar una carrera universitaria, por darnos los recursos económicos, la fuerza y la inteligencia para llegar hasta acá.

A los profesores de la Facultad de Sistemas de la Universidad El Bosque, los de matemáticas, los de física, los de la línea profesional, los de la línea de administración, incluso los de la línea de humanidades y electivas, por su formación, por su tiempo, por sus enseñanzas. De cada uno de ellos nos llevamos un grato recuerdo para nuestra vida personal y profesional.

Al decano, secretario académico, directores de las líneas de investigación, muchas gracias por su esfuerzo constante para mejorar la facultad día a día y hacerla una de las mejores de la Universidad.

A Carlos Varela, nuestro director de proyecto por su apoyo.

A Carlos Angarita, por su asesoría en el hardware domótico, por su visión, su esmero, su actitud servicial y su esmero por sacar adelante un proyecto que comenzó con tan solo una idea y que día a día crece más y más.

A nuestros compañeros de clase, nuestra generación, por tantos momentos compartidos.

A todos los que toman este documento como una fuente de investigación para sus proyectos de grado, adelante!, es una etapa inolvidable en la vida; si necesitan más información o si tienen preguntas a cerca de esta investigación, pueden escribirnos a [jbjuasmajo@gmail.com](mailto:jbjuasmajo@gmail.com)

*Con todo nuestro corazón a nuestros padres, por cada palabra de aliento, por sus sacrificios, por su confianza, a nuestra familia por el constante apoyo y de manera muy especial a nuestra Virgencita por llevarnos de la mano siempre y darnos la fe y el aliento en los momentos de angustia.*

## TABLA DE CONTENIDO

pág.

### INTRODUCCION

<b>1.</b>	<b>PLANTEAMIENTO DEL PROBLEMA</b>	<b>1</b>
1.1	DESCRIPCION DEL PROBLEMA	1
1.2	FORMULACION	2
1.3	DELIMITACION	2
1.4	ESTADO DEL ARTE	3
1.5	JUSTIFICACION	11
1.6	OBJETIVOS	12
1.6.1	Objetivo General	12
1.6.2	Objetivos Específicos	12
<b>2.</b>	<b>MARCO TEORICO</b>	<b>13</b>
2.1	ANTECEDENTES HISTORICOS	13
2.2	MARCO CONCEPTUAL	15
2.2.1	Domótica	15
2.2.2	Sistemas Distribuidos	20
2.2.3	Redes Inalámbricas	21
2.2.4	Tecnología Celular	23
2.2.5	Tecnologías para el desarrollo de Aplicaciones Móviles	30
2.3	MARCO CONTEXTUAL	39
<b>3.</b>	<b>REQUERIMIENTOS</b>	<b>40</b>
3.1	RESUMEN EJECUTIVO	40
3.2	REQUERIMIENTOS FUNCIONALES	41
3.3	REQUERIMIENTOS NO FUNCIONALES	45
<b>4.</b>	<b>METODOLOGÍA</b>	<b>47</b>
<b>5.</b>	<b>DISEÑO GLOBAL</b>	<b>49</b>
5.1	ARQUITECTURA DE HARDWARE	50
5.2	ARQUITECTURA DE SOFTWARE	51
5.2.1	Página Web	51
5.2.2	Midlet	52
5.2.3	Diagrama de Software	54
5.3	ALTERNATIVAS DE SOLUCION	55
<b>6.</b>	<b>DISEÑO DETALLADO</b>	<b>57</b>
6.1	DIAGRAMA DE CLASES SIN PATRONES	57
6.2	DIAGRAMA DE CLASES CON PATRONES	58

<b>7.</b>	<b>IMPLEMENTACION</b>	<b>69</b>
<b>8.</b>	<b>PRUEBAS</b>	<b>75</b>
8.1	PRUEBAS UNITARIAS	75
8.2	PRUEBAS SOBRE EL HARDWARE DOMOTICO	78
8.3	PRUEBAS SOBRE EL MIDLET Y EL SERVLET DEL MIDLET	79
8.4	PRUEBAS SOBRE LA APLICACIÓN WEB Y EL SERVLET	80
	<b>CONCLUSIONES</b>	<b>82</b>
	<b>RECOMENDACIONES</b>	<b>84</b>
	<b>PROYECCIONES</b>	<b>85</b>
	<b>BIBLIOGRAFÍA</b>	<b>86</b>
	<b>APÉNDICE A-DOCUMENTACIÓN CASOS DE USO</b>	<b>90</b>
	<b>APÉNDICE B-MANUALES PÁGINA WEB Y MIDLET</b>	<b>109</b>
	<b>APENDICE C-HARDWARE DOMÓTICO</b>	<b>149</b>

## LISTA DE FIGURAS

	pág.
Figura 1. Entorno gráfico de HomeTalk	4
Figura 2. Controlador GSM para envío y recepción de instrucciones a través de mensajes SMS	5
Figura 3. Aplicación para controlar dispositivos remotamente y ver imágenes de video en tiempo real a través de un celular	6
Figura 4. Arquitectura propuesta de un dispositivo móvil integrado con un sistema domótico	8
Figura 5. Arquitectura de software planteada para el sistema domótico	9
Figura 6. Mapa Conceptual Marco Teórico	13
Figura 7. Arquitectura centralizada	16
Figura 8. Arquitectura distribuida	17
Figura 9. Arquitectura de una red GSM	27
Figura 10. Arquitectura de una red GPRS	29
Figura 11. Arquitectura de capas de J2ME	30
Figura 12. Arquitectura Midlet	32
Figura 13. Estados y Métodos de un Midlet	33
Figura 14. Proceso de verificación de clases en un midlet	34
Figura 15. Arquitectura J2EE multicapa	36
Figura 16. Diagrama de casos de uso Administración de Locaciones	41
Figura 17. Diagrama de casos de uso Administración de Dispositivos	42



Figura 18.	Diagrama de casos de uso Administración de Usuarios	43
Figura 19.	Diagrama de casos de uso Administración de permisos sobre dispositivo	44
Figura 20.	Diagrama de casos de uso Interacción con el hardware domótico	45
Figura 21.	Arquitectura de los sistemas domóticos tradicionales	49
Figura 22.	Arquitectura de Hardware	50
Figura 23.	Estructura lógica de la página Web	52
Figura 24.	Estructura lógica del midlet	53
Figura 25.	Diagrama de software del servlet de la aplicación Web	54
Figura 26.	Diagrama de software del servlet de la aplicación Midlet	55
Figura 27.	Diagrama de clases sin patrones	57
Figura 28.	Patrón Front Controller	58
Figura 29.	Patrón View Helper	59
Figura 30.	Patrón Data Access Object	60
Figura 31.	Estructura de los paquetes usados en la aplicación Web	61
Figura 32.	Diagrama de clases del usuario	62
Figura 33.	Diagrama de clases del dispositivo	63
Figura 34.	Diagrama de clases de la locación	64
Figura 35.	Diagrama de clases de grupo de usuarios	65
Figura 36.	Diagrama de clases servlet controlador domótico	66
Figura 37.	Diagrama de clases Dispositivos Servlet controlador Domótico	66
Figura 38.	Diagrama de clases Usuarios Servlet controlador Domótico	67

Figura 39.	Diagrama de clases Locaciones Servlet controlador Domótico	68
Figura 40.	Interfaz de MySQL Administrator	69
Figura 41.	Archivo web.xml	70
Figura 42.	Página de inicio	71
Figura 43.	Administración del sistemas	71
Figura 44.	Detalle de dispositivos	72
Figura 45.	Midlet domótico	73
Figura 46.	Envío de orden de encender a través del midlet	74

## LISTA DE TABLAS

	<b>pág.</b>
Tabla 1. Requerimientos no funcionales del sistema	45
Tabla 2. Elementos que conforman la arquitectura del sistema	51
Tabla 3. Comparación de los programas de desarrollo que se pueden utilizar para el desarrollo de la aplicación	55
Tabla 4. Pruebas unitarias	75

El presente proyecto está orientado al desarrollo de una aplicación para un dispositivo celular (midlet) capaz de interactuar con un hardware domótico que simula a escala el ambiente de una casa; el objetivo es controlar el encendido y apagado de las luces que lo conforman, manteniendo las características de un sistema distribuido. Para lograr este objetivo, se diseñó un sistema de Servlets siguiendo patrones de programación de J2EE como controladores frontales, despachadores y acceso a datos. La aplicación midlet utilizó RMS para guardar y consultar datos y un computador llamado controlador domótico fue configurado para enviar por el puerto serial una orden de apagado o encendido al hardware domótico para que fuese ejecutada.

Esta investigación se realizó como parte de una serie de alternativas de interfaces que se pueden crear para controlar dispositivos remotamente; próximos desarrollos se pueden basar en los conceptos que se presentan a continuación para manipular dichos dispositivos no solo desde un celular, sino a través de reconocimiento de voz, realidad virtual o sistemas expertos.

*Palabras clave:* Midlet, Domótica, Servlets, Sistemas Distribuidos, XML, RMS.

This Project is self-oriented to the development of an application for a mobile device (midlet) capable to interact with a domotic hardware which simulates on a reduced scale a house environment; the prime objective is to control the actions of turning on and off the lights attached to the device taking into account the features of a distributed system. To achieve this goal, a Servlet system was developed by following J2EE programming patterns such as front controller, dispatchers and data access. The midlet application used the RMS for storing and fetching data while a computer called domotic controller sent by the serial port a “turn off” or “turn on” instruction to the domotic hardware which executed the order.

This research is being carried out as a series of alternatives of interfaces that can be created in order to control devices remotely; further developments might be based on the concepts described up next to manipulating such devices not only from a cell phone, but through voice recognizing, virtual reality or expert systems.

*Keywords:* Midlet, Domotics, Servlets, Distributed Systems, XML, RMS.

## INTRODUCCION

Controlar dispositivos de forma remota se ha convertido en una innovación tecnológica que existe actualmente en Europa y Estados Unidos y que día a día se ha venido introduciendo en Suramérica. Desde enviar ordenes a robots y máquinas a través de Internet hasta apagar o encender la luz de una habitación son las alternativas que ofrecen diferentes empresas dedicadas al desarrollo de esta clase de aplicaciones.

El presente proyecto tratará la parte de domótica, nombre que se le da a los dispositivos electrónicos de una casa que pueden ser controlados por medio de diferentes interfaces, como por ejemplo, un teléfono celular. De todos estos dispositivos se tendrán en cuenta solamente los de iluminación para lo cual se creará una aplicación midlet para que un usuario desde cualquier parte del mundo pueda ver cuáles luces están encendidas, cuales apagadas y poder controlarlas o programarlas.

En la primera parte del documento se describirán las causas y motivaciones que hicieron que este proyecto fuese realizado y porqué es importante para el desarrollo tecnológico de la Universidad. Luego se hará un recuento de las aplicaciones que existen actualmente en el mercado cuya función es controlar dispositivos domóticos de forma remota. Se hablará por ejemplo de cómo IBM ha diseñado un programa que funciona en una palm y que puede controlar prácticamente hasta la temperatura de la estufa de una casa. Se dará también un vistazo a las investigaciones hechas en universidades alrededor del mundo y cómo se integra la domótica con un diseño de base de datos y servidores para darle un valor agregado a estas aplicaciones: administración y seguridad.

La segunda parte comprenderá todas las bases teóricas necesarias para llevar a cabo el desarrollo del proyecto. Se tocaran temas básicos de domótica y sus últimos adelantos; también se describirán las características más importantes del software y la arquitectura que intervienen para crear una aplicación Midlet. Por último, se dará un repaso a lo que son los sistemas distribuidos ya que estos dan una utilidad importante a cualquier administración que se haga de tipo cliente-servidor.

En la tercera parte del proyecto comprendida por los requerimientos, el diseño global y detallado, la implementación y las pruebas, se describe todo el desarrollo del software, qué herramientas de desarrollo se utilizaron, cómo se hizo la arquitectura de hardware, cómo es la arquitectura de software, qué patrones se utilizaron para realizar los Servlets, en qué momento interviene RMI en el sistema, entre otros.

La cuarta parte comprende las pruebas al sistema y las consideraciones; en esta parte se documentan los posibles errores y las cosas que hay que tener en cuenta para desarrollar aplicaciones de este estilo.

Por último, los autores plantean las conclusiones a las que llegaron después de haber desarrollado todo el proyecto, se formulan posibles mejoras al software y se dan recomendaciones para que futuros proyectos se basen en el que a continuación se presenta.

## **1. PLANTEAMIENTO DEL PROBLEMA**

### **1.1 DESCRIPCION DEL PROBLEMA**

En los últimos años, las construcciones de casas y edificios han venido incluyendo en sus arquitecturas un nuevo concepto, dispositivos domóticos los cuales, pueden ser controlados sin que las personas tengan que estar en ese lugar físicamente. Sin embargo, se hace necesaria una arquitectura de hardware y software que no genere mayores cambios ni sobre costos al momento que se requiera adicionar un nuevo dispositivo domótico y así mismo, que tenga niveles de seguridad de tal forma que no todos los usuarios tengan los mismos privilegios de controlar su funcionamiento y que además sea capaz de poder recibir instrucciones remotas como por ejemplo, una orden de encender las luces de la casa a través de Internet, un palm o un teléfono celular.

Debido a la vertiginosa velocidad con que la tecnología ha avanzado, hoy en día es posible encontrar en el mercado diferentes tipos de dispositivos, la mayoría de ellos inalámbricos, capaces de integrar varios procesos en uno solo. Ejemplo de esto son los celulares y la creciente demanda de usuarios que ya no solo requieren suplir la necesidad de hablar y estar comunicados con sus allegados, sino la de encontrar la mayor capacidad de almacenamiento, conexiones bluetooth, cámara digital integrada, consulta de correo electrónico y archivos de la empresa, entre otros.<sup>1</sup>

Aprovechando la tecnología celular, la presente investigación plantea de qué manera pueden interactuar un dispositivo móvil y uno de hardware domótico, de tal forma que sin importar el número de usuarios o el número de dispositivos domóticos conectados o si la información es procesada en diferentes servidores, al final se pueda tener un sistema seguro, confiable y escalable.

Varias empresas ya ofrecen este servicio, principalmente en Europa y Estados Unidos. Proxima Systems<sup>2</sup> de España, por ejemplo, presenta una solución en donde el usuario a través de su celular puede ver lo que está pasando en su casa e incluso enviar mensajes SMS para dar instrucciones de apagado o encendido de luces. En América Latina esta tecnología está comenzando y en Colombia ya existen varios proyectos en edificios que son construidos con elementos domóticos.

---

<sup>1</sup> Asociación de la Industria Celular en Colombia. [en línea]. (Consultado en Febrero de 2007). <http://www.asocel.org.co/prensa.php>

<sup>2</sup> Proxima Systems. [en línea]. (Consultado en Enero de 2007). <http://www.proximasystems.net/es/productos/lincegprs.php>



De igual manera, se han hecho investigaciones en varias universidades de Colombia y el mundo de cómo puede integrarse en una arquitectura ya diseñada de servidores de autenticación, bases de datos o controladores de hardware, un nuevo dispositivo móvil que no implique rediseñar toda la estructura ya establecida, lo que podría generar altos costos.

El presente proyecto se basará en las mencionadas investigaciones para formular una arquitectura que permita controlar un sistema domótico remotamente, en este caso, desde un teléfono celular. Esto traerá un aporte de conocimiento para que empresas colombianas no necesiten en un futuro importar aparte del hardware, el software que controle dicho sistema sino que por el contrario, por ser diseñado en una plataforma con características de sistema distribuido y software libre, funcione para cualquier tipo de hardware domótico, con varios usuarios interactuado con el sistema al mismo tiempo, sin necesidad de tener que hacer inversiones significativas.

## **1.2 FORMULACIÓN**

A través de herramientas de desarrollo basadas en software libre para dispositivos móviles y aplicaciones Web, ¿De qué manera puede un teléfono celular controlar las luces de un hardware domótico dentro de una arquitectura distribuida?

## **1.3 DELIMITACIÓN**

El presente proyecto hace parte de uno de los primeros desarrollos de aplicaciones para dispositivos domóticos en la línea de investigación de Telecomunicaciones de la Facultad de Ingeniería de Sistemas en la Universidad El Bosque. La base de su desarrollo serán dispositivos móviles tales como teléfonos celulares por lo que está fuera del alcance considerar su funcionamiento para palms o algún otro dispositivo inalámbrico diferente al propuesto.

Todo el control de apagado, encendido o programación se hará a través del dispositivo móvil. Se utilizará software libre como Netbeans 5.5 el cual incluye el paquete Mobility Pack para el desarrollo de la aplicación del dispositivo móvil y la versión 1.6 de JDK para la parte del servidor. La aplicación cliente correrá en un emulador de celular que soporte aplicaciones en Java y todas las aplicaciones de la parte del servidor correrán bajo Windows XP. La base de datos será diseñada en MySQL 5.0 y el servidor de aplicaciones será administrado por Tomcat versión 5.0.

El sistema domótico será simulado en un hardware construido y referenciado en la tesis de Angarita, Carlos y Castro, Fidel. “Estudio e Implementación de un Sistema

Domótico”<sup>3</sup>; éste hardware es una maqueta que incluye una tarjeta X-10 la cual permite controlar diferentes dispositivos a escala conectados, como por ejemplo, las luces de una casa las cuales podrán ser encendidas, apagadas o programadas para realizar dichas funciones de forma automática.

Se utilizará también una interfaz Web que permitirá configurar el sistema para su uso remoto. Por ejemplo, a través de la página Web se ingresarán los diferentes nombres de usuarios y habrá una opción para asignar permisos a éstos; así mismo se podrán registrar los dispositivos domóticos que se necesiten controlar.

## 1.4 ESTADO DEL ARTE

Varias empresas han venido lanzando al mercado productos que permiten un control de los dispositivos domóticos conectados en el hogar a través de forma remota, ya sea por un celular, una palm o una pagina Web.

En el año 2005 comenzó un proyecto europeo llamado HomeTalk<sup>4</sup> en el cual participaron entre otros, IBM de República Checa y Telefónica de España. Su objetivo fue aprovechar diferentes interfaces para ofrecer un servicio de control de los dispositivos de la casa a personas que van desde los que les gusta la tecnología hasta los ancianos y discapacitados. El principal atractivo fue una palm que reconocía órdenes de la voz humana.

Una persona podía, por ejemplo, estar en una ciudad X, en la cocina de su casa y encender la estufa para preparar la cena. Otra persona a kilómetros de distancia veía en su palm la imagen de la persona que está cocinando y podía con su voz darle la orden a la estufa de encenderse a 150 grados y apagarse en 20 minutos. Al mismo tiempo y en la misma palm, se podía ver el estado de la lavadora, de las luces o de más dispositivos conectados al sistema.

La figura 1 muestra imágenes de diferentes ambientes los cuales pueden ser monitoreados o controlados desde el dispositivo móvil.

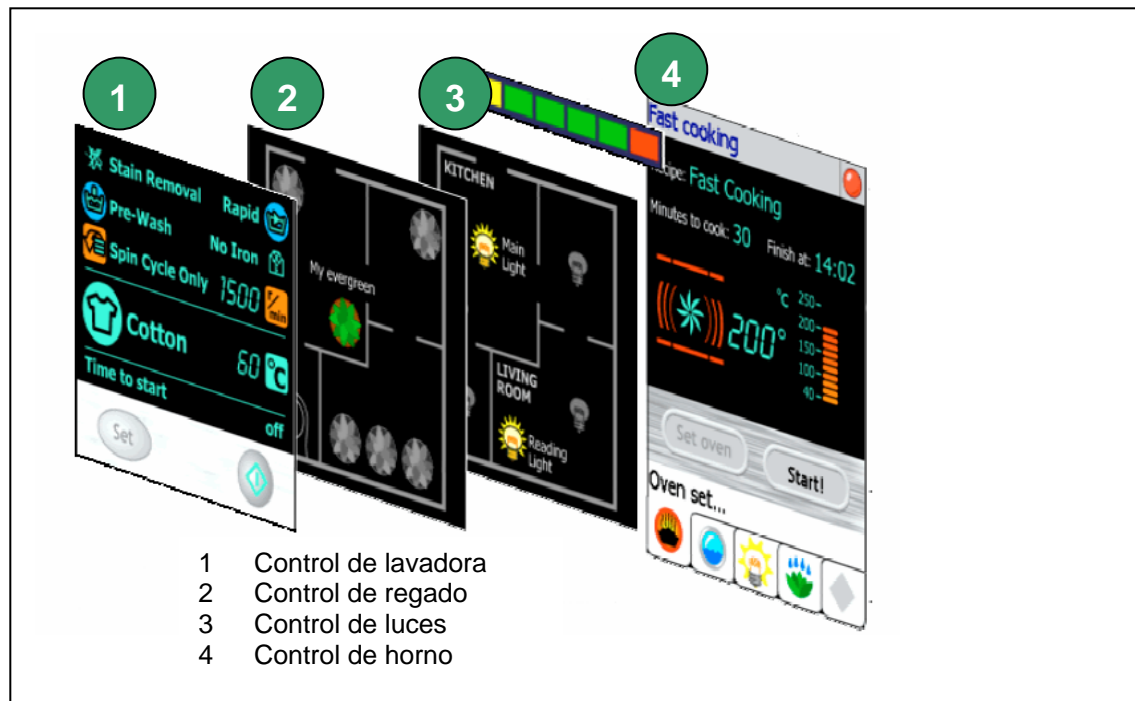
En España este proyecto fue probado en diez hogares donde se instaló el sistema que consistía de una lavadora y un horno inteligentes. Los usuarios además podían conectar otros dispositivos a las tomas domóticas. En Grecia, por su parte, el proyecto se orientó a personas mayores con alguna discapacidad y se instaló en dos centros para personas especiales.

---

<sup>3</sup> ANGARITA, Carlos y CASTRO Fidel. Estudio e implementación de un sistema domótico. 2001. 120p. Trabajo de Grado (Ingeniero Eléctrico) Universidad Antonio Nariño. Facultad de Electrónica

<sup>4</sup>KLEINDIENST, Jan., MACEK, Tomáš., SERÉDI, Ladislav y ŠEDIVÝ, Jan. Vision-Enhanced Multi-Modal Interactions in Domotic Environments. [en línea]. (Consultado en Marzo de 2007). [http://ui4all.ics.forth.gr/workshop2004/files/ui4all\\_proceedings/adjunct/interactive\\_applications/103.pdf](http://ui4all.ics.forth.gr/workshop2004/files/ui4all_proceedings/adjunct/interactive_applications/103.pdf)

Hoy en día ese proyecto es más que realidad. El Grupo Pinar de España construyó una serie de viviendas bajo el nombre de Torre Pinar las cuales ya vienen con dispositivos domóticos donde la iluminación y los sistemas anti-intrusión entre otros, pueden ser controlados por teléfonos móviles o por Internet.<sup>5</sup>



**Figura 1. Entorno gráfico de HomeTalk (Tomado de Vision-Enhanced Multi-Modal Interactions in Domotic Environments)<sup>6</sup>**

HomeTalk no solo puede ser manejado por una palm sino por diferentes interfaces como un computador conectado a Internet o un teléfono celular. La desventaja de utilizar dispositivos móviles es su limitación en la pantalla comparada con la de un pc, debido a que parte del software está desarrollado en flash; esto hace que la interacción con el usuario sea diferente dependiendo del medio que utilice para controlar los dispositivos de la casa y más aún si se trata de una persona mayor de edad o con limitaciones físicas. Para superar estos limitantes, en su proceso experimental se utilizó una iPaq equipada con Pocket PC 2002 y se instaló en ella un software de reconocimiento de voz (IBM voiceXML technology) capaz de recibir y procesar diferentes instrucciones habladas al mismo tiempo.

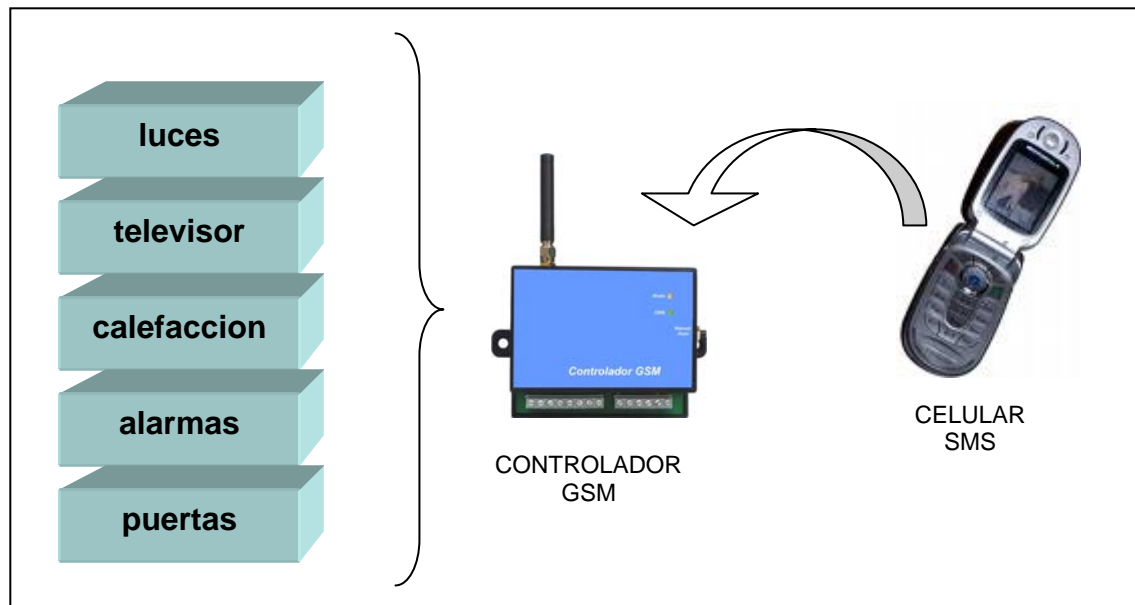
A parte de las instrucciones de voz propuestas en HomeTalk, existe otra manera de controlar dispositivos domóticos de forma remota: a través de mensajes SMS.

<sup>5</sup> Casadomo. [en línea]. (Consultado en Marzo de 2007). <http://www.casadomo.com/promocionesDetalle.aspx?id=7&idm=122&pat=122&prov=&sec=&year.>

<sup>6</sup> KLEINDIENST, MACEK, SERÉDI, ŠEDIVÝ, Op cit.

El usuario envía un mensaje de texto desde su teléfono celular a un controlador GSM que se encuentra instalado en su casa el cual, interpreta dicho mensaje y ejecuta la acción correspondiente. Pueden ejecutarse acciones como el encendido y apagado de luces, la activación de alarmas, el monitoreo de electrodomésticos como la estufa o la lavadora, entre otros. En el celular solo se necesita instalar una aplicación pequeña que permitirá que el usuario pueda interactuar con el sistema. Empresas como Proxima Systems<sup>7</sup> o LSB<sup>8</sup> en Europa, ofrecen entre sus productos este servicio.

La arquitectura que se muestra en la figura 2 es básicamente sencilla. Se debe tener un celular compatible con SMS y en casa el dispositivo GSM con una tarjeta SIM disponible para instalarla dentro. Todo lo que el usuario requiera administrar remotamente, en este caso los dispositivos domóticos, debe estar conectado al controlador GSM quien se encargará de recibir órdenes o de enviar alarmas al celular de la persona.



**Figura 2. Controlador GSM para envío y recepción de instrucciones a través de mensajes SMS**

Este servicio SMS para controlar dispositivos domóticos por ahora solo se conoce en Europa y Norteamérica y varias empresas que lo ofrecen no cobran cargos adicionales en el momento que un usuario envía un mensaje.

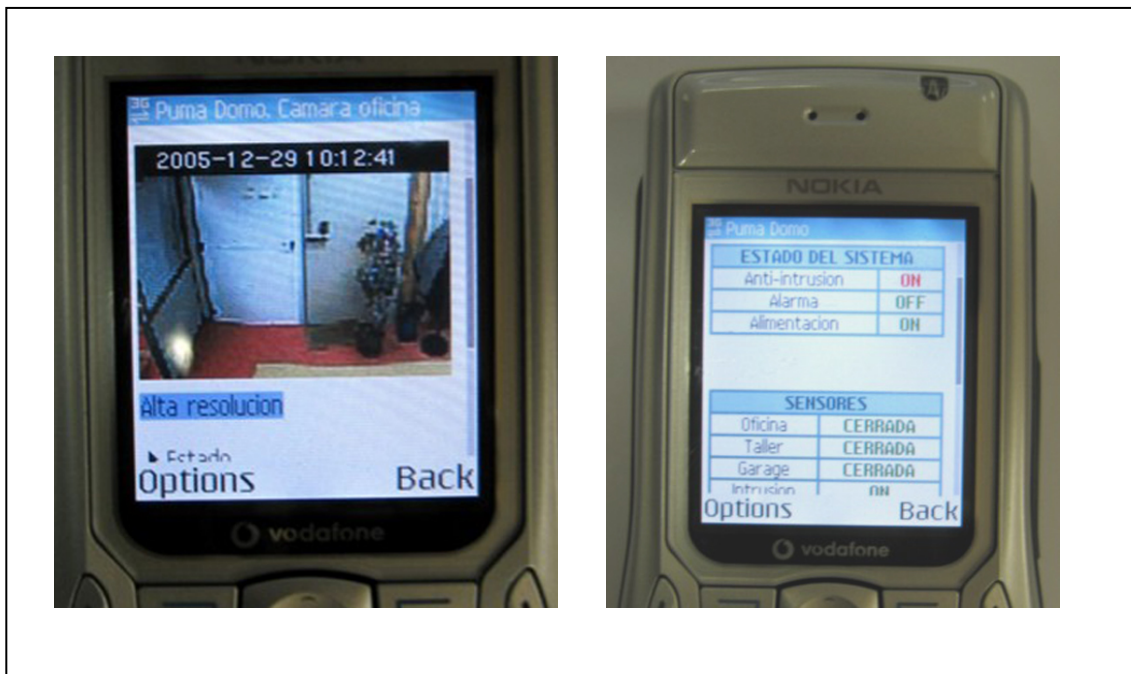
A diferencia de la opción SMS, actualmente diferentes compañías comercializan productos los cuales permiten ver y controlar cámaras Web a través del teléfono celular, así como el estado de sensores de intrusión o el estado de las luces de

<sup>7</sup> Proxima Systems. [en línea]. (Consultado en Abril de 2007). <http://www.proximasystems.net/es/productos/telecontrolgsm.php>

<sup>8</sup> LSB. [en línea]. (Consultado en Abril de 2007). [http://www.lsb.es/subhome\\_hogar.asp](http://www.lsb.es/subhome_hogar.asp)

determinada zona de la casa o de la oficina.<sup>9</sup> El gran problema de este sistema es sin duda su interfaz gráfica. El usuario debe contar con un celular de última tecnología y con una capacidad de memoria grande para que pueda visualizar los colores de las imágenes de la cámara de video o para poder al menos, ver el módulo de control de estado del sistema. La figura 3 ilustra cómo se ve en la pantalla del celular de un usuario la transmisión en vivo de una cámara instalada y un módulo para controlar diferentes dispositivos conectados al sistema.

Las compañías que comercializan estos productos los importan de Estados Unidos y Europa y ofrecen el paquete completo de hardware y software. Technoimport<sup>10</sup> por ejemplo, es una empresa colombiana que ofrece entre sus productos la automatización total de la casa que va desde abrir y cerrar las cortinas hasta todo un sistema de monitoreo y alarmas. También ofrecen el servicio de control por medio del teléfono celular y la opción de envío de mensajes cuando una novedad ocurre. Bogotá cuenta ya con esta tecnología implementada en varias construcciones sobretodo en el norte de la ciudad como el Edificio del Prado en el Chicó.



**Figura 3. Aplicación para controlar dispositivos remotamente y ver imágenes de video en tiempo real a través de un celular (Tomado de Casadomo)<sup>11</sup>**

<sup>9</sup> Casadomo. [en línea]. (Consultado en Marzo de 2007). <http://www.casadomo.com/noticiasDetalle.aspx?id=8628&c=6&idm=10&pat=10>

<sup>10</sup> Technoimport. [en línea]. (Consultado en Marzo de 2007). [http://www.latinpyme.com.co/asp\\_noticia.asp?ite\\_id=8907&pla\\_id=2&cat\\_id=5474&cat\\_nom=Actualidad](http://www.latinpyme.com.co/asp_noticia.asp?ite_id=8907&pla_id=2&cat_id=5474&cat_nom=Actualidad)

<sup>11</sup> Casadomo. Op cit.

A pesar de todo el confort que ofrecen los sistemas domóticos, existen varias desventajas para analizar. Altadil A., Féliz M. y Molina M. de Telefónica, publicaron un artículo titulado “¿Será el hogar digital un lugar sostenible?”<sup>12</sup>. En él analizan las diferentes tecnologías usadas actualmente para controlar dispositivos domóticos y concluyen que las personas mayores, enfermos con limitaciones o discapacitados pueden sacar el mejor provecho a estas aplicaciones debido a la facilidad de su manejo y a la amigable interacción. En los hogares comunes sin embargo, se observa un impacto ambiental especialmente en el consumo de energía debido a los nuevos dispositivos electrónicos que se deben conectar, sumado a los altos costos de instalación.

A parte de los productos que se comercializan alrededor del mundo, varias universidades han hecho investigaciones al respecto.

En el año 2004, en el segundo simposio sobre Tecnología Ambiental en la Unión Europea, R. Jimeno, Z. Salvador, A. Lafuente, M. Larrea y A. Uribarren publicaron un artículo titulado “Arquitectura para el control personalizado de recursos domóticos”<sup>13</sup> en donde los autores proponen un diseño para integrar un dispositivo móvil como una PDA o un celular, con un sistema domótico. Ellos emularon por medio de UPNP virtuales (Dispositivos Plug and Play) un reloj y un sistema de luces que pudieran ser controlados remotamente.

La figura 4 muestra el planteamiento propuesto, el cuál consta de (1) un servidor de Internet con una interfaz Apache Tomcat que permite la interacción de los usuarios con los componentes domóticos por medio de JSP. (2) Un manejador de dispositivos hecho en Java que registra en una base de datos los diferentes componentes conectados en el sistema domótico y que por medio de las órdenes que recibe del JSP interactúa con ellos. (3) Un manejador de usuario y perfiles hecho en Java que administra la creación y posteriormente la identificación y autenticación de los usuarios.

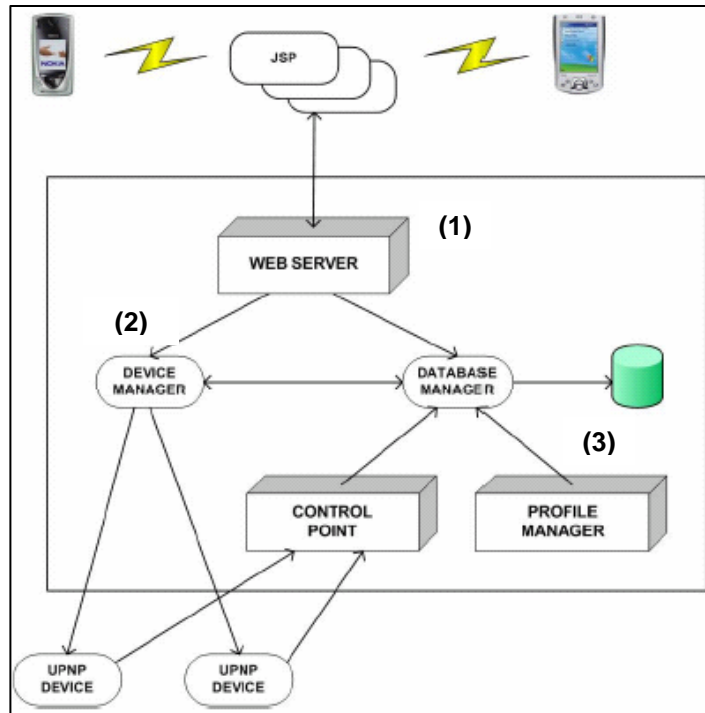
El sistema funciona a través del navegador del dispositivo móvil. Cuando el usuario ingresa a la página Web se autentica con un nombre y una contraseña; posteriormente, el servidor se conecta con el manejador de dispositivos y muestra al usuario los que están activos en ese momento. Dependiendo de los permisos asociados al usuario, los cuales están almacenados en una base de datos, éste puede enviar una orden al dispositivo domótico. Así mismo, se pueden establecer permisos para que determinado usuario sólo pueda controlar ciertos componentes; éstos son establecidos por un administrador del sistema.

---

<sup>12</sup> ALTADIL A., FÉLIZ M. y MOLINA M.. Será el hogar digital un lugar sostenible? Telefónica de España. [en línea]. (Consultado en Febrero de 2007). <http://www.tid.es/documentos/sostenibilidad/hogar.pdf>.

<sup>13</sup> R. JIMENO, Z. SALVADOR, A. LAFUENTE, M. LARREA, A. URIBARREN. An architecture for the personalized control of domotic resources. Noviembre 2004. Eindhoven, The Netherlands. [en línea]. (Consultado en Febrero de 2007). <http://portal.acm.org/citation.cfm?id=1031432>.

Los autores concluyen que este sistema tiene la ventaja de ser compatible con cualquier dispositivo UPNP que se requiera conectar pues el software es totalmente independiente del hardware domótico; así mismo, el hecho de manejar perfiles de usuarios, ofrece un valor agregado ya que éstos dispositivos domóticos no manejan dicha seguridad.



**Figura 4. Arquitectura propuesta de un dispositivo móvil integrado con un sistema domótico (Tomado de An architecture for the personalized control of domotic resources)<sup>14</sup>**

Por otra parte, cada uno de los dispositivos UPNP tiene su propia IP privada, lo que haría que controlarlos desde Internet no fuera posible. La arquitectura propuesta en cambio, centraliza todas las peticiones y órdenes que recibe en un solo nodo el cual, está conectado directamente a Internet, lo que le permite al dispositivo móvil interactuar con el.

En Colombia también se realizó un proyecto de grado que plantea una forma para controlar dispositivos domóticos de forma remota.

En el año 2005, Aguirre Gustavo., Pulgarín Ricardo. y Ríos Lus., de la Universidad de Manizales, realizaron el trabajo de grado titulado “Sistema domótico para el manejo remoto de dispositivos electrónicos a través de redes eléctricas”<sup>15</sup>. En la

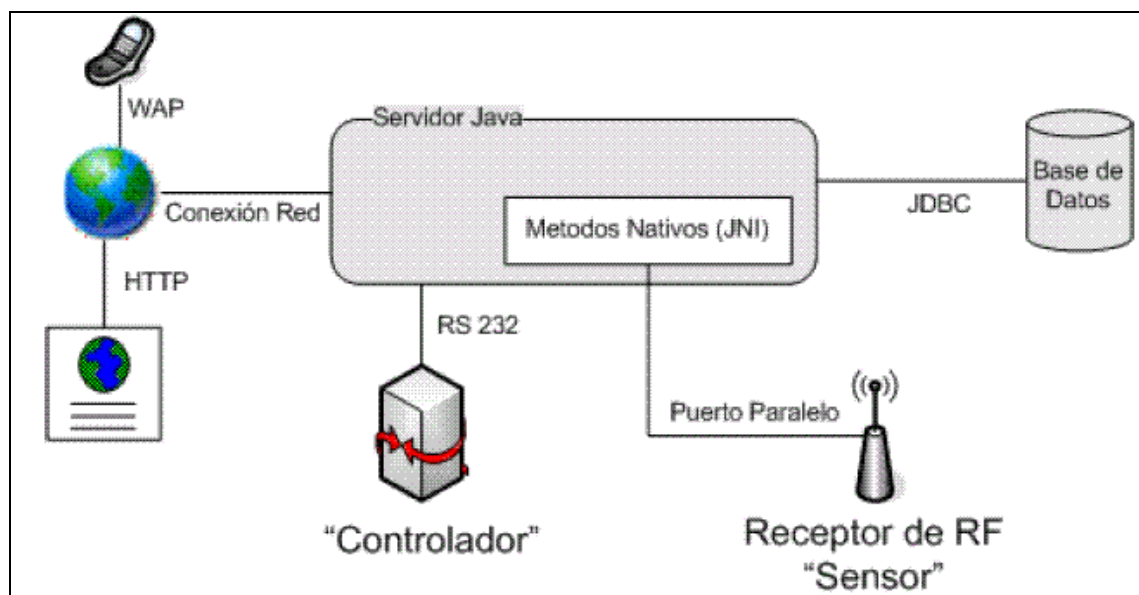
<sup>14</sup> Ibíd.

<sup>15</sup> 2005. 190p. Trabajo de Grado (Ingeniero de Sistemas). Universidad de Manizales. Facultad de Ingeniería. Disponible en la página Web de la Universidad de Manizales. <http://ingenieria.umanizales.edu.co/UserFiles/File/comites/invest/2005/Acta%2007-2005.pdf>



problemática plantean el acceso de los hogares colombianos a los dispositivos electrónicos como sensores o luces y hacen énfasis en que no se han encontrado registros de desarrollos hechos en Colombia en cuanto a infraestructura domótica por lo que sugieren el diseño de redes privadas para el manejo remoto de los dispositivos conectados en el hogar teniendo en cuenta el uso de materiales de bajo costo que permita una demanda de sistemas domóticos en el mercado.

Dentro de sus objetivos estaba el diseñar e implementar un sistema domótico para el manejo remoto de los electrodomésticos a través de Internet o tecnología WAP utilizando la red eléctrica de baja tensión de la casa. La metodología que implementaron consistió en el diseño de dos módulos de software; el primero se encargaba de la interacción del cliente con el servidor; el segundo consistía de un receptor de radio frecuencias que luego se encargaba de enviar las órdenes a un computador para que ejecutara la acción correspondiente en el sistema domótico como activar o apagar una alarma, controlar la iluminación de la casa, o el encendido y apagado de los electrodomésticos entre otros.



**Figura 5. Arquitectura de software planteada para el sistema domótico (Tomado de Trabajo de Grado)<sup>16</sup>**

Para llevar a cabo este diseño, utilizaron el servidor Apache Tomcat el cual almacenaba las páginas Web; la base de datos fue diseñada en SQL Postgres y el lenguaje de programación Java para crear las páginas JSP que interactúan con la aplicación WAP y para diseñar el código que enviaba instrucciones al puerto paralelo del computador. La figura 6 muestra la arquitectura de software que se

<sup>16</sup> Ibíd.



planteó la cual integra todos los componentes del sistema domótico para que interactúen entre sí.

Los autores concluyen en primer lugar que se debe configurar el puerto paralelo del computador en modo EPP ya que brinda mayor seguridad al momento de transmitir los datos; en segundo lugar, a pesar que el servidor corre bajo Windows por ser éste el más comercial, plantean que se pueden usar otras arquitecturas si se configura la interfaz Java para que pueda acceder al puerto paralelo y le dan crédito al software libre por su eficacia y confiabilidad y por su bajo costo al momento de desarrollar.

Otra manera de controlar dispositivos domóticos de forma remota es a través de Servicios Web. Esta investigación se llevó a cabo en Europa en una investigación de una Universidad en Italia.

Aiello M., del Departamento de Información y Tecnologías en Telecomunicaciones de la Universidad de Trento, Italia, escribió un artículo en el año 2005 titulado “El papel de los servicios Web en el hogar”<sup>17</sup> en donde propone una arquitectura basada en servicios Web (Web Services) para controlar los dispositivos de un sistema domótico en el hogar.

En esta publicación el autor hace una comparación de diferentes arquitecturas domóticas que existen en la actualidad y las agrupa en cuatro escenarios. El primero se refiere a un diseño en bus al cual todos los dispositivos domóticos están conectados. Sus desventajas son el ancho de banda limitado así como su escalabilidad pues máximo se pueden interconectar dieciséis elementos. El segundo se refiere a la necesidad de controlar dichos dispositivos remotamente en donde intervienen un gateway y un servidor, pero que limitan el sistema debido a que no todos los elementos domóticos comparten el mismo protocolo. En el tercer escenario se adhiere una interfaz para solucionar este problema, como un controlador bluetooth o infrarrojo el cual se comunica con el PC de la casa y este a su vez con el servidor. El cuarto escenario no solo existe un servidor que controle los dispositivos, sino que puede haber varios. Un ejemplo de este es la tecnología JINI.

El autor propone emplear los servicios Web para hacer la arquitectura escalable, es decir que se puedan introducir nuevos dispositivos sin ninguna complicación y que también sea heterogénea, lo que significa que sin importar el lenguaje de programación en que este desarrollado el servicio Web o la aplicación del cliente e independiente del medio de comunicación como bluetooth o GPRS, el cliente pueda invocar los servicios del servidor. Para esto se instalo una cámara Web a la cual se accedía por medio de una palm o un teléfono celular. El servidor era un Pentium IV con Windows XP; el celular era un Nokia 6600 con el sistema

---

<sup>17</sup> AIELLO, M. The role of web services at home. University of Trento. [en línea]. (Consultado en Abril de 2007). <http://eprints.biblio.unitn.it/archive/00000886>

operativo Symbian y la aplicación realizada en C++; la aplicación de la palm se diseñó en J2ME.

Se concluyo que gracias a los servicios Web, el acceso al sistema era totalmente dinámico; el introducir un nuevo elemento a la arquitectura no iba a ser un inconveniente pues se usó el escenario número tres planteado por el autor. La ventaja de usar servicios Web para controlar los un dispositivo domótico le dará al sistema escalabilidad y heterogeneidad.

## 1.5 JUSTIFICACION

La presente investigación integra dos conceptos que van de la mano: software, representado en un midlet y hardware representado en un dispositivo domótico. Ambos hacen parte de un proyecto de domótica que tiene un alcance que va desde el control por medio de dispositivos remotos hasta el reconocimiento de voz (para personas con limitaciones físicas), uso de la realidad virtual (visualización de los dispositivos en 3D) y diseño de sistemas expertos (casas capaces de “pensar”).

Como aporte a una de las primeras etapas, en esta investigación se propondrá un modelo cliente servidor que permitirá no solo que las luces puedan ser encendidas o apagadas, sino que aportará las bases de conocimiento para que diferentes dispositivos domóticos puedan ser conectados y que a su vez, diferentes usuarios puedan controlarlos. En un futuro, al continuar con los diferentes desarrollos de interfaces como por ejemplo, el reconocimiento de voz, no será necesario rediseñar desde cero la arquitectura ni el software; por el contrario, por ser un sistema distribuido, solo se tendrá que pensar en la interfaz para que el controlador interprete las señales auditivas.

Así mismo, con la realización del presente proyecto, las futuras investigaciones en la línea de Telecomunicaciones de la Facultad de Ingeniería de Sistemas de la Universidad El Bosque, se pueden basar en las partes del código para diseñar midlets, los cuales hacen parte de los desarrollos para celulares ofrecidos en el mercado actualmente. Los profesores y estudiantes tendrán un material de referencia bibliográfica que integra la mayoría de los conceptos tratados en Ingeniería de Software y Comunicaciones Digitales como son el desarrollo de aplicaciones móviles integrando RMS, o el diseño de páginas JSP para acceder a bases de datos, entre otros.

Con la estandarización de los protocolos de comunicación, las nuevas tecnologías como JINI<sup>18</sup> (basado en Java) hacen que se emplee el concepto de Plug and Play en los dispositivos domóticos. Gracias a que el diseño es hecho con herramientas de software libre, entre ellas Java, el conectar un dispositivo de una marca u otra no implicará grandes inversiones ni cambios, lo que abrirá un espacio en el

---

<sup>18</sup> JINI. [en línea]. (Consultado en Mayo de 2007). [http://www.jini.org/wiki/Main\\_Page](http://www.jini.org/wiki/Main_Page)

mercado de las compañías que comercializan estos productos pues prácticamente cualquier dispositivo se podrá conectar al sistema domótico y así será controlado desde cualquier interfaz inalámbrica.

## **1.6 OBJETIVOS**

### **1.6.1 Objetivo General**

Diseñar e implementar una aplicación para un teléfono celular que permita controlar el encendido y apagado de las luces de un dispositivo de hardware domótico que simula a escala el ambiente de una casa.

### **1.6.2 Objetivos Específicos**

Diseñar una interfaz Web que permita adherir usuarios y dispositivos domóticos al sistema, para que a través del celular se puedan generar las distintas acciones de control sobre el hardware domótico.

Programar el hardware domótico para que pueda recibir órdenes desde el controlador de dispositivos.

Diseñar una aplicación para el controlador de dispositivos que permita enviar órdenes al hardware domótico y viceversa con el fin de controlar el encendido y apagado de las luces de éste.

## 2. MARCO TEORICO

Para el desarrollo de la presente investigación, se desarrollarán los temas que a continuación se presentan en la figura 6.

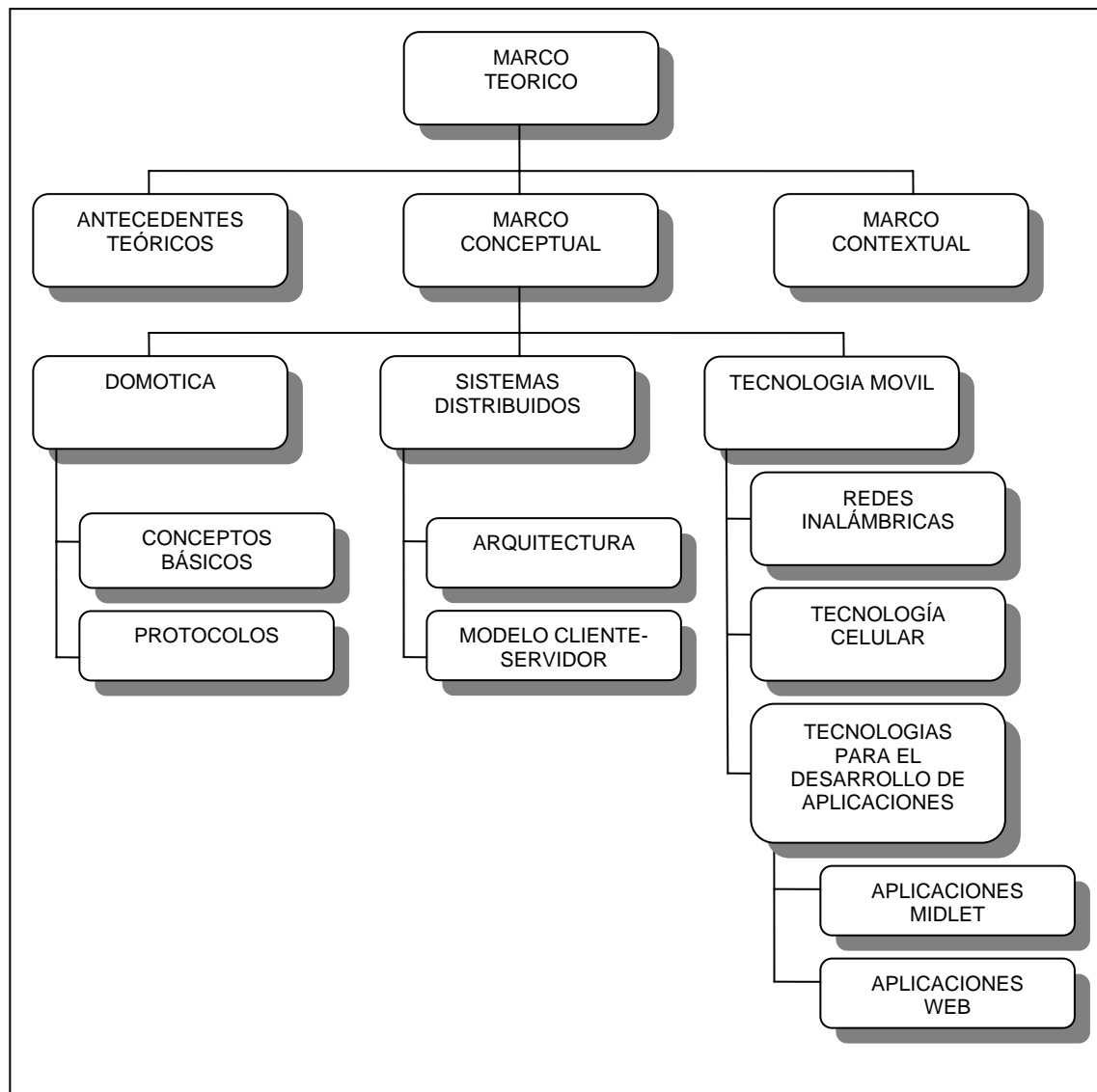


Figura 6. Mapa Conceptual Marco Teórico

### 2.1 ANTECEDENTES TEÓRICOS

La historia de la domótica es muy reciente. En los 80's los edificios empezaron a incluir en sus arquitecturas entre otros, sistemas de alarmas, control de puertas y sensores debido a que se propuso incrementar la seguridad de las instalaciones

en cuanto a intrusos o situaciones de emergencia como incendios. Más adelante, en los años 90, esta tecnología comenzó a integrarse en los hogares de Europa, Japón y Estados Unidos. Al principio, los dispositivos no se podían comunicar unos con otros pues eran producidos por diferentes fabricantes lo que originó el surgimiento de protocolos tales como X-10 hace mas de 30 años o Lonworks en los 80's que solucionaban este problema y hacían que su administración fuera más integral.

En Colombia hasta hace poco comenzaron a construirse proyectos inmobiliarios que incluían dispositivos domóticos, sobretodo en Bogota hacia el norte de la ciudad y desde el 2000 se han venido creando empresas que ofrecen estos servicios; algunas de ellas son Technoimport<sup>19</sup> que fue creada en el 2004 y tiene su sede en la capital y Homewireless E.U.<sup>20</sup> con sede en Manizales ambas dedicadas a la importación y comercialización de sistemas domóticos.

Con la llegada de Internet en los años 90, productos revolucionarios salieron al mercado, tales como las cámaras IP que podían ser monitoreadas a distancia, así como el control de los dispositivos conectados al sistema como luces y accesorios eléctricos. A medida que la tecnología fue ofreciendo nuevas alternativas, las interfaces de acceso a los sistemas domóticos se ampliaron; diferentes dispositivos inalámbricos se integraron a la arquitectura tales como controles remotos por Bluetooth o radiofrecuencia, las palm o los teléfonos celulares y se vio una oportunidad de aprovechar estas ventajas para hacer aun más optima la interacción del usuario con su casa.

En el caso de los celulares, por ejemplo, al principio solo se podían hacer llamadas. Hacia los años 80 en Estados Unidos y Europa la demanda de celulares era tal que las bandas tuvieron que ser aumentadas en una forma considerable y debido a la cantidad de fabricantes se creo GSM para estandarizar la manera de comunicación. Luego se vio la necesidad de integrar más servicios a parte de la voz; fue entonces cuando surgió GPRS que ofrecía mensajería de texto e Internet integrando una nueva tecnología no orientada a conexión, al mismo tiempo que nuevos dispositivos salieron al mercado con más capacidad de almacenamiento y de funcionalidades como cámaras, reproductores Mp3, entre otros.

A Colombia la tecnología celular llegó en 1994 e Internet un año después; hoy se calcula que el número de usuarios de celulares, según la Superintendencia de Industria y Comercio<sup>21</sup>, es de 28 millones aproximadamente y de más de 6

---

<sup>19</sup> Technoimport. [en línea]. (Consultado en Abril de 2007). <http://www.technoimport.com.co/>

<sup>20</sup> Homewireless E.U. [en línea]. (Consultado en Abril de 2007). <http://www.catalogodetecnologia.com/procesos/link.aspx?ie=4624919&idi=5&type=1&web=www.tuhogardigital.com>

<sup>21</sup> Superintendencia de Industria y Comercio. [en línea]. (Consultado en Abril de 2007). [http://www.sic.gov.co/Articulos\\_Pagina\\_Principal/Noticias/2006/Telefonia/Operadores\\_telefonia.php](http://www.sic.gov.co/Articulos_Pagina_Principal/Noticias/2006/Telefonia/Operadores_telefonia.php)

millones en el uso de Internet según informó la Comisión Reguladora de Telecomunicaciones<sup>22</sup>.

Al mismo tiempo que eran vendidos millones de teléfonos celulares en el mundo, éstos iban de la mano con las aplicaciones o sistemas operativos que venían con ellos. Symbian<sup>23</sup> por ejemplo, es un sistema operativo para celulares de última generación, que viene en más de 110 millones de dispositivos en el mundo y que integra funciones de multimedia y comunicación de datos y voz. Sus dos competidores más fuertes son RIM<sup>24</sup>, quienes lideran el mercado actual de las Black Berry, famosas por su integración con las aplicaciones de la oficina tales como archivos y correo electrónico empresarial y Microsoft Mobile<sup>25</sup> el cual aprovecha su paquete de aplicaciones office para introducirlas en los dispositivos móviles.

En la universidad El Bosque, se han realizado diferentes proyectos en tecnología móvil. Ortega Jimmy y Cárdenas Eduardo en su proyecto de grado titulado “Aplicación para consultas de sitios de entretenimiento desde dispositivos móviles”<sup>26</sup> diseñaron un midlet para que el usuario pudiera ver diferentes sitios de entretenimiento como restaurantes, bares, museos entre otros desde su celular; por otra parte, Rodríguez Julio y Calvache Miguel en su proyecto titulado “Simulación de la transmisión de datos en una red gprs por medio de ns2”<sup>27</sup> realizan diferentes escenarios de simulación para establecer los retardos en la transmisión de datos a través de GPRS.

## 2.2 MARCO CONCEPTUAL

**2.2.1 Domótica.** La domótica es un conjunto de sistemas automatizados domésticos, cuya función básica es brindar a las personas servicios de seguridad, control, comunicación, bienestar etc., y que utiliza sensores y controladores para llevar a cabo sus tareas. Por ejemplo, un sistema domótico puede controlar el apagado o encendido de las luces de la casa, programar el riego de plantas en el jardín, abrir y cerrar persianas, climatización automática, alimentación de mascotas, entre otros.

Los sistemas domóticos se pueden clasificar en los siguientes elementos:

---

<sup>22</sup> Comisión Reguladora de Telecomunicaciones. [en línea]. (Consultado en Abril de 2007). [http://www.crt.gov.co/crt\\_2001-2004/paginas/internas/biblioteca/regulatorio\\_a.htm](http://www.crt.gov.co/crt_2001-2004/paginas/internas/biblioteca/regulatorio_a.htm)

<sup>23</sup> Symbian. [en línea]. (Consultado en Abril de 2007). <http://www.symbian.com/>

<sup>24</sup> RIM. [en línea]. (Consultado en Abril de 2007). <http://www.rim.com>

<sup>25</sup> Microsoft Mobile. [en línea]. (Consultado en Abril de 2007). <http://www.microsoft.com/windowsmobile/default.mspx>

<sup>26</sup> ORTEGA, Jimmy y CARDENAS, Eduardo. Aplicación para consultas de sitios de entretenimiento desde dispositivos móviles. 2006. 137p. Trabajo de Grado (Ingeniero de Sistemas) Universidad El Bosque. Facultad de Ingeniería de Sistemas

<sup>27</sup> RODRIGUEZ, Julio y CALVACHE Miguel. Simulación de la transmisión de datos en una red gprs por medio de ns2. 2006. 108p. Trabajo de Grado (Ingeniero de Sistemas) Universidad El Bosque. Facultad de Ingeniería de Sistemas

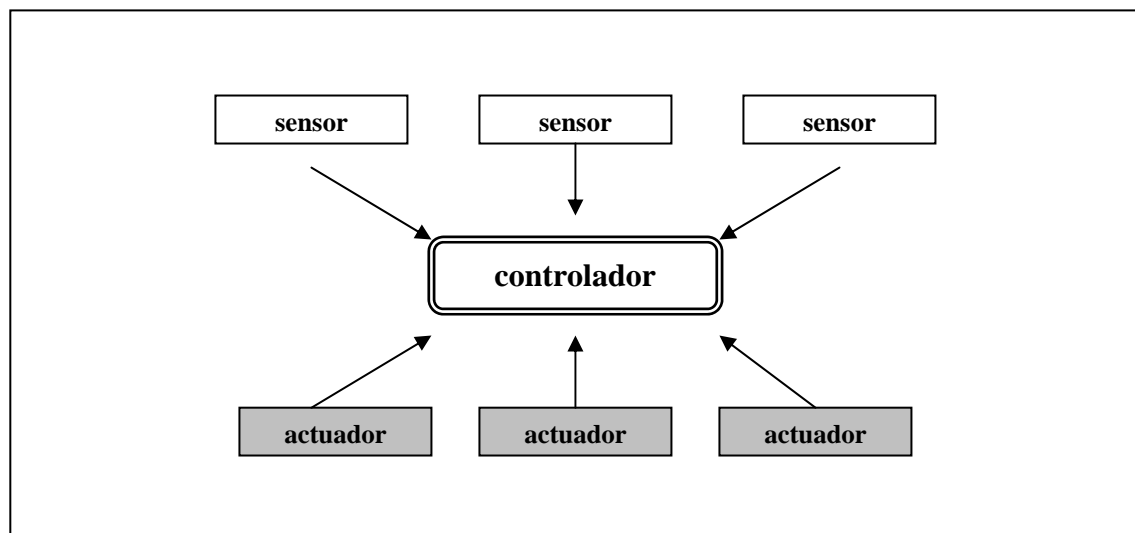
**Controlador:** El controlador es el dispositivo que administra el sistema. Por medio de este el software puede interactuar con el usuario final por medio de una interfaz como una pantalla, un sonido, una señal, etc.

**Actuador:** El actuador es un dispositivo de salida cuya función es llevar a cabo las tareas asignadas por el controlador como encender o apagar, abrir o cerrar, disminuir o aumentar, etc.

**Sensor:** El sensor es un dispositivo que monitorea de forma permanente los cambios o alteraciones sufridas en ciertos estados con el fin de enviar estos eventos al controlador. Existen varios tipos de sensores como climáticos, de movimiento, de reconocimiento de voz, de humo, etc.

- **Arquitectura de un sistema domótico.** La arquitectura de los sistemas domóticos se refiere a la forma como los diferentes elementos que conforman el sistema van a ser ubicados, qué medio utilizarán para comunicarse unos con otros y a través de qué interfaces. Existen dos tipos de arquitecturas, Arquitectura centralizada y Arquitectura distribuida.

**Arquitectura centralizada.** En esta arquitectura existe un controlador principal al cual van conectados todos los dispositivos que quieran ser monitoreados o manejados, como por ejemplo las puertas de la casa, las ventanas, las luces, un sistema de alarmas, la calefacción, etc., como se muestra en la figura 7.



**Figura 7. Arquitectura centralizada (Tomado de Casadomo)<sup>28</sup>**

<sup>28</sup> Casadomo. Domótica, Introducción. [en línea]. (Consultado en Abril de 2006). <http://www.casadomo.com/noticiasDetalle.aspx?c=14&m=21&idm=21&pat=20&n2=20>

Cada uno de estos dispositivos envía una señal al controlador principal, quien se encargará de procesarla y luego enviar las órdenes correspondientes para que se lleven a cabo.

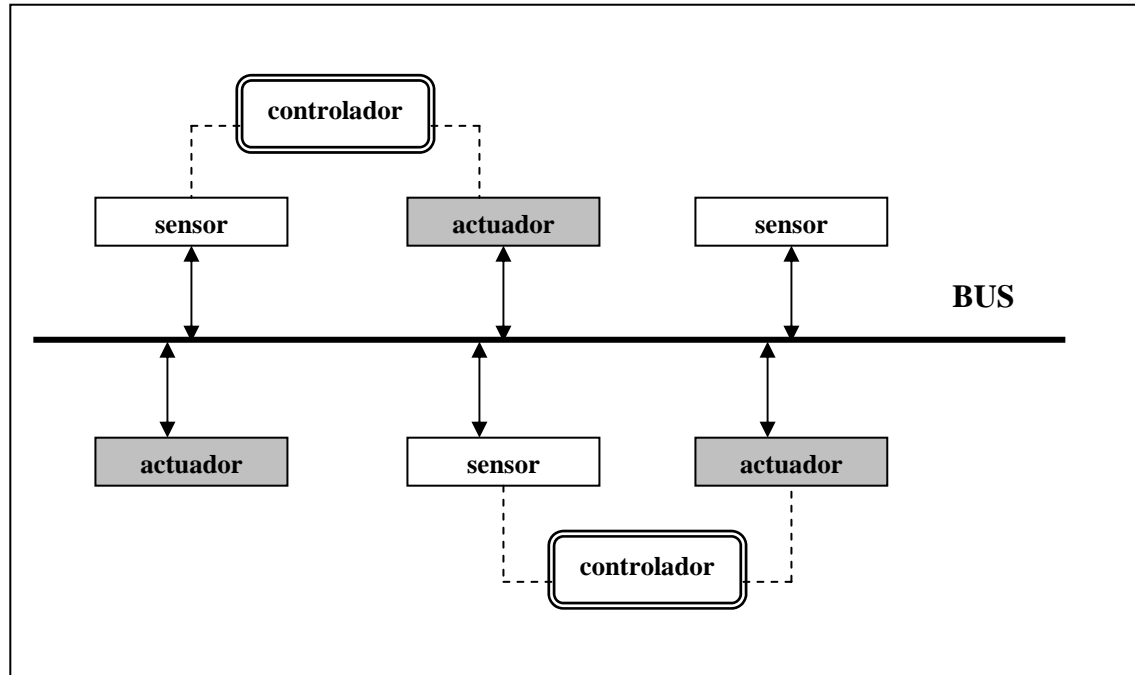


Figura 8. Arquitectura distribuida (Tomado de Casadomo)<sup>29</sup>

**Arquitectura distribuida.** En este tipo de arquitectura no existe un solo controlador principal, sino varios distribuidos, es decir, cada dispositivo puede tener su propio controlador y la manera como todo el sistema se comunica es generalmente a través de un bus de comunicaciones como se muestra en la figura 8. La manera como se distribuye cada controlador puede depender de la funcionalidad que éste brinde.

- **Medios de transmisión.** Los dispositivos conectados en un sistema domótico necesitan de un medio físico o inalámbrico para transmitir la información. A continuación se describirán los siguientes: Líneas de Distribución de Energía Eléctrica, Soportes Metálicos, Conexión sin hilos, Infrarrojos y Radiofrecuencia.

**Líneas de Distribución de Energía Eléctrica.** En este tipo de medio, se utilizan las líneas de energía eléctrica existentes en la casa lo que implica un bajo costo al momento de instalar los dispositivos, aunque al enviar o recibir datos se note una desventaja en cuanto a otros medios de transmisión.

<sup>29</sup> Casadomo. Ibid



***Soportes Metálicos.*** Este tipo de medio se refiere a los cables hechos de cobre, muy comunes en cuanto a comunicación se refiere. Los más comunes son el par trenzado y el coaxial.

- *Par trenzado.* El cable par trenzado está conformado por pares de cobre revestidos de material plástico. Existen dos tipos de cables trenzados, el STP (shielded twaisted pair), que tiene un revestimiento adicional de blindaje que le da mayor aislamiento a las ondas eléctricas y el UTP (unshielded twaisted pair) que no tiene este blindaje.

- *Cable coaxial.* Está formado por un conductor de cobre rodeado por una capa aislante en forma de malla los cuales a su vez, están protegidos por una capa aislante de plástico. Este cable es muy usado en redes de comunicación de banda ancha como transmisión de televisión y otros.

- *Fibra Óptica.* La fibra óptica está hecha de filamentos de vidrio o plástico por donde viajan haces de luz de un extremo al otro. En este proceso de transmisión de la luz intervienen un transmisor, encargado de transformar las ondas electromagnéticas en energía óptica y un receptor o detector óptico que transforma esta energía óptica de nuevo en ondas electromagnéticas.

***Conexión sin hilos.*** La conexión sin hilos se refiere a los la arquitectura donde no se utiliza un medio de transmisión físico de datos, sino que se usa un medio de transmisión por ondas infrarrojas o de radiofrecuencias.

***Infrarrojos.*** La conexión infrarroja se realiza a través de un diodo cuya función es emitir una luz en la banda IR sobre la que se superpone una señal. También interviene un receptor que se encarga de extraer la información generada a partir de esta señal. Según estándares internacionales la comunicación infrarroja debe ser bidireccional punto a punto en donde interviene un rayo de luz, un ángulo no mayor a 30 grados y una distancia de menos de un metro. Este tipo de transmisión puede alcanzar velocidades hasta de 16Mbps.<sup>30</sup>

***Radiofrecuencias.*** Las radiofrecuencias son radiaciones electromagnéticas emitidas por aparatos eléctricos y electrónicos, como por ejemplo televisores, radios, radares, etc. Estas radiofrecuencias oscilan entre 10khz y 300 Ghz.

---

<sup>30</sup> Medios de Transmisión/Componentes. [en línea]. (Consultado en Abril de 2006). <http://tecnologias.gio.etsit.upm.es/domotica/>

- **Tecnologías y Protocolos.** Luego de establecer una arquitectura para el sistema se puede definir el protocolo de comunicaciones, es decir, el lenguaje con el que un dispositivo se entenderá con otro. Entre los mas destacados están:

**X-10.** El control se hace a través de la línea de corriente de 110 o 220 voltios lo que permitía identificar cualquier dispositivo conectado al sistema por medio de la modulación de impulsos de 120khz, todo esto gracias a un protocolo de 16 grupos de direcciones llamados “housecodes” y 16 direcciones individuales llamadas “unitcodes”.<sup>31</sup> Este protocolo agrupaba bits que formaban 6 comandos los cuales eran recibidos por los dispositivos conectados; estos comandos eran: encendido, apagado, reducir, aumentar, todo encendido, todo apagado.

**Lonworks – LonTalk.** Surgió con el objetivo de crear un protocolo (LonTalk) para que cualquier dispositivo domótico pudiera comunicarse con otros, sin importar la marca o la funcionalidad. La plataforma Lonworks es independiente de los medios de comunicación, es decir, que si un dispositivo se comunica por cable UTP y otro por infrarrojo, no habría problema alguno al momento de uno enviar información al otro.<sup>32</sup>

**Protocolo LonTalk.** Con este protocolo los dispositivos conectados en un sistema domótico pueden comunicarse entre si. Sus principales características son:

- Soporta acuses de recibo
- Soporta diferentes medios de comunicación como par trenzado, fibra óptica, radiofrecuencia, red eléctrica y cable coaxial.
- Utiliza un algoritmo para evitar colisiones al momento de enviar y recibir paquetes
- El protocolo ha sido diseñado para implementarse en un solo chip de bajo costo
- Esta diseñado para trabajar sobre un modelo OSI

---

<sup>31</sup> Grun Technik. [en línea]. (Consultado en Abril de 2006). <http://www.gruntechnik.com/estandares.htm>

<sup>32</sup> Lonwork. [en línea]. (Consultado en Abril del 2006). <http://odisea.ii.uam.es/esp/recursos/Lonwork.htm>

**Cebus.** Su objetivo es unificar los protocolos de señalización infrarroja para el control remoto de electrodomésticos. Cebus soporta diferentes tipos de medio como cable trenzado, coaxial, de fibra óptica, radiofrecuencia o red eléctrica.

**Batibus.** La tecnología Batibus se basa en que todos los dispositivos conectados a la red pueden escuchar lo que ha enviado cualquier otro, todos procesan esta información, pero solo los interesados tomaran la información. Esta tecnología utiliza CSMA-CA para el control de acceso al medio y se les puede asignar una dirección fija para ser identificados en la red.

**Konnex.** Konnex nació en europa con el ánimo de estandarizar las tecnologías EHS, Batibus y EIBus. El objetivo de la unificación de estos estándares es el ofrecer al consumidor una forma de comprar productos de domótica de cualquier marca europea y que éstos sean compatibles entre sí, además de convertirse en un competidor de los estándares americanos Lonworks y Cebus.

**Bluetooth.** Bluetooth es un enlace radio de corto alcance que se extiende a un espacio de funcionamiento personal con un radio de hasta 10 metros. Funciona a un rango de frecuencias de 2,402 GHz a 2,480 GHz. Por Bluetooth se puede transferir voz, datos y hasta video y tiene un uso muy funcional en la domótica pues sirve de control remoto para enviar órdenes a los dispositivos conectados o para que uno se comunique con otro.

**Ethernet.** Esta tecnología es la más usada en el mundo y se basa en el protocolo TCP/IP, lo que hace posible la comunicación entre dos o más computadores gracias a que es compatible con cualquier sistema operativo y con cualquier tipo de hardware. Esta tecnología es independiente del medio físico por lo que puede ser implementada en cables coaxial, fibra óptica, radiofrecuencias, entre otros, pero el más común es el par trenzado y utilizando una topología de estrella, bus o anillo con el uso de switches y routers.

**2.2.2 Sistemas Distribuidos.** Los sistemas distribuidos consisten en conectar varios computadores en una red para comunicarse entre si, compartir recursos y coordinar actividades bajo un protocolo de cliente y servidor. Estos sistemas cuentan con varias características que hacen que el servicio sea eficiente, entre los cuales están: Transparencia, Flexibilidad, Confiabilidad, Desempeño, Escalabilidad, Seguridad y Tolerancia de fallas.<sup>33</sup>

- **Arquitectura de los sistemas distribuidos.** Dependiendo del tipo de servicio que se quiera prestar será el tipo de servidor. Entre los tipos de servidores mas

---

<sup>33</sup> TANENBAUM, Andrew S. Sistemas Operativos Distribuidos. México: Prentice-Hall. 1996. 20 p.

comunes se encuentran: Servidores de Archivos, Servidores de Bases de Datos, Servidores Web y Servidores de Aplicación.

Para que el software funcione correctamente en los sistemas distribuidos debe cumplir con tres componentes principales que son Presentación, Lógica de aplicación y Base de Datos. Los protocolos más utilizados son IP (Protocolo de Internet), TCP (Protocolo de Control de Transmisión), HTTP (Protocolo de Transferencia de Hipertexto), SMTP (Protocolo de Transferencia de Correo Simple) y POP3 (Protocolo de Oficina de Correo).<sup>34</sup>

- **Modelo Cliente-Servidor.** Los modelos cliente servidor manejan unas características esenciales que tienen que ver en cuanto al esquema del proyecto, direccionamiento y arquitectura que depende del nivel de abstracción del sistema. En este modelo se pueden manejar varias arquitecturas de acuerdo a lo que se necesite, entre las que mas se utilizan están la Arquitectura de Dos capas, Arquitectura de tres capas, Representación Distribuida, Representación Remota, Lógica Distribuida, Gestión Remota de Datos y Base de Datos Distribuidas.<sup>35</sup>

**2.2.3 Redes Inalámbricas.** Las redes inalámbricas permiten tener una comunicación directa con varios dispositivos o computadoras sin la necesidad de utilizar cables, facilitando la reubicación a la hora de trabajar, sin importar el lugar donde se encuentre el dispositivo, además de una fácil y rápida instalación con un menor costo. Para esta comunicación se utiliza componentes con ondas electromagnéticas, entre los cuales están los infrarrojos, microondas, láser y radio.

Para este tipo de redes existen dos categorías: De Larga Distancia que se utilizan para transmitir la información entre redes WAN es decir, entre ciudades o países cercanos manejando velocidades de 4.8 a 19.2 Kbps. Y las De Corta Distancia utilizadas en redes LAN como por ejemplo para la transmisión de información entre edificios cercanos con velocidades de 280 Kbps hasta 2 Kbps.<sup>36</sup> En las redes de Larga Distancia se manejan dos tipos de redes que son las Redes de Conmutación por paquetes y las Redes Telefónicas Celulares.

Entre las redes inalámbricas existen varias tecnologías para la transmisión de la información que dependen del uso que a estas se les de. Entre estas se encuentran:

**Bluetooth.** Es el nombre con que se le conoce a la especificación industrial IEEE 802.15.1 y cuya tecnología permite la comunicación inalámbrica entre varios

<sup>34</sup> Sistemas Distribuidos. [en línea]. (Consultado en Mayo de 2007). <http://www.monografias.com/trabajos16/sistemas-distribuidos/sistemas-distribuidos.shtml>

<sup>35</sup> Ibid Sistemas Distribuidos.

<sup>36</sup> AGUIRRE, José Eduardo. Redes Inalámbricas. [en línea]. (Consultado en Abril de 2007). <http://www.monografias.com/trabajos/redesinalam/redesinalam.shtml>.

dispositivos a corta distancia que generalmente es de 30 metros o menos, transmitiendo voz y datos a través de radiofrecuencia. Bluetooth proporciona una vía de interconexión inalámbrica entre varios aparatos que contengan esta tecnología como en el caso de los celulares, consolas, dispositivos PDA, cámaras digitales, computadores portátiles, impresoras, entre otros. La frecuencia de radio con la que trabaja esta tecnología esta en un rango de 2.4 a 2.48 GHz con un espectro amplio y saltos de frecuencia, que tienen la posibilidad de transmitir en Full Duplex y con un máximo de 1600 saltos/s. Estos saltos de frecuencia se dan entre un total de 79 frecuencias con intervalos de 1 MHz, lo que permite mayor seguridad y robustez. El hardware del dispositivo Bluetooth se compone de un Dispositivo de radio que esta encargado de modular y transmitir la señal y un Controlador digital que esta formado por una CPU, por un procesador de señales digitales DSP (Digital Signal Processor) llamado Link Controller (Controlador de enlace) y de los interfaces con el dispositivo anfitrión.<sup>37</sup> Cuando se forma una red con un máximo de siete dispositivos se forma una relación maestro-esclavo llamado *piconet* y la unión de varias de estas redes *piconet* forman una red mas grande llamada *scatternet*<sup>38</sup> y es utilizada generalmente para la comunicación en redes privadas de tipo PAN (Personal Area Network).

Bluetooth cuenta con varias versiones que son: Versión 1.1, versión 1.2, versión 2.0 y la versión 2.1. La versión 1.2, a diferencia de la 1.1, da una solución inalámbrica complementaria para que se pueda unir bluetooth con WiFi en el espectro de los 2.4 GHz sin que se presente interferencia entre ellos. Además de contar con varias versiones, esta tecnología tiene una clasificación de los dispositivos los cuales están en Clase 1, Clase 2 o Clase 3 dependiendo de la potencia de transmisión de cada y en donde no interfieren una con la otra.

**WIFI (Wireless Fidelity).** Se utiliza para la comunicación entre varios equipos que están situados dentro de un área de cobertura determinada ya sea que estén en el interior o en el exterior del edificio. WiFi se realizo bajo el estándar 802.11, la cual ha tenido varias actualizaciones que trabajan a diferentes velocidades como por ejemplo 802.11a, 802.11b, 802.11g. Uno de los problemas que actualmente enfrenta esta tecnología es la seguridad ya que muchas redes por su simplicidad de implementación, son instaladas por los administradores sin tener en cuenta la seguridad de la información que por ella pasa.<sup>39</sup>

**WIMAX (Worldwide Interoperability for Microwave Access).** “Interoperabilidad Mundial para Acceso por Microondas” Así se le conoce al estándar 802.16a la cual ofrece un mayor ancho de banda y alcance que las que ofrecen las diferentes versiones de WiFi. Mientras que WiFi maneja una tasa de transferencia de 11 Mbps. con una distancia de hasta 350 metros, WIMAX tiene una tasa de

<sup>37</sup> Bluetooth. [en línea]. (Consultado en Junio de 2007). <http://es.wikipedia.org/wiki/Bluetooth>

<sup>38</sup> QUINTAS, Agustín Froufe y CARDENES, Patricia Jorge. J2ME Java 2 Micro Edition. Manual de Usuario y tutorial. México: Alfaomega Grupo Editor, 2004. 572 p.

<sup>39</sup> WIFI. [en línea]. (Consultado en Junio de 2007). <http://72.14.253.104/search?q=cache:IDRayDVChhQJ:es.wikipedia.org/wiki/Wi-Fi+redes%2Binalambricas&hl=es&ct=clnk&cd=3&gl=us>

transferencia de 70 Mbps. con distancias que van hasta los 50 Km.<sup>40</sup> Como características importantes esta la gran productividad que tiene, Sistema escalable, Cobertura, OoS y por último Coste y riesgo.<sup>41</sup>

**WAP (Wireless Application Protocol).** Protocolo de Aplicaciones Inalámbricas. Esta tecnología ofrece unas aplicaciones digitales para los celulares y dispositivos móviles de manera que se pueda visualizar los contenidos de Internet como el correo electrónico, noticias y demás, con un formato especial de texto para los celulares con tecnología GSM. WAP utiliza un microbrowser con estándar WML (Wireless Markup Language) que es muy parecido a HTML y ofrece varias opciones para el transporte de información y para dispositivos, incluyendo SMS, 9.6 Kbit/s, GSM data y GPRS. Para que se pueda ofrecer un buen servicio de Internet para dispositivos móviles, WAP tiene unas características especiales para estos dispositivos como la pantalla más pequeña, limitaciones en la memoria y en la capacidad del procesador.<sup>42</sup> Su versión más reciente es WAP 2.0 que mejora sus servicios y seguridad para que puedan ser compatibles con Internet a diferencia de la versión anterior WAP 1.<sup>43</sup>

**2.2.4 Tecnología Celular.** Desde que empezó todo el auge de la telefonía celular ha habido un gran interés en cuanto a que estos sean cada vez más capaces de incorporar nuevas herramientas, que ayuden al usuario en las tareas y retos que el día a día va exigiendo. Aunque esta tecnología fue pensada para la comunicación por voz, en la actualidad existen muchos dispositivos que se encargan de otro tipo de servicios como lo son datos, audio y video y que para poderlos utilizar dependen del tipo o generación del celular que se valla a emplear.

- **Tecnologías de acceso celular.** Para poder transmitir la información a través de las redes, en la actualidad se utilizan tres tecnologías que son FDMA, TDMA y CDMA. A continuación se hace un descripción de cada uno de estos:

**FDMA (Frequency Division Multiple Access).** Esta tecnología es utilizada en la transmisión analógica ya que separa el espectro en distintos canales de voz separando el ancho de banda en pedazos, según su frecuencia en divisiones uniformes. FDMA utiliza todo el canal de 30 KHz para cada llamada telefónica que se realiza haciendo todo un derroche, lo que se puede mejorar utilizando la misma frecuencia utilizando el método Duplex por división de tiempo TDD (*Time Division Duplex*) el cual utiliza un canal y crea intervalos de tiempo.

---

<sup>40</sup> AGUIRRE, José Eduardo. Op. Cit.

<sup>41</sup> WiMAX. [en línea]. (Consultado en Junio de 2007). <http://es.wikipedia.org/wiki/WiMAX>

<sup>42</sup> BATES, Regis J. (Bud). Comunicaciones inalámbricas de banda ancha. Madrid: McGraw-Hill, 2003. 345 p.

<sup>43</sup> Wireless Application Protocol. [en línea]. (Consultado en Abril de 2007). <http://es.wikipedia.org/wiki/WAP>

**TDMA (Time Division Multiple Access).** En esta se comprime las conversaciones que son digitales y las envía utilizando una señal de radio por un tercio de tiempo, proveyendo acceso múltiple a un reducido número de frecuencias. TDMA hace posible que la transición entre los sistemas privados y públicos donde haya la necesidad de interconexión entre sistemas de diversos operadores y el sistema independiente de una instalación.

**CDMA (Code Division Multiple Access).** Digitaliza la información y la transmite a través de todo el ancho de banda que dispone para luego asignarles un código de secuencia a las llamadas que son colocadas en el canal.<sup>44</sup> El principio básico que sustenta CDMA es el uso del ruido de fondo para transportar señales radioeléctricas. Con CDMA se modifica el modo en que nos comunicamos mediante la mejora de la capacidad telefónica de los operadores celulares, mejora de la calidad de las comunicaciones de voz y la eliminación de imperfecciones audibles, reducción de la incidencia de llamadas caídas, reducción o eliminación de interferencias con otros dispositivos electrónicos en la zona, limitación de los riesgos potenciales para la salud y reducciones de los costes de operación ya que se necesitan menos emplazamientos<sup>45</sup>.

- **Generaciones de Telefonía Celular:** A lo largo del tiempo la tecnología celular va cambiando, introduciendo en ella nuevos componentes que permiten que se pueda tener un mejor manejo a las diferentes aplicaciones que se producen como consecuencia de las nuevas necesidades que se van presentando cada día. Para cumplir con estas necesidades y requerimientos, los celulares tienen que también incluir componentes o programas para poder llevar a cabo estas aplicaciones, dando como resultado una serie de generaciones que van desde aplicaciones básicas hasta la ultima generación en la que se incluyen muchas herramientas para poder llevar a cabo todas estas aplicaciones. Entre estas generaciones están:

**Primera Generación (1G).** Esta generación introdujo el concepto de la telefonía celular en el mundo y cuya característica principal era la transmisión por voz analógicamente. La velocidad de conexión no superaba a los 2400 bauds, no había preocupación en cuanto a seguridad y el tamaño de estos celulares era demasiado grande. Esta generación utilizaba FDMA (Frecuencia de división múltiple), lo que hace que la transferencia de celdas sea muy imprecisa y con baja capacidad, además se limitan el número de usuarios que se pueden manejar simultáneamente.<sup>46</sup>

---

<sup>44</sup> Tecnología de acceso celular. [en línea]. (Consultado en Marzo de 2007). <http://www.yucatan.com.mx/especiales/celular/tecnologiascelulares.asp>

<sup>45</sup> BATES, Regis J. (Bud). Op. Cit.

<sup>46</sup> Historia del Teléfono móvil. [en línea]. (Consultado en Marzo de 2007). [http://es.wikipedia.org/wiki/Historia\\_del\\_tel%C3%A9fono\\_m%C3%B3vil](http://es.wikipedia.org/wiki/Historia_del_tel%C3%A9fono_m%C3%B3vil)

**Segunda Generación (2G).** Aunque la generación anterior tuvo éxito por ser la que impulso la tecnología celular, en esta se presentaban muchos inconvenientes que había que resolver y que hacían que esta no fuera muy practica, entonces se vio la necesidad de desarrollar una nueva tecnología que mejorara todos estos problemas y de allí salió GSM (Global System Móvil). Esta generación se caracterizo por introducir los circuitos digitales de datos conmutados, el uso del TDMA (frecuencia por división de tiempo), que a diferencia de la anterior generación, esta permite mayor número de usuarios por canales separados.

**Generación 2.5G / 2.75G.** En si no existe un estándar o una tecnología llamada 2.5G o 2.75G, lo que sucede es que así se les suele llamar a algunos teléfonos móviles de segunda generación que incorporan algunas de las mejoras de la 2G y tecnologías de la 3G, con tasas de transferencia de datos mayores a los teléfonos de segunda generación y menores a los de la tercera generación.<sup>47</sup> Es marcada por el uso del sistema GPRS (General Packet Radio Service) que permite una velocidad moderada en la transmisión de datos utilizando los canales TDMA. Para poder transferir datos al ordenador, cámaras digitales, móviles y otros dispositivos se hace por medio de puertos bluetooth, IrDA o por conexión por cable.

**Tercera Generación (3G).** Esta se caracteriza por hacer la transmisión de voz y datos pero ya con acceso a Internet lo que la hace apta para utilizar aplicaciones de multimedia entre estos video conferencia, editores de texto, bases de datos, semejándose a un computador e incluyendo el sistema operativo y características de las laptops. Los estándares de esta generación utilizan CDMA (Acceso Multiple por division de código) para poder compartir el espectro entre varios usuarios. La evolución de la tecnología permite ofrecer al suscriptor velocidades de descarga superiores a 3 Mbit/s.<sup>48</sup> Se basa en los UMTS (servicios General de Telecomunicaciones Móviles) y dentro de esta existe una especificación conocida como TDD (Time Division Duplex), donde los links poseen la misma frecuencia pero usan distintos segmentos de tiempo. Sin embargo, TDD no se implementará en los mercados por un tiempo.<sup>49</sup>

En esta generación, IP esta basado en paquetes, lo que permite que se pueda estar en línea todo el tiempo que se desee sin tener que pagar por ello, sino solo por la transmisión de los datos. La tercera generación tiene soporte de conmutación de paquetes IP y soporte IP para videojuegos, comercio electrónico, video y audio.<sup>50</sup>

**Cuarta Generación (4G).** Esta generación no esta totalmente definida pero de acuerdo a lo que ya se ha establecido se puede decir que esta generación estará

---

<sup>47</sup> Telefonía móvil 2G. [en línea]. (Consultado en Junio de 2007). [http://es.wikipedia.org/wiki/Telefon%C3%ADa\\_m%C3%B3vil\\_2G](http://es.wikipedia.org/wiki/Telefon%C3%ADa_m%C3%B3vil_2G)

<sup>48</sup> Telefonía móvil 3G. [en línea]. (Consultado en Junio de 2007). [http://es.wikipedia.org/wiki/Telefon%C3%ADa\\_m%C3%B3vil\\_3G](http://es.wikipedia.org/wiki/Telefon%C3%ADa_m%C3%B3vil_3G)

<sup>49</sup> Telefonía Celular. [en línea]. (Consultado en Febrero de 2007). <http://www.monografias.com/trabajos34/telefonía-celular/telefonía-celular.shtml>

<sup>50</sup> Telefonía móvil 3G Op. Cit.



basada totalmente sobre IP y que con la convergencia de diferentes tecnologías y dispositivos, esta podrá ofrecer velocidades de acceso entre 100 Mbps en movimiento y 1 Gbps en reposo, manteniendo siempre la calidad del servicio (OoS) con una alta seguridad para así ofrecer servicios de cualquier tipo, en cualquier momento y con un costo menor. El concepto de la cuarta generación englobado dentro de “Beyond 3G” incluye técnicas de avanzado rendimiento de radio como MIMO y OFDM, además de esto, los principales nodos dentro de la implementación son el ‘Evolved Node B’ (BTS evolucionada) y el ‘System Access Gateway’, que actuará también como interfaz a Internet, conectado directamente al ‘Evolved Node B’. El servidor RRM será otro de los componentes que se utilizaran para facilitar la inter-operabilidad con las otras tecnologías.<sup>51</sup>

- **GSM (Groupe Special Mobile).** Entre sus características para el envío de datos inalámbricos sin importar el lugar y el momento, están: la velocidad de transferencia de 9.6 Kbps, el tiempo de establecimiento de conexión de 15 a 30 segundos y el pago que se hace por el tiempo de conexión. La baja velocidad de transferencia hace que gran cantidad de los servicios de Internet sean limitados, ya que con 9.6 Kbps no se puede navegar satisfactoriamente por Internet. Por estos tres factores es que GSM es utilizada principalmente para la transmisión de voz pero no para datos.<sup>52</sup> Para la arquitectura de la red GSM se tiene que tener en cuenta tres componentes principales que son:

**La Unidad o Estación móvil:** Consiste en un teléfono o terminal móvil que incluye una tarjeta inteligente conocida como modulo de interfaz o modulo de identidad del abonado SIM (*Subscriber Identity Module*), la cual le proporciona al usuario movilidad para que pueda realizar y recibir llamadas y disponer de los servicios contratados sin importar el Terminal del que se disponga, así este sea temporal. Cada unidad móvil se compone de dos identificadores específicos para los equipos que son: El identificador internacional del equipo móvil IMEI (*International Mobile Equipment Identity*) y el que contiene la tarjeta SIM, Identidad Internacional del abonado móvil IMSI (*International Mobile Subscriber Identity*) que contiene la información de autenticación.

**El subsistema de estación base:** Consta de dos partes muy importantes, la primera es una unidad móvil compuesta por los equipos transceptores de la estación base BTS (*Base Transceiver Station*), que es la que contiene los sistemas de radio para la interfaz aire o interfaz radio<sup>53</sup> hacia la unidad móvil del abonado y la segunda es el controlador de la estación base BSC (*Base Station Controller*) que gestiona los recursos de por lo menos una estación BTS y que además es la interfaz entre la estación base y la estación de conmutación del

---

<sup>51</sup> Telefonía móvil 4G. [en línea]. (Consultado en Junio de 2007). [http://es.wikipedia.org/wiki/Telefon%C3%ADa\\_m%C3%B3vil\\_4G](http://es.wikipedia.org/wiki/Telefon%C3%ADa_m%C3%B3vil_4G)

<sup>52</sup> Tecnología GSM. [en línea]. (Consultado en Junio de 2007). <http://www.casadomo.com/noticiasDetalle.aspx?c=61&m=164&idm=73&pat=148&n2=148>

<sup>53</sup> BATES, Regis J. (Bud) Op. Cit. Pág. 139

servicio móvil MSC (*Mobile Switching Center*). Estos dos componentes, el BTS y el BSC forman el subsistema de estación base BSS (*Base Station Subsystem*).

**El subsistema de conmutación y red:** Su núcleo es MSC que actúa como una central telefónica de clase 5 de la Red telefónica conmutada pública PSTN (*Public Switched Telephoned Network*) y que ejecuta todas aquellas funciones que intervienen en la conmutación y procesamiento que se necesitan para el establecimiento de las llamadas que se originan o terminan en la estación móvil, también se encarga de la autenticación, el traspaso de llamadas móviles, el registro en la red, las funciones con la base de datos y proporciona la interfaz a las redes de señalización para el establecimiento y liberación de la llamada a través de redes CCS7 (SS7).

En la figura 9 se muestran estos componentes así como los elementos que en ésta arquitectura intervienen y cómo se interrelacionan con cada uno de estos.

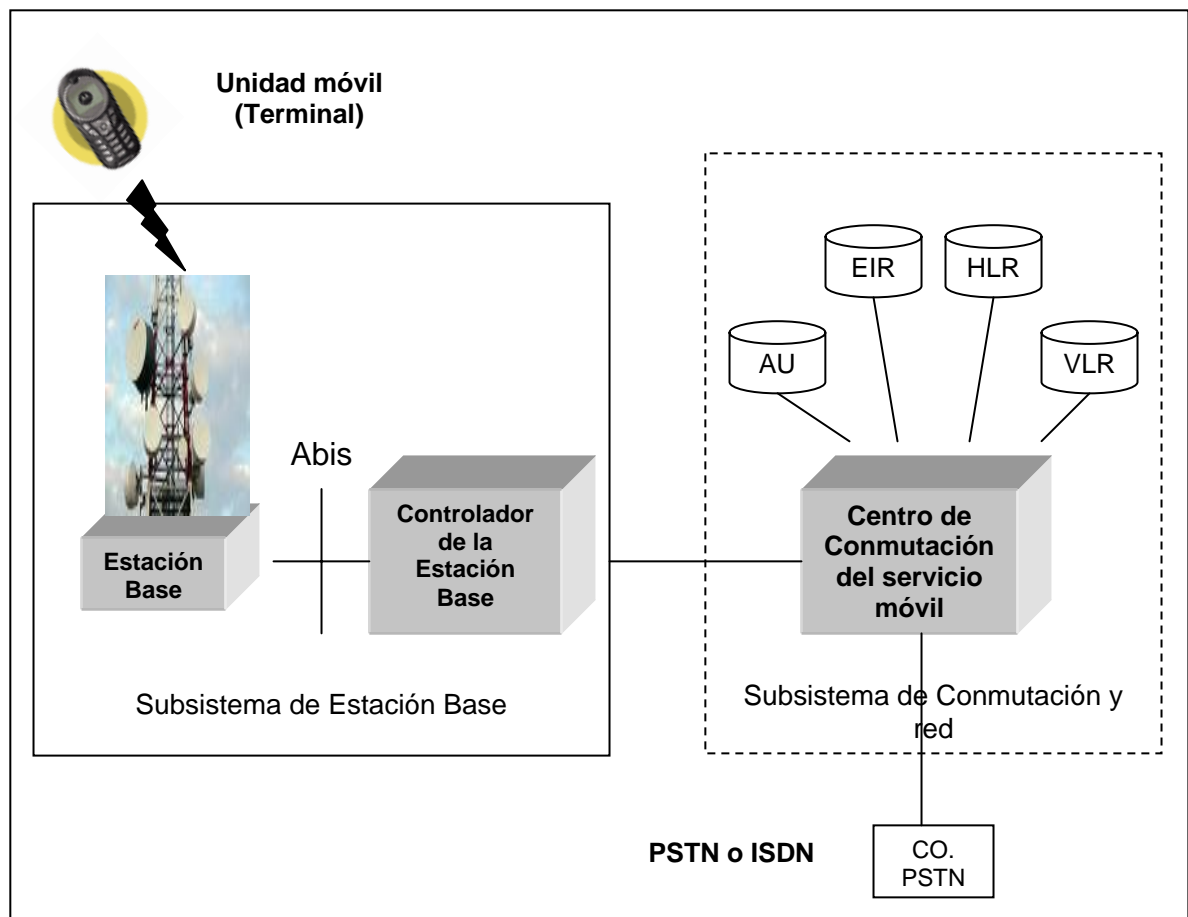


Figura 9. Arquitectura de una red GSM (Tomado de Comunicaciones Inalámbricas de banda ancha)<sup>54</sup>

<sup>54</sup> Ibid. Pág. 138

En el MSC se encuentra una base de datos con los usuarios que están registrados en la red y este es el Registro de localización de abonados locales o domésticos HLR (*Home Location Register*). En cada red GSM solo podrá haber un HLR y este será el propietario de la tarjeta SIM de los abonados de un determinado operador de red. El Registro de localización de abonados visitantes VLR (*Visiting Location Register*) es el encargado cuando los usuarios se mueven de un sistema a otro y mantiene una base de datos temporal en la que contiene los diferentes dispositivos que visitan cierta área del sistema y le notifica al HLR que el dispositivo se encuentra en una nueva posición haciendo que este se haga cargo de todos los pedidos de servicio. También se encuentran los registros del Centro de Autenticación AUC (*Authentication Center*) que es el encargado de validar y proteger la información del usuario como contraseñas y claves de autenticación que son utilizadas en la interfaz de radio y el Registro de Identidad de Equipos EIR (*Equipment Inventory Register*) que es una base de datos en la que se guarda la información del proveedor de los dispositivos de radiocomunicación que utilizan la red y el IMEI.

GSM utiliza para la interfaz aire una combinación entre TDMA y FDMA utilizando la banda de 25 MHz y dividiéndola en 124 frecuencias portadoras espaciadas por una separación de canales de 200 KHz.<sup>55</sup>

- **GPRS (General Packet Radio Service).** En un principio el Servicio general de Radiocomunicación por paquetes GPRS fue desarrollado para GSM, utilizando un principio de radiopaquetes para poder transferir los paquetes de datos del usuario entre las estaciones móviles GSM y las PDN externas. GPRS mejora las deficiencias de GSM como en el caso de la velocidad de transferencia de hasta 144 Kbps, la conexión permanente con un tiempo de establecimiento de conexión menor al segundo y el pago es por la cantidad de información transmitida, mas no por el tiempo de conexión.<sup>56</sup>

En el caso de la arquitectura de GPRS, mostrada en la Figura 10, para que esta pudiera integrarse con la que ya existía en GSM fue necesario crear nuevos nodos de red llamados nodos de soporte de GPRS GSN (*GPRS Support Nodes*) que son los que se encargan de la entrega y del recorrido de los paquetes de datos entre las estaciones móviles y las redes PDN externas. Para este fin hay dos tipos de nodos:

Un nodo de soporte de GPRS servidor SGSN (*Serving GPRS Support Node*): Se encarga de toda la entrega de los paquetes que van hacia y desde las estaciones móviles dentro del area de servicio. Además de esto también cumple con las funciones de autenticación y tarificación, encaminamiento y transferencia

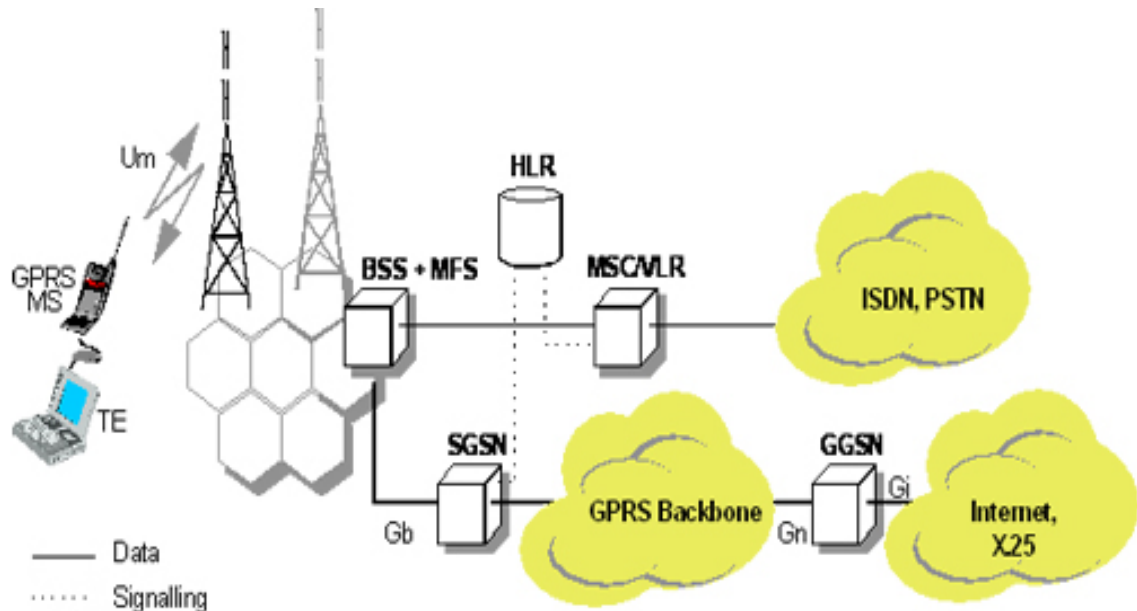
---

<sup>55</sup> Ibid. Pág. 140

<sup>56</sup> GPRS. [en línea]. (Consultado en Junio de 2007). [www.casadomo.com/noticiasDetalle.aspx%3Fc%3D65%26m%3D29%26idm%3D77%26pat%3D148%26n2%3D148+GPRS&hl=es&ct=clnk&cd=9&gl=us](http://www.casadomo.com/noticiasDetalle.aspx%3Fc%3D65%26m%3D29%26idm%3D77%26pat%3D148%26n2%3D148+GPRS&hl=es&ct=clnk&cd=9&gl=us)

de los paquetes, la gestión de la movilidad y la gestión del enlace lógico. El registro de localización SGSN guarda toda la información de los usuarios GPRS que están registrados en el SGSN, su la localización como la célula actual y el VLR, los perfiles del usuario IMSI y direcciones utilizadas en la red.

Un nodo de soporte de GPRS pasarela GGSN (*Gateway GPRS Support Nodes*): Es la interfaz entre la red troncal de GPRS y las redes externas de paquetes, realizando funciones de autenticación y tarificación. Este nodo convierte los paquetes GPRS que vienen del nodo SGSN a un formato especial del protocolo de datos por paquetes PDP (*Packet Data Protocol*) para luego enviarlos por la red PDN que le corresponde. Cuando los paquetes de datos entrantes traen la dirección PDP esta pasa a convertirse en la dirección GSM del usuario destino y para redireccionar los paquetes se llama al nodo SGSN que es responsable, guardando en el nodo GGSN la dirección del SGSN actual del usuario junto con el perfil del registro de localización.<sup>57</sup>



**Figura 10. Arquitectura de una red GPRS (Tomado de GPRS Arquitectura)<sup>58</sup>**

La figura 10 muestra las interfaces que hay entre estos nodos y la red GSM. La misma arquitectura del BSS para el servicio de conmutación de circuitos se utiliza para brindar los servicios GPRS. Los servicios que son basados en conmutación de paquetes son enrutados a través de las interfaces GB hacia el SGSN y Gn

<sup>57</sup> BATES, Regis J. (Bud) Op. Cit. Pág. 177

<sup>58</sup> GPRS Arquitectura. [en línea]. (Consultado en Mayo de 2007). [www.mailxmail.com/cursos/informatica/gsm/capitulo6.htm+arquitectura%2Bgsm&hl=es&ct=clnk&cd=2&gl=us](http://www.mailxmail.com/cursos/informatica/gsm/capitulo6.htm+arquitectura%2Bgsm&hl=es&ct=clnk&cd=2&gl=us)

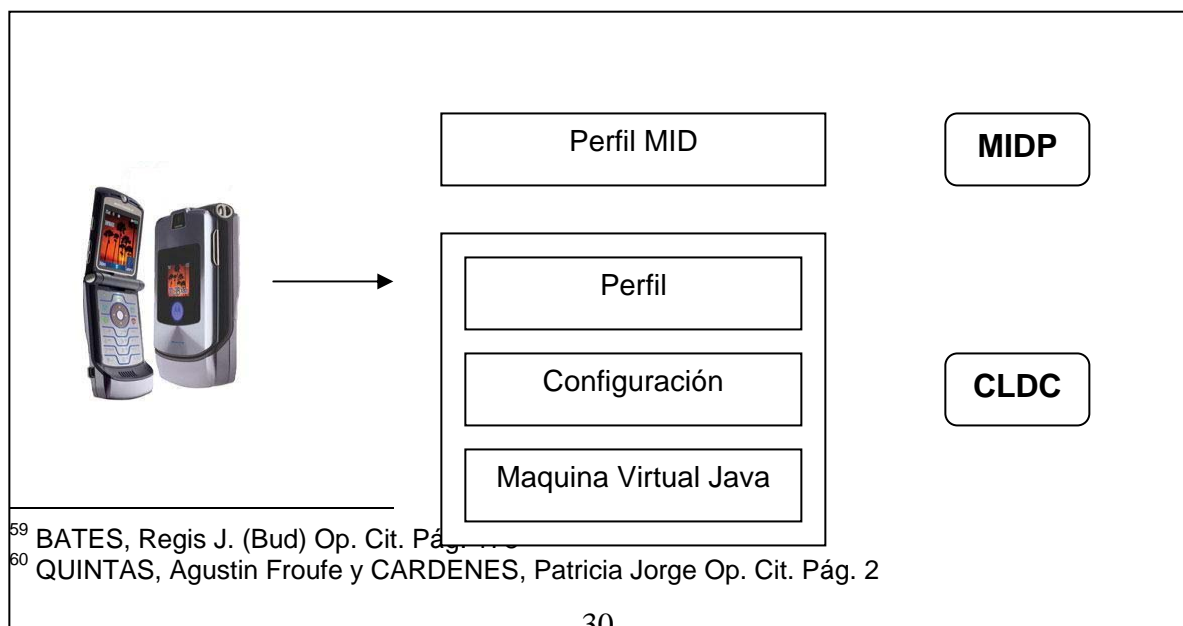
hacia el GGSN antes de que se conecte a la Red de Paquetes de Datos PDN. En el BSS el CPU controla la actividad GPRS de una celda y es el responsable de las funciones de la capa inferior como el Control de Enlace de Radio (Radio Link Control) y Control de Acceso al Medio (Medium Access Control) en la interfaz Um.

Todos los nodos GSN están conectados a través de una red troncal o de transporte GPRS basada en IP y en donde todos los nodos GSN encapsulan los paquetes PDN y se transmiten por medio del Protocolo de túneles GPRS GTP (*GPRS Tunneling Protocol*).<sup>59</sup>

**2.2.5 Tecnologías para el desarrollo de aplicaciones móviles.** Sun Microsystems en la conferencia JavaOne de 1999<sup>60</sup> decidió reunir en varias ediciones las diferentes tecnologías que involucran Java y que van desde las grandes aplicaciones por red hasta las más pequeñas utilizadas en los dispositivos móviles. Para esto Sun creo la Plataforma Java 2 que contiene las tres ediciones según corresponda la aplicación e incluyen cada una el conjunto de APIs y las diferentes herramientas de desarrollo. Entre estas se encuentran Java 2 Standard Edition (J2SE), Java 2 Enterprise Edition (J2EE) y por ultimo Java 2 Micro Edition (J2ME).

- **Aplicaciones Midlet**

**-J2ME (Java 2 Micro Edition).** J2ME es un subconjunto de J2SE y va dirigida hacia la aplicación para dispositivos con pocos recursos como en el caso de los teléfonos celulares, PDAs, buscapersonas, dispositivos para la navegación en coches y entre otros, que no cuentan con mucha capacidad de memoria disponible y con limitaciones en la pantalla grafica, etc. La figura 11 muestra las capas en las que se divide para lograr estas características.



**Figura 11. Arquitectura de capas de J2ME (Tomado de J2ME Java 2 Micro Edition)<sup>61</sup>**

La primer capa de Maquina Virtual Java permite que el dispositivo se adapte y soporte el sistema operativo que se ejecuta en el, implementándole una maquina virtual para cada dispositivo. La segunda capa corresponde a la Configuración del dispositivo donde se definirán las características y librerías de Java para dispositivos específicos. La tercer capa de Perfil esta dirigida hacia la aplicación y definición de APIs disponibles para determinados dispositivos, incluyendo las librerías de las clases que se van a utilizar pero con mucho mas detenimiento que las de la capa de configuración.

Por ultimo la capa de Perfil para Dispositivos de Información Móvil (MIDP) es la que nos permite crear las interfases de los usuarios, las conexiones de red y el manejo de los datos, del sonido, de la seguridad, entre otros. Como se puede ver en el grafico la unión de las tres primeras capas forman las configuraciones CLDC y que junto con la capa del MIDP conforman un entorno de ejecución para las aplicaciones y servicios que llegaran finalmente a ser descargadas por el usuario.

**Configuraciones.** El API de J2ME esta compuesto por configuraciones que son un conjunto de clases Java para poder ejecutar la aplicación para determinados dispositivos móviles con características similares. Actualmente hay dos configuraciones CDC y CLDC.

CDC (Connected Device Configuration) que esta orientada hacia dispositivos que tienen la capacidad de conectarse a una red y que casi siempre son redes inalámbricas, con microprocesadores de 32 bits y con 2 o más Mb de memoria total.<sup>62</sup> “Con CDC se contempla una conexión de red de baja velocidad y con desconexiones habituales, parecidas a las de CLDC, pero además esta tiene en cuenta la existencia en los dispositivos de conexiones rápidas y fiables. Para proveer las funcionalidades, CDC se apoya sobre una maquina virtual Java estándar (JVM)”<sup>63</sup>.

CLDC (Connected Limited Device Configuration) que es dirigida hacia dispositivos con limitación en potencia de calculo y alimentación pobre, con bajo suministro de baterías y con grandes restricciones en la interfaz del usuario<sup>64</sup>, con procesadores de 16 o 32 bits y con al menos 160-192 Kb. de memoria total. En cuanto al mínimo

---

<sup>61</sup> Ibid. Pág. 12

<sup>62</sup> Ibid. Pág. 23

<sup>63</sup> PRIETO MARTIN, Manuel Jesús. Desarrollo de juegos con J2ME. Java 2 Micro Edition. México: Alfaomega Grupo Editor, 2005. 12 p.

<sup>64</sup> QUINTAS, Agustin Froufe y CARDENES, Patricia Jorge Op. Cit. Pág. 23

de memoria de 160 Kb., se van a utilizar 128 Kb. de memoria no volátil para la maquina virtual Java y para las librerías del API de CLDC, y 32 Kb. de memoria volátil para el sistema de ejecución (Java Runtime System). La seguridad que se maneja dentro de CLDC es muy sencilla por medio del modelo sandbox.<sup>65</sup>

**Midlets.** El perfil MIDP tiene una parte muy importante llamada midlet la cual extiende a la clase MIDlet, que es la que ayuda a que una aplicación tenga la posibilidad de controlar los diferentes cambios de estado que se pueden presentar, recuperar propiedades y además permite que se pueda constituir la interfaz entre la ejecución del dispositivo y el código de esta.

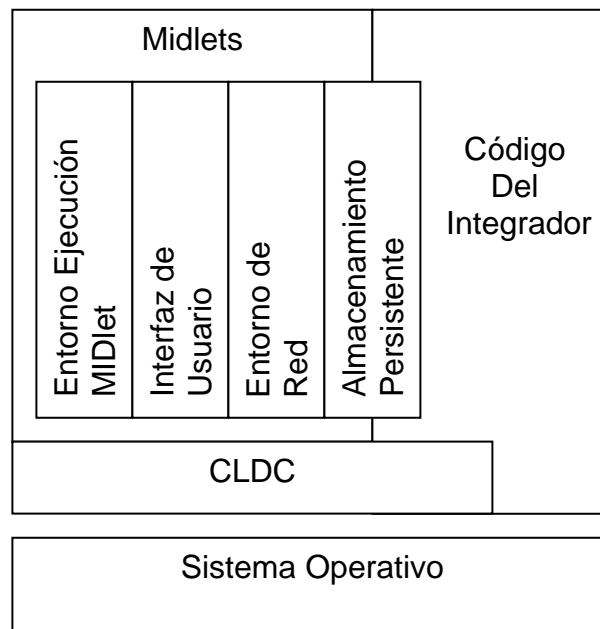


Figura 12. Arquitectura Midlet (Tomado de J2ME Java 2 Micro Edition)<sup>66</sup>

En la figura 12 se puede ver como el midlet puede utilizar de forma directa todo lo que ofrece el perfil MIDP y APIs de la configuración CLDC.

Los midlets manejan unos estados los que se dan durante la transición del ciclo de vida de los midlets desde que se crea hasta que se elimina. Los estados y métodos que se involucran en este proceso se muestran en la figura 13. Se puede ver que cuando el midlet se crea por primera vez y no se ha ejecutado el método *startApp()* es porque el midlet esta en el estado Detenido, al cual también se puede tener acceso cuando se utilizan los métodos *pauseApp()* y *notifyPause()*. En este estado el midlet esta a la espera de cualquier notificación, manteniendo un numero mínimo de recursos.

<sup>65</sup> PRIETO MARTIN, Manuel Jesús. Op. Cit. Pág. 11

<sup>66</sup> QUINTAS, Agustin Froufe y CARDENES, Patricia Jorge Op. Cit. Pág. 32

Si se llega a presentar algún problema en el constructor del midlet, este pasa al estado Destruído y no será posible realizar transiciones a otro estado, ya que estará descartado totalmente. Luego de que se ejecuta *startApp()* pasa al estado Activo en el cual se ejecuta el midlet. El método *startApp()* debe ser incluido en la clase del midlet, ya que en este interactúan todos los recursos que necesita el midlet y se preparan los controladores de los diferentes eventos. Cuando se utiliza el método *pauseApp()* es porque necesitamos detener temporalmente la ejecución del midlet, haciendo que este libere los recursos que mas pueda y guarde el estado en el que se encontraba para poder continuar mas adelante con el.

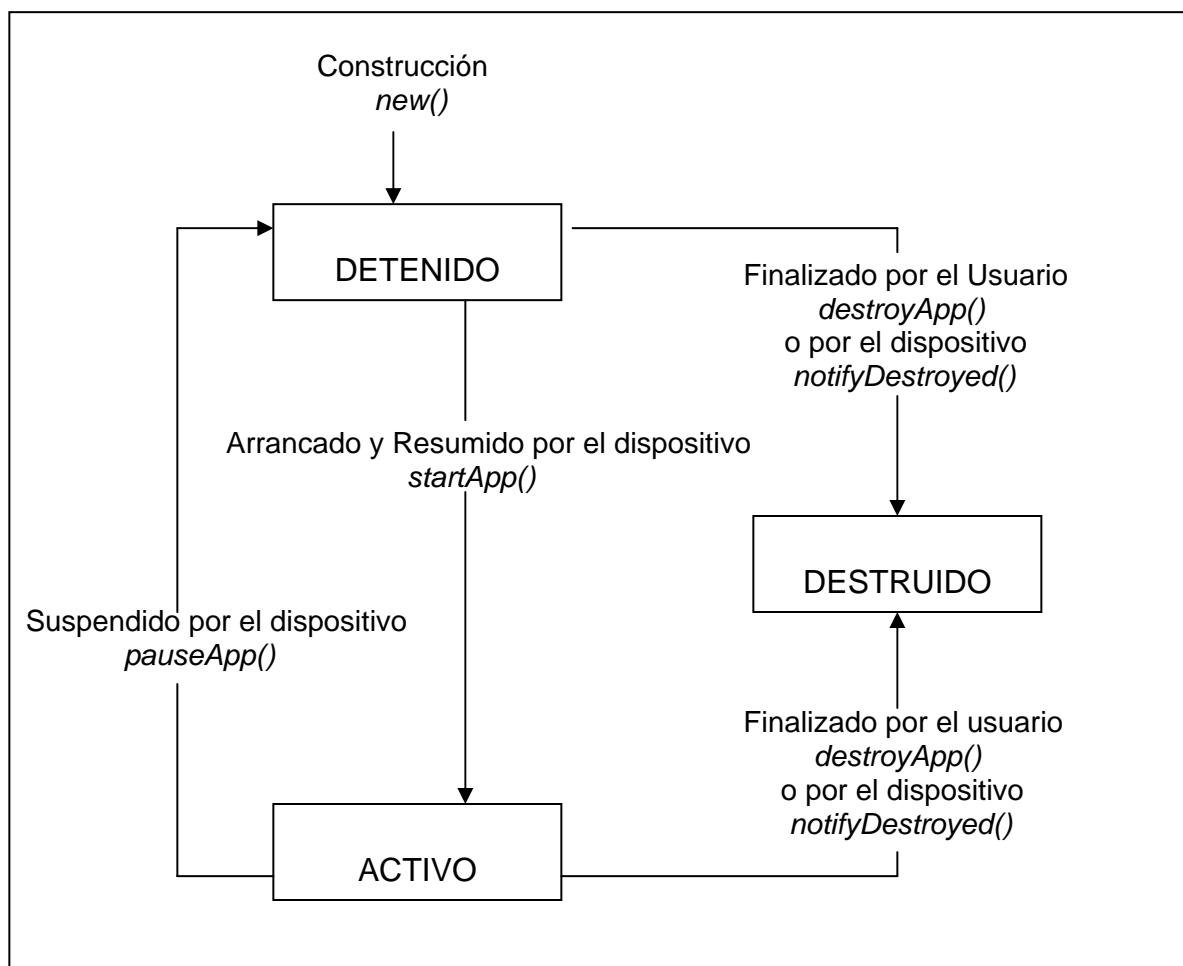


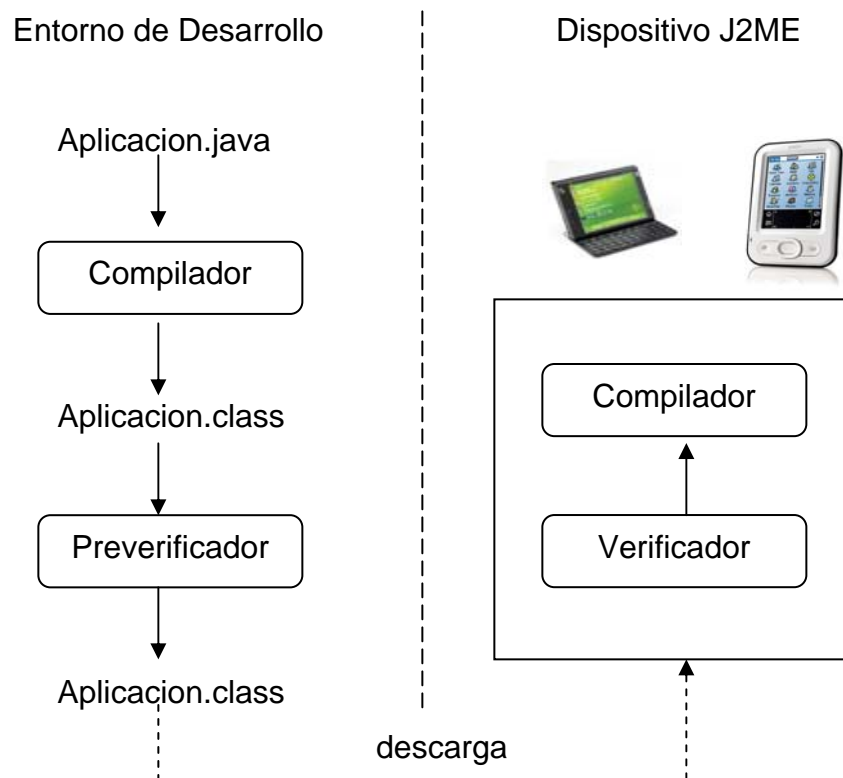
Figura 13. Estados y Métodos de un Midlet (Tomado de J2ME Java 2 Micro Edition)<sup>67</sup>

El método *destroyApp()* es utilizado por el sistema cuando se desea eliminar el midlet o por el mismo midlet antes de terminar su ejecución. Cuando se llama a este método se liberan todos los recursos que el midlet utilizó en la ejecución.

<sup>67</sup> Ibid. Pág. 48



Cabe decir que cuando el midlet es eliminado por el dispositivo se utiliza este método, pero cuando es el midlet el que quiere concluir la ejecución debido a que no hay mas acciones que realizar o porque el usuario desea salir, se llamara al método *notifyDestroyed()*.



**Figura 14. Proceso de verificación de clases en un midlet (Tomado de J2ME Java 2 Micro Edition)<sup>68</sup>**

Cuando se realiza la ejecución de la aplicación J2ME, de acuerdo al entorno definido por MIDP, basados en los midlets o suite<sup>69</sup> de midlets e implementando el Perfil MIDP, el entorno tiene que ser compartido por todos los midlets de la distribución y por cualquier otro midlet que quiera interactuar con un midlet de la suite.<sup>70</sup>

<sup>68</sup> Ibid. Pág. 55

<sup>69</sup> "suite" es una colección de midlets que están agrupados en un archivo JAR

<sup>70</sup> Ibid. Pág. 53

En la figura 14 se muestra cómo el proceso de verificación de las clases se reparte entre el entorno de desarrollo y el dispositivo móvil que va a ejecutar las clases.

Lo nuevo en esta parte del ciclo de desarrollo es el proceso de preverificación que es el que se encarga de comprobar el bytecode<sup>71</sup> Java antes de ejecutarse y si es correcto este no se modifica para que pueda ser ejecutado en cualquier máquina virtual y los resultados que arroja son anotados como atributos en los ficheros *.class*. Los midlets son empaquetados en ficheros con extensión *.jar* y generalmente se requiere de más información para poder poner en marcha estas aplicaciones, y dicha información se almacena en el fichero de *manifiesto* que va incluido en el fichero *.jar* y en un fichero descriptor con extensión *.jad*. Un fichero *.jar* típico se compone de clases del Midlet, clases de soporte, recursos (imágenes, sonidos...) y manifiesto (fichero *.mf*).<sup>72</sup>

**RMS.** El perfil MIDP define un conjunto de clases que van a tener como propósito controlar el almacenamiento de los datos para que se pueda guardar y recuperar la información en el mismo dispositivo móvil y formando así un base de datos en el propio dispositivo llamado RMS (*Record Management System*). Este sistema de almacenamiento permite que los datos se puedan mantener a través de la ejecución de un midlet o que se puedan compartir los datos que están en el RMS entre distintos midlets que están en una misma aplicación.

RMS maneja los registros (*record*) y por medio de un identificador único (*record id*) y una serie de bytes forman todo lo que es la información en cada fila de la base de datos, denominada almacén de registros (*record store*). Cada bloque de datos es un registro dentro de la zona de almacenamiento y están relacionados por medio de un modelo de lista enlazada en donde cada registro tiene que mantener un enlace con el siguiente registro. También se tiene una cabecera que mantiene un enlace con el primer bloque de datos y otro con el primer bloque de la zona de espacio libre para que se pueda reproducir el funcionamiento de la base de datos.

Además de esto la cabecera también puede contener el número de registros en la zona de almacenamiento, el número de versión, la última fecha de modificación y un identificador que se le asignara al siguiente registro creado en la zona de almacenamiento.<sup>73</sup>

- **Tecnologías para el desarrollo de aplicaciones Web**

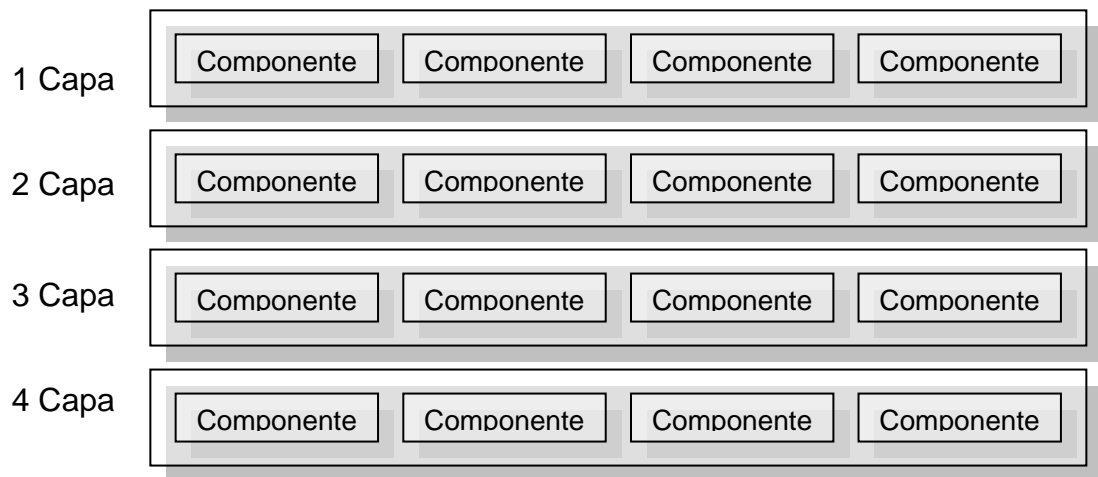
---

<sup>71</sup>“bytecode” es un programa muy pequeño que es generado por Java y que hace que se ocupe muy poca memoria en el dispositivo móvil para que luego puedan ser descargados por medio de una pagina Web o enviados por correo al mismo dispositivo.

<sup>72</sup>PRIETO MARTIN, Manuel Jesús. Op. Cit. Pág. 14

<sup>73</sup> Ibid. Pág. 272

**-J2EE (Java 2 Enterprise Edition).** Java 2 Edición Empresarial es una mejora de J2SE la cual contiene las API que se necesitan para construir las aplicaciones con arquitecturas multicapa. J2EE simplifica la creación de las aplicaciones empresariales ya que toda la funcionalidad esta en los componentes de J2EE, permitiendo así que los diseñadores y programadores puedan distribuir la aplicación de acuerdo a sus funciones, en los componentes del servidor. Los componentes de las aplicaciones que se construyen con J2EE se comunican entre si por medio de métodos estándar de comunicación como lo son HTTP, SSL, XML, RMI e IIOP<sup>74</sup>.



**Figura 15. Arquitectura J2EE multicapa (Tomado de J2EE Manual de Referencia)<sup>75</sup>**

La arquitectura de J2EE multicapa, consta de cuatro capas que son: la capa cliente conocida como capa de presentación o aplicación, la capa Web, la capa de Enterprise JavaBeans conocida como capa de transporte y la capa de sistemas de información empresarial. Cada una de estas capas se va a enfocar en proporcionar a una aplicación un tipo específico de funcionalidad.

La primera capa es la Capa Cliente que esta formada por los programas que interactúan con el usuario, los cuales piden información al usuario y la convierten en peticiones, que son enviadas a un componente para que procese esta información para luego devolver el resultado al programa cliente.

La segunda capa es la Capa Web que le proporciona a las aplicaciones J2EE la funcionalidad de Internet, en la cual los componentes para poder recibir peticiones y enviar respuestas utilizan HTTP sin importar en que capas puedan estar los clientes.

<sup>74</sup> KEOGH, Jim. J2EE. Manual de Referencia. Madrid: McGraw-Hill, 2003. 803 p.

<sup>75</sup> Ibíd. Pág. 30

La tercera capa es la Capa de Enterprise JavaBeans (EJB) la cual contiene toda la lógica del negocio de las aplicaciones J2EE y en donde se encuentran uno o más Enterprise JavaBeans las cuales llaman al cliente en forma directa. Esta capa es la primordial para todas las aplicaciones J2EE ya que los EJB que se encuentran en esta capa le permitirán a las posibles instancias de la aplicación tener acceso a la lógica y datos del negocio de forma concurrente sin que se vea afectado su rendimiento.

Y por último esta la capa de sistemas de información empresarial o EIS la cual conecta a la aplicación J2EE con los recursos y sistemas heredados que están en la red de la empresa. En el EIS es donde se conecta a las aplicaciones J2EE directa o indirectamente con las diferentes tecnologías que están en el sistema de misión crítica que mantienen el funcionamiento de la empresa como lo son el DBMS y los mainframes.<sup>76</sup>

**Java y XML.** Cuando las empresas necesitan gestionar información que no es tan detallada se puede hacer utilizando la forma de documentos, por medio de XML que permite que la empresa pueda identificar los componentes del documento por medio de etiquetas y atributos. XML no depende de la plataforma y además permite extenderse sobre los otros sistemas que ya existen haciendo que su funcionamiento sea muy bueno. Un componente J2EE puede crear, leer o manipular un documento XML por medio de interfaces de programación de aplicaciones. Una aplicación J2EE puede generar un documento XML de acuerdo con unas reglas de negocio que se codifican en uno o más componentes J2EE. Generalmente estos componentes son servlets, Java ServerPages (JSP), Enterprise JavaBeans o componentes del Servicio de Mensajería de Java.

Para generar un documento XML, una JSP o servlet envían a su salida cadenas que contienen etiquetas y directivas XML. La salida del servlet o JSP se envía a la conexión HTTP que invoca al servlet, el cual casi siempre proviene del navegador<sup>77</sup>.

**Servlets.** Un servlet es un programa del servidor que llama la interfaz de usuario u otro componente J2EE y que contiene la lógica del negocio para procesar una petición. El programa procesa la petición y devuelve un conjunto de datos explícitos e implícitos. Los datos explícitos hacen referencia a la información del cliente que son introducidos en la interfaz de usuario, como por ejemplo una clave de acceso o una identificación de usuario, que pueden ser enviados por un navegador, un applet o por una aplicación cliente.

El procesamiento de una petición se puede hacer llamando a un servlet, un Enterprise JavaBean y algún sistema heredado. El resultado de procesar la petición se envía al cliente en forma de datos explícitos que generalmente es por medio de una página HTML que se forma dinámicamente por el servlet.<sup>78</sup>

---

<sup>76</sup> Ibid. Pág. 32

<sup>77</sup> Ibid Pág. 309

<sup>78</sup> Ibid Pág. 338

**JavaServer Pages (JSP).** Un JSP es un programa del servidor con el cual se puede ofrecer un servicio Web que depende de la aplicación J2EE cuando un cliente lo llama. La JSP procesa la petición por medio de la lógica que incluye, llamando a otros componentes que están creados con tecnología de servlets, Enterprise JavaBeans o con otra tecnología, para que luego de procesar la petición, pueda enviar el resultado al cliente. Una JSP se escribe en HTML, XML o en el formato del cliente, entremezclado con elementos de código, directivas y acciones escritas en Java y con sintaxis de JSP.<sup>79</sup>

Una JSP proporciona las mismas características que ofrece un servlet ya que en el momento en que el cliente llama a la JSP por primera vez este se convertirá en un servlet. Cuando se solicita una JSP y cuando la JSP termina en forma normal se llaman en forma automática tres métodos que son `jspInit()`, `jspDestroy()` y `service()`. El método `jspInit()` es igual que el método `init()` de un servlet y de un applet que se utiliza antes de enviar la primera petición al servlet, inicializando los objetos y variables que se van a utilizar a lo largo de la JSP. El método `jspDestroy()` es el mismo `destroy()` del servlet y se llama en forma automática cuando la JSP termina en forma normal. El método `destroy()` limpia y libera los recursos utilizados por la JSP durante su ejecución. Y por ultimo el método `service()` que se llama automáticamente es el encargado de enviar el resultado de la JSP por medio de HTTP.<sup>80</sup>

**Invocación remota de métodos (RMI).** Los objetos que utiliza una aplicación J2EE no tienen que ejecutarse necesariamente en la misma JVM ya que la aplicación de J2EE y sus componentes pueden llamar estos objetos que están situados en otra JVM mediante la Invocación remota de métodos de Java RMI (Remote Method Invocation). RMI se utiliza para conectar un cliente con un servidor, en donde el cliente es una aplicación que solicita un servicio de un objeto distinto para atender a una petición y el servidor crea un objeto y lo pone a disposición de los clientes. Un cliente contacta con el servidor para obtener una referencia y llamar al objeto mediante RMI.<sup>81</sup>

Para poder localizar un objeto remoto un cliente puede utilizar el registro de nombres de RMI y le pasa una cadena que hace referencia al objeto remoto. El sistema de RMI devuelve una referencia al objeto remoto, la cual el cliente puede utilizar como si el objeto estuviera en JVM local. Para que el objeto este disponible a los clientes remotos hay que diseñar el objeto, compilarlo y ponerlo disponible a los clientes por medio de la red, además de definir toda la lógica del negocio del objeto y la interfaz remota para este<sup>82</sup>. Los métodos que puede llamar un cliente se llaman *métodos de cliente* y además de estos, se pueden definir otros métodos que puedan dar soporte al procesamiento de los métodos que invoca el cliente y que se conocen como *métodos de servidor*.

---

<sup>79</sup> Ibid Pág. 365

<sup>80</sup> Ibid Pág. 366

<sup>81</sup> Ibid Pág. 463

<sup>82</sup> Ibid Pág. 464

El servidor de un objeto remoto consiste de una interfaz remota que define la percepción del objeto que tiene el cliente y del objeto que la implementa ya que este proporciona la lógica del negocio que atiende las peticiones de los clientes cuando estos los llaman. El programa cliente llama al objeto remoto el cual realiza operaciones complejas específicas de la aplicación y devuelve la información requerida al cliente.<sup>83</sup>

---

<sup>83</sup> Ibid Pág. 467

## **2.3 MARCO CONTEXTUAL**

El presente proyecto se lleva a cabo dentro de la línea de investigación en Telecomunicaciones de la Facultad de Ingeniería de Sistemas de la Universidad El Bosque. Su propósito es desarrollar el software que permita controlar un hardware domótico ya construido que simula las luces de una casa en escala reducida. El desarrollo se enmarca dentro del diseño de un midlet para un teléfono celular como dispositivo de interfaz entre el usuario y el hardware domótico. Para esto se utilizarán conceptos de sistemas distribuidos e ingeniería de software y la ayuda de herramientas de desarrollo con licencia de uso libre, conociendo sus ventajas de escalabilidad, flexibilidad, confiabilidad y bajos costos.

### 3. REQUERIMIENTOS

#### 3.1 RESUMEN EJECUTIVO

En la presente investigación se desarrolla una aplicación midlet que tiene la función de controlar el encendido y apagado de las luces de un hardware domótico. Este hardware representa a escala reducida el ambiente de una casa y todas las partes que puedan tener un medio de iluminación como la cocina, el cuarto principal, el baño, el garaje entre otros.

Para que un usuario desde el midlet pueda interactuar con el hardware domótico, se deben configurar varios parámetros. Esta función es hecha por un administrador del sistema que a través de una página Web gestiona todo el control que tienen los usuarios sobre el hardware domótico. A continuación se describen dichos parámetros.

**Locaciones:** las locaciones son lugares físicos (casa, oficina, finca, etc.) las cuales tienen dispositivos de iluminación domóticos. Básicamente las locaciones es lo primero que debe crear un administrador pues a estas pertenecerán los dispositivos y los grupos de usuarios. Para la presente investigación se toma como locación una maqueta de una casa.

**Dispositivos:** una locación puede tener varios dispositivos domóticos conectados, en este caso, dispositivos de iluminación. El administrador puede crearlos y asignarlos a la locación que desee.

**Usuarios:** son las personas que controlan los dispositivos domóticos de una determinada locación a través del midlet. Cuando el administrador crea los usuarios, debe asignarlos a un grupo de usuarios.

**Grupos de usuarios:** debido a que no todos los usuarios tienen los mismos permisos sobre los dispositivos de iluminación, es necesario que pertenezcan a un grupo de usuario; para mayor seguridad y administración, estos permisos se dan exclusivamente al grupo. Cada grupo puede ejercer sobre los dispositivos sólo las acciones que el administrador establezca.

**Acciones y permisos:** sobre los dispositivos de iluminación se pueden ejercer dos acciones principales las cuales son *apagar el dispositivo* y *encender el dispositivo*; el administrador dará permisos a los grupos de usuarios para que puedan o no realizar estas acciones. Adicionalmente, existen dos permisos que el administrador puede dar a los grupos, los cuales son *programar*, donde el usuario puede establecer una fecha y hora para apagar o encender la luz automáticamente y el permiso de *ver estado* que le permite saber si la luz se encuentra apagada o encendida. Este último permiso es importante pues no todos los grupos de usuarios pueden ver todas las luces de la locación.



Basados en estos parámetros, el usuario a través del midlet accede con un nombre de usuario y contraseña. El sistema muestra las locaciones a las cuales él puede tener acceso; posteriormente, dependiendo de la locación seleccionada, el sistema muestra los dispositivos que pertenecen a esta. Si el grupo del usuario no tiene permiso de *ver estado*, no se listará en la pantalla dicho dispositivo. Luego, al escoger un dispositivo, el usuario puede realizar las acciones de *encender*, *apagar* o *programar*, siempre y cuando su grupo tenga esos permisos.

En los requerimientos funcionales que se describen a continuación, se dará un panorama más detallado acerca de estos parámetros y de cómo el usuario a través del midlet controla las luces del hardware domótico.

### 3.2 REQUERIMIENTOS FUNCIONALES

A continuación se describirán los requerimientos funcionales del sistema mediante los diferentes diagramas de casos de uso. Primero se representarán para el actor *Administrador* el cual interactúa solamente desde la página Web y es el que configura y administra todo el sistema; posteriormente se mostrarán los diagramas para el actor *Usuario* encargado del acceso al hardware domótico a través del midlet.

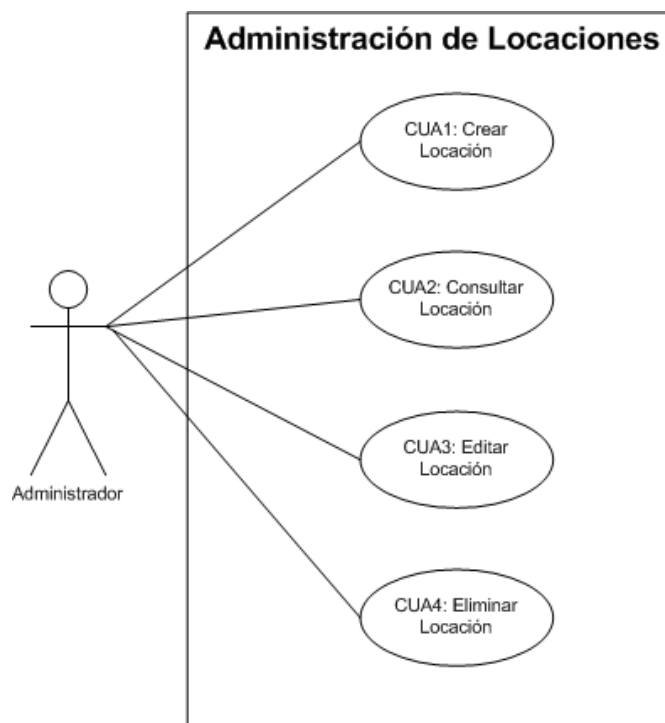


Figura 16. Diagrama de casos de uso Administración de Locaciones

Para mayor facilidad de entendimiento, se han dividido los diagramas según la tarea que realiza cada actor. El *Administrador* tiene 4 tareas: *administrar locaciones*, *administrar dispositivos*, *administrar usuarios* y *administrar permisos sobre dispositivos*. Entiéndase por administrar a crear, consultar, editar y eliminar. El *usuario* por su parte tiene la tarea de *interactuar con el hardware domótico*.

La figura 16, muestra el diagrama de casos de uso de la tarea *administrar locaciones*. Una locación es lo primero que debe crear el administrador pues a estas pertenecerán los dispositivos de iluminación y los grupos de usuarios.

La descripción de cada uno de los casos de uso de este primer escenario para el actor *Administrador* se muestra desde la Tabla 1 hasta la Tabla 4 en el Apéndice A.

La figura 17 muestra el diagrama de casos de uso de la tarea *administrar dispositivos*. Cada locación puede tener varios dispositivos de iluminación los cuales podrán ser controlados a través del midlet por los usuarios. El administrador al crear cada dispositivo los asigna a una determinada locación; no puede existir el mismo dispositivo en dos locaciones diferentes.

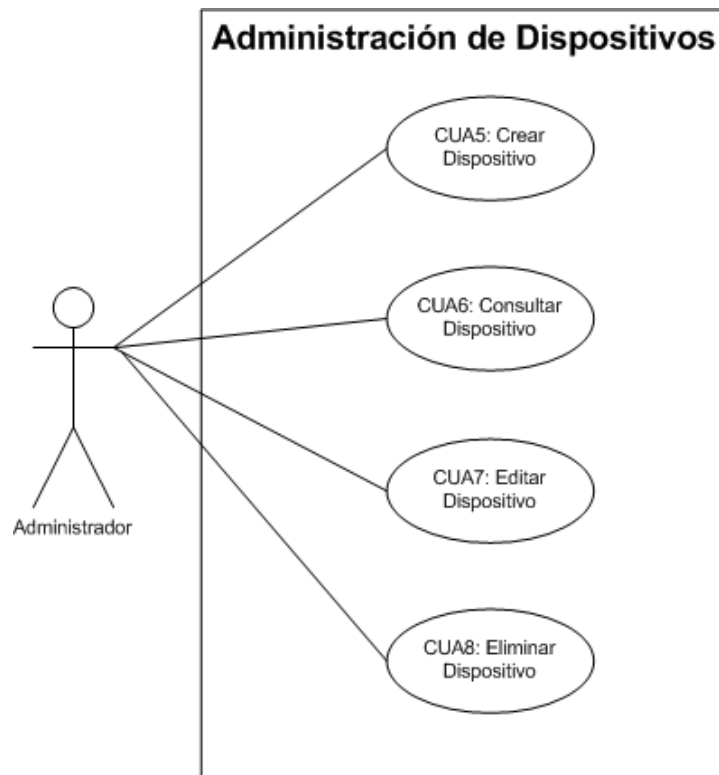
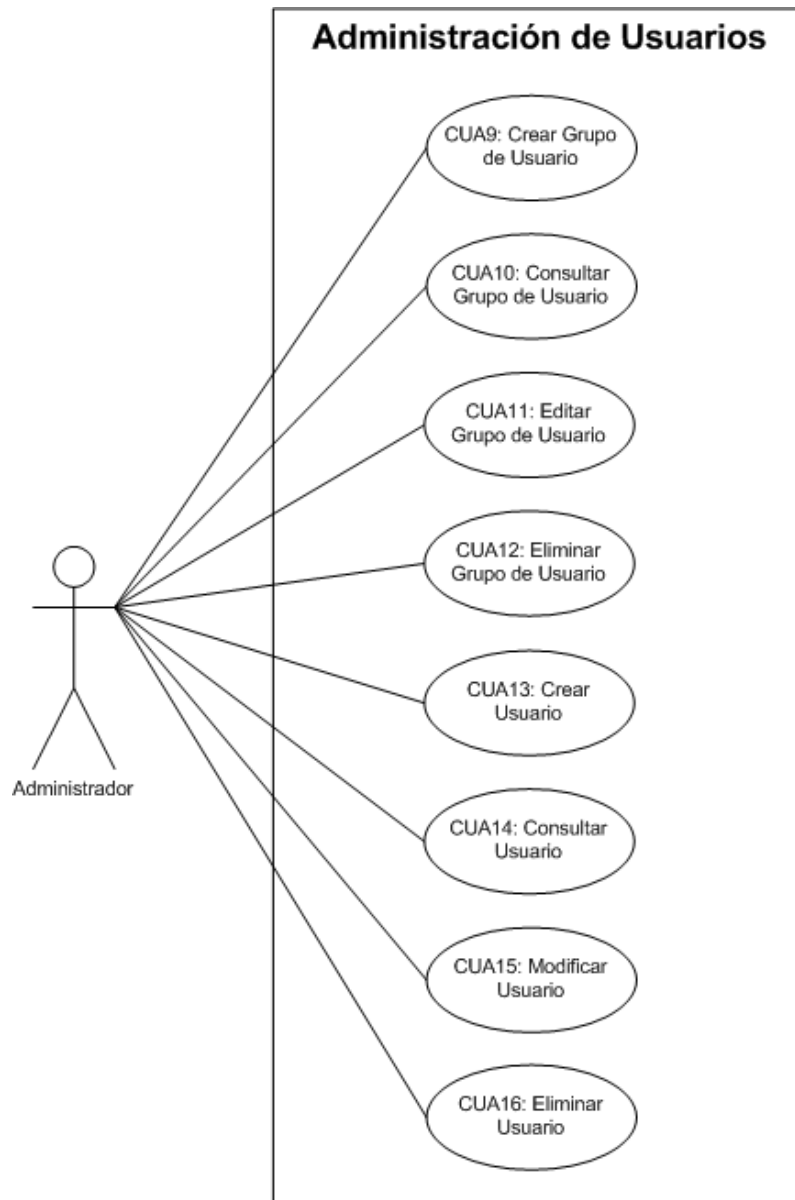


Figura 17. Diagrama de casos de uso Administración de Dispositivos

La descripción de cada uno de los casos de uso de del escenario *Administración de Dispositivos* del actor *Administrador* se muestra desde la Tabla 5 hasta la Tabla 8 en el Apéndice A.

En la figura 18 se muestra el diagrama de casos de uso de la tarea *administrar usuarios*. Cada usuario debe pertenecer a un grupo de usuario al cual el administrador dará permisos sobre los dispositivos de iluminación. Un usuario solo puede estar en un grupo y cada grupo solo puede pertenecer a una locación.

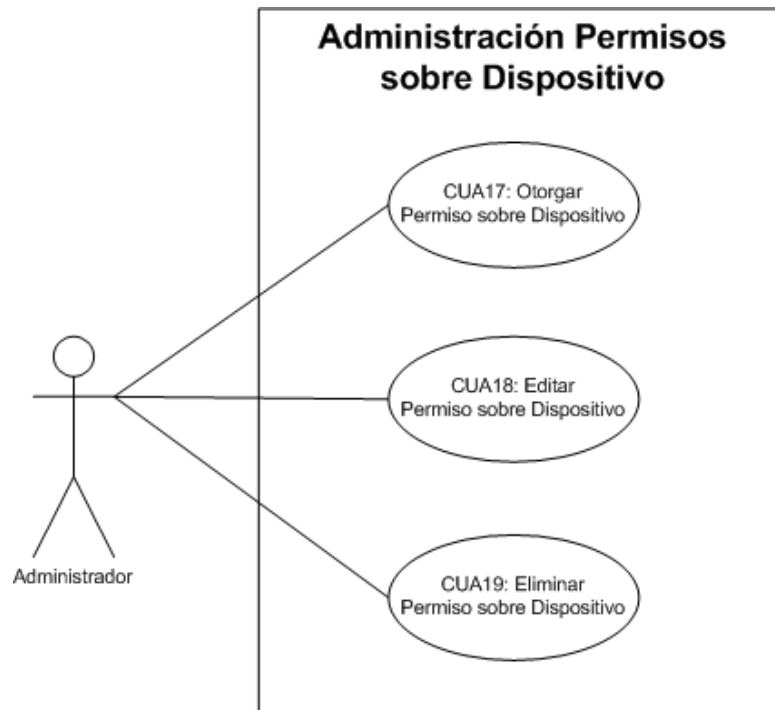


**Figura 18. Diagrama de casos de uso Administración de Usuarios**

La descripción de cada uno de los casos de uso del escenario *Administración de usuarios* del actor *Administrador* se muestra desde la Tabla 9 hasta la Tabla 16 en el Apéndice A.

En la figura 19 se muestra el diagrama de casos de uso de la tarea *administrar permisos sobre dispositivos*. El administrador puede otorgar permisos a los grupos

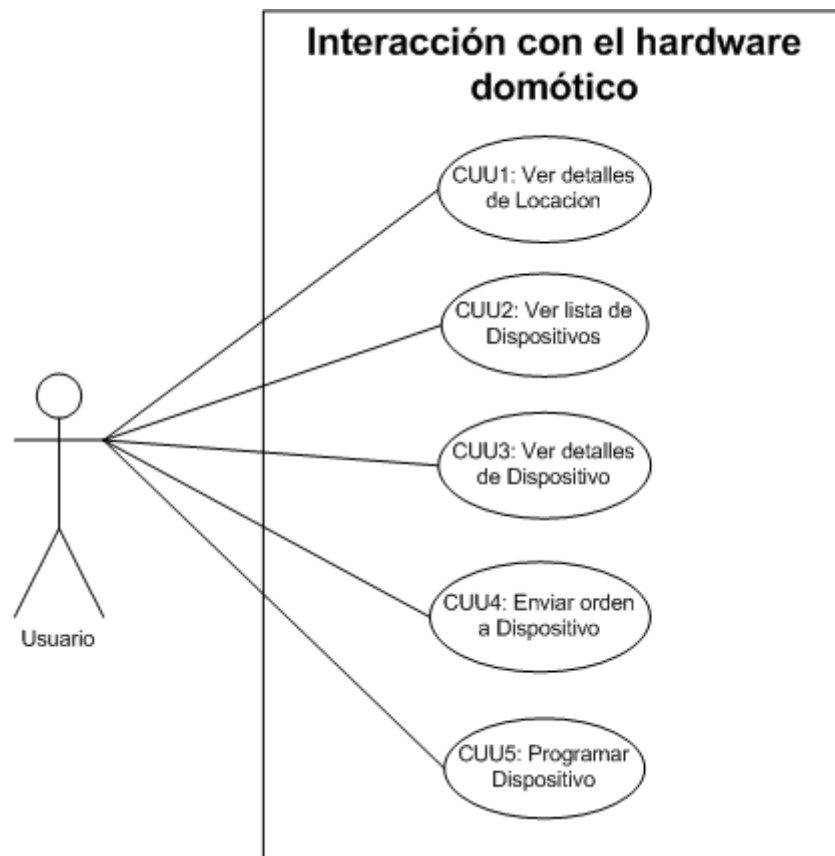
de usuario de *encender, apagar, ver estado o programar* los dispositivos de iluminación.



**Figura 19. Diagrama de casos de uso Administración de permisos sobre dispositivo**

La descripción de cada uno de los casos de uso del escenario *Administración de permisos sobre Dispositivo* del actor *Administrador* se muestra desde la Tabla 17 hasta la Tabla 19 en el Apéndice A.

A continuación se presentan los casos de uso del actor *usuario*. Como se mencionó anteriormente, el *usuario* tiene la tarea de *interactuar con el hardware domótico* la cual se muestra en la figura 20. El usuario puede escoger una locación a la cual ingresar y luego de haber seleccionado un dispositivo de iluminación, envía la orden de encenderlo, apagarlo o programarlo. Al enviar una orden de ejecución, el controlador del hardware domótico (computador al que se encuentra conectada la tarjeta X-10), se notifica que hay pendiente una orden por ejecutar y la lleva a cabo.



**Figura 20. Diagrama de casos de uso Interacción con el hardware domótico**

La descripción de cada uno de los casos de uso del escenario *Interacción con el hardware domótico* del actor *Usuario* se muestra desde la Tabla 20 hasta la Tabla 24 en el Apéndice A.

### 3.3 REQUERIMIENTOS NO FUNCIONALES

La Tabla 1 muestra los principales requerimientos no funcionales de hardware, seguridad, software, entre otros, que se tendrán en cuenta al momento de desarrollar la aplicación Web y el midlet.

**Tabla 1. Requerimientos no funcionales del sistema**

Id	Requerimiento	Tipo	Prioridad	Casos de uso afectados
1	El celular debe soportar aplicaciones .JAD. y tener acceso a Internet (Pare el presente	Hardware	Alta	CUU1, CUU2, CUU3, CUU4, CUU5

	proyecto se simulará en MobilityPack de NetBeans)			
2	El computador del usuario debe estar encendido y conectado a Internet cuando se requiera ejecutar una orden.	Disponibilidad	Alta	CUU5
3	Todos los usuarios deben pertenecer a un grupo de usuario.	Seguridad	Alta	CUA13
5	La interfaz en el celular debe tener la menor cantidad de pantallas e imágenes posible.	Funcionalidad, Calidad, Desempeño	Media	CUU1, CUU2, CUU3, CUU4, CUU5
6	La aplicación Web debe ser desarrollada en JSP	Diseño	Alta	Todos los casos de uso del <i>administrador</i>
7	La aplicación debe validar los datos antes de ser guardados en la base de datos mediante mensajes de error o confirmación.	Implementación y Diseño	Baja	CUA1, CUA3, CUA4, CUA5, CUA7, CUA8, CUA9, CUA11, CUA12, CUA13, CUA15, CUA16, CUA18, CUA19
8	La aplicación midlet no debe superar 5Mb de tamaño.	Portabilidad, Flexibilidad	Baja	CUU1, CUU2, CUU3, CUU4, CUU5
9	El hardware domótico y el computador del usuario deben tener un puerto serial disponible.	Hardware	Alta	CUU4, CUU5

## 4. METODOLOGÍA

Para la realización del presente proyecto se tomaron como base las cuatro etapas del ciclo de desarrollo de software: *análisis de requerimientos*, *análisis detallado*, *diseño del sistema* y *pruebas de la aplicación* las cuales se describen a continuación.

*Análisis de requerimientos:* en esta etapa se estudiaron los diferentes escenarios que se pueden presentar en la vida real en el momento en que un usuario quisiese interactuar con su sistema domótico. Para esto, se hicieron investigaciones sobre los productos actuales existentes y las diferentes alternativas que ofrecen las compañías de domótica para que dichos productos puedan ser controlados de forma remota. Así mismo, se hizo una recopilación de información de trabajos e investigaciones realizadas en universidades alrededor del mundo para tener una referencia de en qué forma se han integrado los desarrollos de aplicaciones para domótica en ambientes distribuidos.

El hardware domótico, debido a que ya había sido construido, no implicó requerimientos significativos; sin embargo fue necesario realizar pruebas de envío de instrucciones a través del puerto serial del computador las cuales se describen en el capítulo de pruebas.

Al tener esta información se formularon los requerimientos funcionales del sistema dentro de los cuales, se vio la necesidad de crear una página Web para que un usuario administrador pudiera configurar el sistema, ya que hacer esta operación por medio del midlet resultaría engorroso. La aplicación midlet entonces, se destinó para ejecutar las operaciones de encendido, apagado y programación de las luces del hardware domótico.

El funcionamiento de la aplicación se planteó de la siguiente forma: existe un usuario administrador el cual, por medio de una página Web, configura el sistema. El administrador crea las locaciones, los dispositivos de iluminación de esas locaciones, los grupos de usuario y los usuarios que pueden tener acceso a esos dispositivos. Un usuario, desde el midlet, puede controlar los dispositivos a los cuales el administrador le haya dado permiso. Para esto, se crearán dos servlets: uno para la aplicación Web y otro para la aplicación midlet. Por su parte, el controlador domótico (computador al que está conectado el hardware) periódicamente enviará peticiones al servidor preguntando si tiene alguna instrucción nueva que ejecutar. Si el servidor responde que si, se llevará a cabo la orden; si responde que no, seguirá preguntando.

*Análisis detallado:* en la siguiente etapa se fueron identificando los requerimientos de plataforma de hardware, software y base de datos que se utilizarían para el desarrollo de la aplicación. Para seguir un modelo distribuido, se planteó una

arquitectura en donde se incluyeran (1) un servidor de aplicaciones el cual autenticaría los usuarios y enviaría las diferentes instrucciones que recibiera del teléfono celular al hardware domótico (en esta parte del análisis se propuso que no solo se pudiera controlar un solo hardware domótico, sino que el sistema estuviera en la capacidad de controlar n); (2) uno de bases de datos que guardaría los datos de usuarios, dispositivos, locaciones, entre otros y (3) un controlador domótico; fue necesario este último pues debía existir un medio que pudiera recibir instrucciones desde el servidor y al mismo tiempo enviarlas de regreso por lo que se planteo utilizar sockets. Sin embargo, luego de realizar varias pruebas se llegó a la conclusión que implementar RMI en vez de un socket resultaría más fácil y confiable.

*Diseño del sistema:* para el desarrollo de la parte midlet y la aplicación Web se utilizó Java por su funcionalidad para diseñar aplicaciones para dispositivos celulares y por la opción de crear Servlets para la parte Web. La base de datos que se utilizó fue MySQL y Tomcat como servidor de la aplicación, ambos por ser compatibles con las aplicaciones realizadas en Java. En esta parte se definieron además, los diferentes procesos que intervendrían desde que el usuario entrara en la aplicación midlet hasta que la confirmación de que “la tarea fue hecha” se mostrara en la pantalla de su celular. Para esto se crearon diferentes bocetos de flujogramas luego de los cuales se tomaron decisiones acerca de cuáles procesos eran realmente necesarios y cuáles podían ser modificados o no tenidos en cuenta en la realización del software tanto en la parte Web como en el midlet; así entonces, se diseñó el diagrama de clases para determinar los parámetros y acciones a utilizar de tal forma que siguieran la estructura del flujograma; posteriormente este diagrama de clases fue convertido en el diagrama de entidad relación.

Para que la aplicación midlet fuera fácil de utilizar y no implicara que el celular necesitara espacio considerable de almacenamiento, se diseñaron la menor cantidad de ventanas posibles, sencillas, sin mucho color y sin imágenes. Para este desarrollo se utilizó Mobility Pack de NetBeans por su facilidad al momento de crear esta clase de aplicaciones. La página Web se desarrolló en DreamWeaver con el mismo concepto de la menor cantidad de enlaces posibles y que no tuviera demasiadas imágenes que pudieran hacer que su visualización fuera demorada en un navegador con una conexión a Internet estándar.

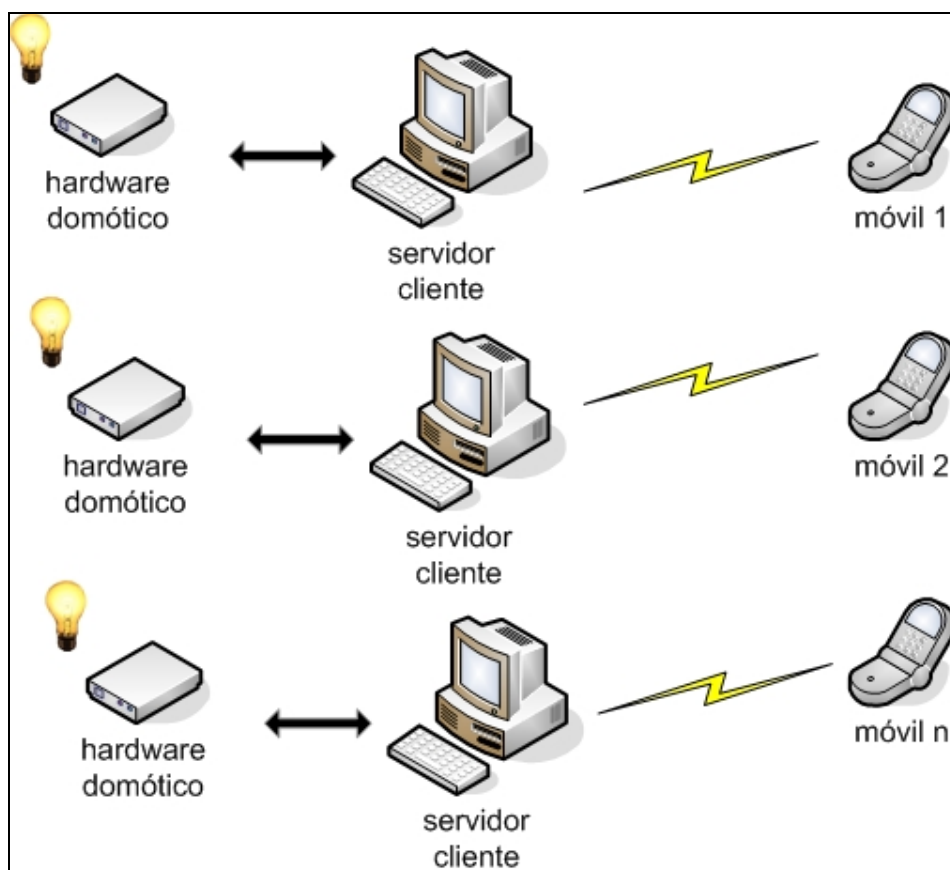
*Pruebas a la aplicación:* la última etapa consistió en las pruebas del sistema. En primer lugar se probó la página Web en donde se tomó un usuario administrador para ingresar al sistema, crear locaciones, dispositivos y usuarios y se analizaron los errores al momento de ejercer acciones sobre la base de datos los cuales fueron corregidos. Igualmente se probó la consistencia de los datos y que el servlet cumpliera con los requerimientos del sistema. Posteriormente se probó el midlet y se validaron los datos que retornaba desde el servlet, así como la validación de los permisos sobre los dispositivos de iluminación. Otra prueba que se hizo fue la del hardware domótico, la cual consistió en enviar a través del



puerto serial una orden para que fuera efectuada. Los resultados se detallan mas adelante en la sección de pruebas.

## 5. DISEÑO GLOBAL

La mayoría de los sistemas domóticos que son controlados remotamente tienen una arquitectura en la que el computador cliente hace las veces de servidor. Generalmente la aplicación es controlada por un mismo usuario el cual ingresa a una página Web o a través de su dispositivo móvil por medio de WML y ejecuta acciones sobre los dispositivos domóticos. La figura 21 indica dicha arquitectura.



**Figura 21. Arquitectura de los sistemas domóticos tradicionales**

Para que el sistema funcione bajo un ambiente distribuido, debe existir una arquitectura que permita que varios usuarios puedan desde sus celulares ingresar al sistema y controlar el hardware domótico. Así mismo, debe integrarse un servidor de aplicaciones el cual estará en la capacidad de administrar de 1 a n controladores domóticos y almacenar la información en la misma base de datos.

A continuación se describirán la arquitectura de hardware y de software del sistema que se considerarán para llevar a cabo tal fin.

## 5.1 ARQUITECTURA DE HARDWARE

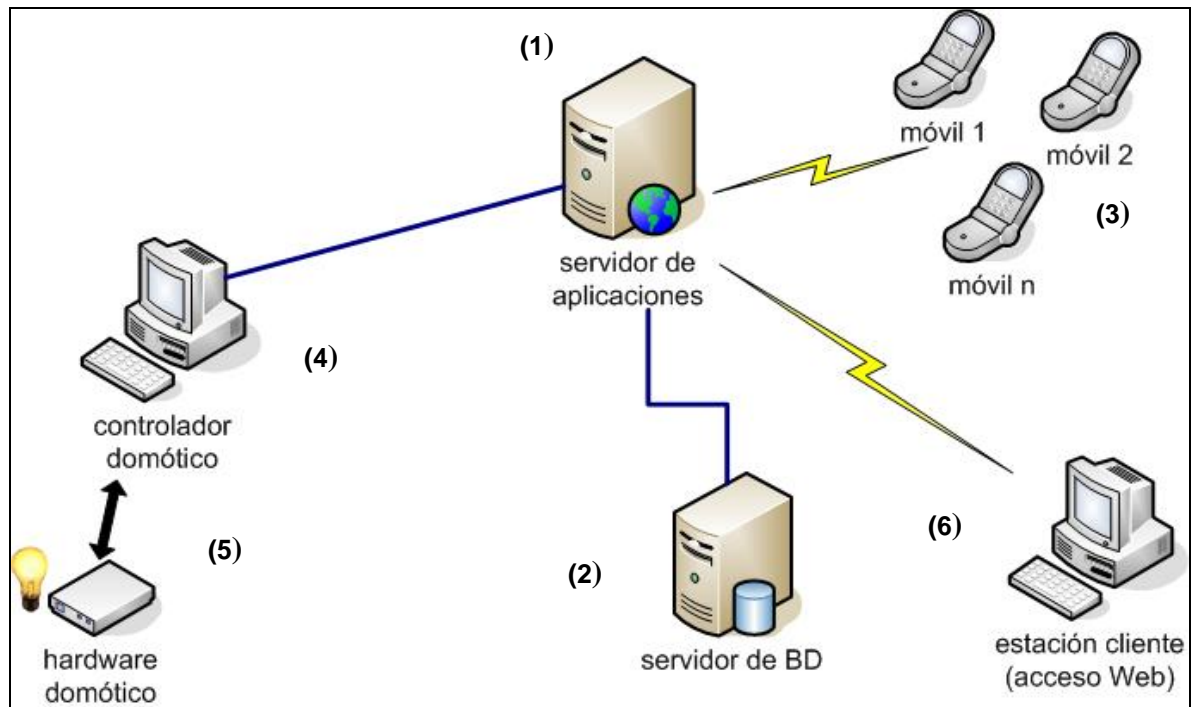


Figura 22. Arquitectura de Hardware

La figura 22 ilustra los componentes que integran la arquitectura de hardware la cual, está conformada por (1) un servidor de aplicaciones encargado de alojar los servlets a los cuales el sistema accede, en este caso un servlet para la página Web y otro para la aplicación Midlet; (2) un servidor de base de datos encargado de almacenar los datos de los usuarios y los dispositivos domóticos; (3) un dispositivo móvil (teléfono celular) el cual será la herramienta que el usuario utilizará para tener control sobre las luces del hardware domótico; (4) un controlador domótico encargado de enviar las órdenes de apagado o encendido al hardware domótico y de estar monitoreando con el servidor de aplicaciones si hay tareas pendientes por realizar; (5) un hardware domótico que simula a escala el ambiente interno de una casa y su sistema de iluminación y (6) una estación cliente desde la cual el administrador puede configurar y administrar los valores del sistema como por ejemplo, agregar usuarios o dispositivos domóticos.

La tabla 2 incluye las características físicas de los elementos que integran el sistema y que se aplicarán para el desarrollo del presente proyecto. Las configuraciones pueden variar dependiendo del alcance que se quiera dar a la aplicación las cuales se describen en la columna de observaciones.

**Tabla 2. Elementos que conforman la arquitectura del sistema**

<b>Elemento</b>	<b>Características físicas</b>	<b>Dirección IP</b>	<b>Observaciones</b>
Servidor de aplicaciones	Pentium IV 1Gh, 256Mb de RAM, disco duro de 20Gb, tarjeta de red de 100Mbps	Fija - Pública	Para un alcance mayor se debe tener un servidor con procesador Intel/AMD 2Ghz, 2Gb de RAM, disco duro de 100Gb, tarjeta de 1000Mbps
Servidor de Bases de Datos	Pentium IV 1Gh, 256Mb de RAM, disco duro de 40Gb, tarjeta de red de 100Mbps	Fija-Privada	Recomendable un disco duro de 80Gb, memoria de 1Gb, tarjeta de 1000Mbps
Dispositivo móvil	Celular con tecnología GSM, memoria 3Mb, compatible con J2ME MIDP2.0	N/A	Para el presente proyecto se simulará en Mobility Pack.
Estación cliente	Cualquiera con conexión a Internet	Cualquiera	
Controlador domótico	Intel/AMD 1Gh, 256Mb de RAM, disco duro de 20Gb, tarjeta de red de 100Mbps	Fija-Privada	Algunos proveedores de Internet dan a cada usuario una IP dinámica.

## 5.2 ARQUITECTURA DE SOFTWARE

Luego de la definición de la estructura y características de hardware, a continuación se definirá la arquitectura de software para la parte Web y la parte midlet. Así mismo, se dará un vistazo a las opciones de software de desarrollo que se pueden utilizar para realizar la aplicación en ambas partes y se escogerá la más apropiada.

**5.2.1 Página Web.** La figura 23 muestra la estructura lógica de la página Web. A través de esta página se podrá configurar todo el sistema.

En la primera parte se ve el menú principal que muestra tres opciones: la primera es la *administración de locaciones* en donde se crean, editan, eliminan o consultan las locaciones a las que pertenecen los dispositivos de iluminación; la segunda opción es la de *administrar dispositivos* en la cual se crean, editan, eliminan o consultan los dispositivos domóticos conectados al sistema, en el caso del

presente proyecto, las luces de la casa a escala; len esta opción el administrador da permisos a los grupos de usuarios sobre dichos dispositivos; la tercera opción *administrar usuarios*, permite crear, editar, eliminar o consultar los usuarios que accederán a través del midlet; como se mencionó anteriormente, cada usuario debe pertenecer a un grupo de usuario, el cual es creado en la misma opción.

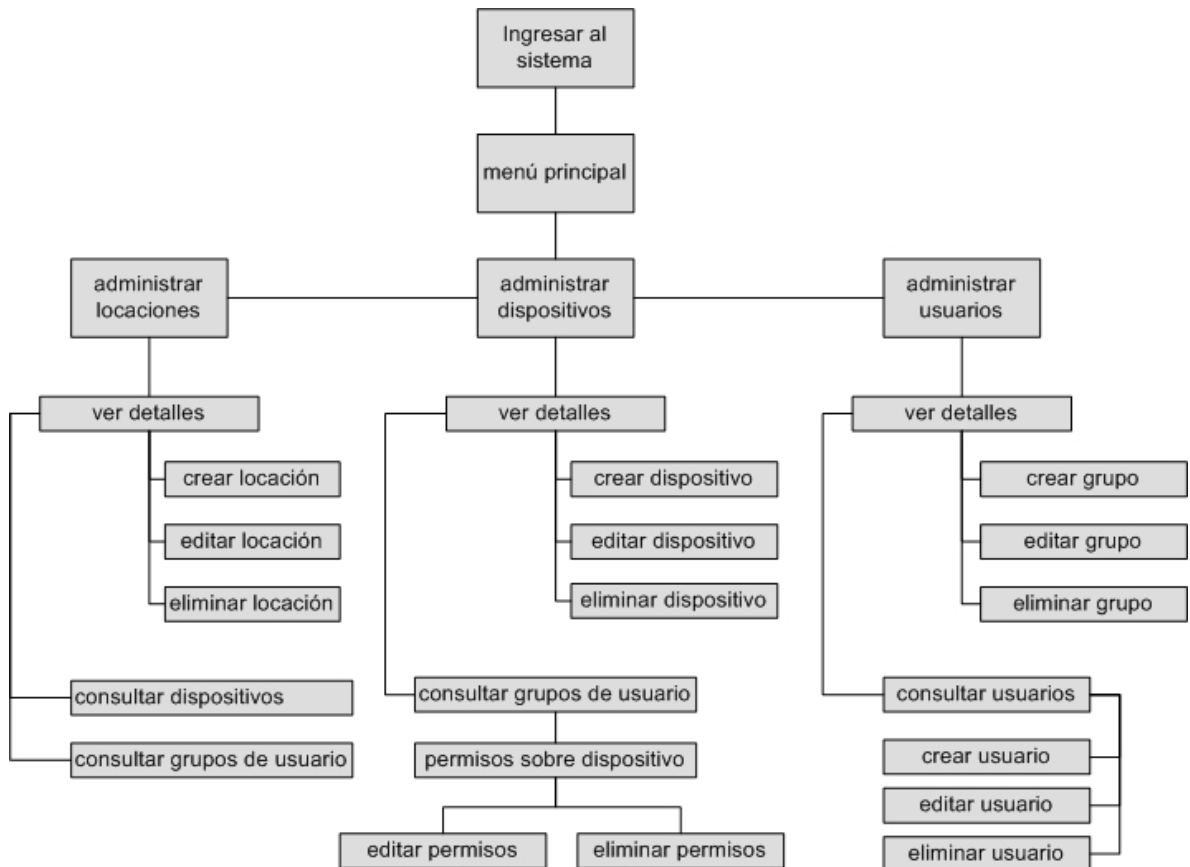
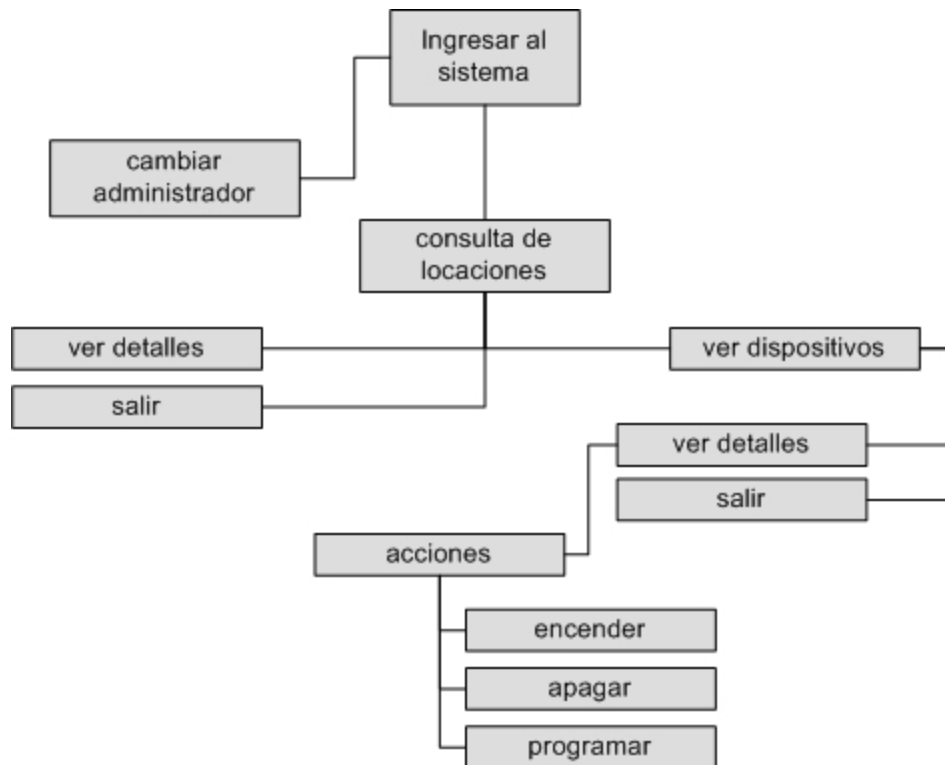


Figura 23. Estructura lógica de la página Web

Por ejemplo, el papá de la casa llamado “X” tiene el rol de administrador del sistema; crea un grupo llamado “hijos” al cual pertenecen su hija “Xa” y su hijo “Xb”. El papa “X” solo quiere que sus hijos controlen las luces de sus habitaciones por lo que otorga permisos al grupo “hijos” de controlar únicamente estas luces.

**5.2.2 Midlet.** Una vez configurado el sistema, se puede tener acceso al hardware domótico desde el teléfono celular. La figura 24 muestra la estructura lógica de la aplicación midlet.



**Figura 24. Estructura lógica del midlet**

Antes que un usuario pueda usar la aplicación midlet, el administrador debe validarse en el sistema. Los datos de ese administrador quedan guardados en el celular para que al momento en que el usuario entre, el servidor sepa a que administrador pertenece. Luego de ser validado con su contraseña, al usuario visualiza en la pantalla las locaciones a las cuales tiene acceso. El usuario selecciona una y a continuación ve los dispositivos de iluminación que pertenecen a esta. En esta parte, el usuario toma la decisión de enviar una orden, ya sea apagar, encender o programar el dispositivo de iluminación.

Para llevar a cabo todas estas actividades, el sistema consta de dos servlets encargados cada uno de controlar el sistema. Los diagramas de software que se muestran a continuación, dan un panorama de cómo estos servlets reciben una petición y la devuelven de nuevo al usuario, ya sea el administrador desde la aplicación Web o el usuario final desde el midlet.

### 5.2.3 DIAGRAMAS DE SOFTWARE

#### Servlet de la aplicación Web

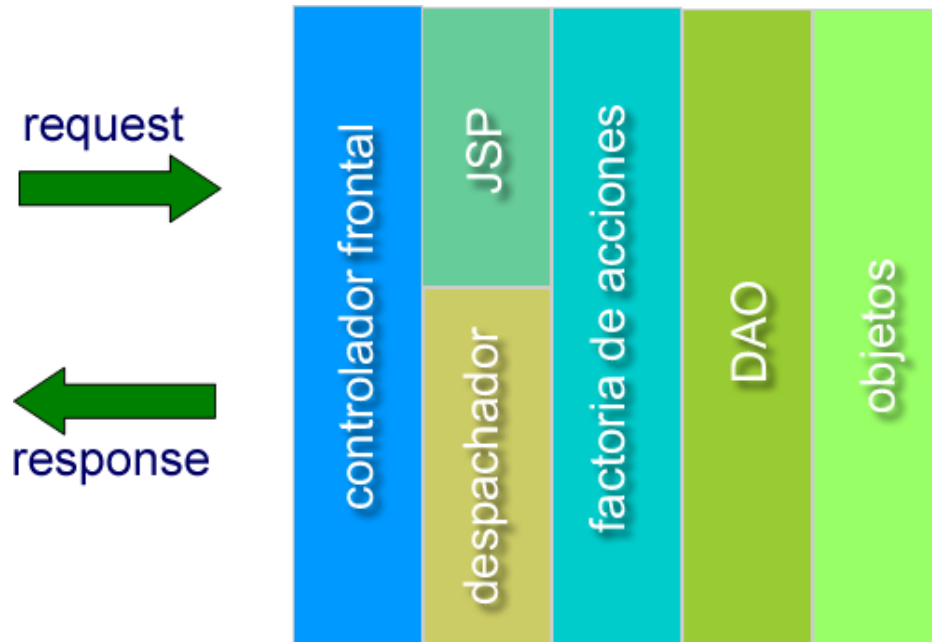


Figura 25. Diagrama de software del servlet de la aplicación Web

La figura 25 muestra el diagrama de software de la aplicación Web; las peticiones que envía el cliente (requests) son recibidas por el controlador frontal el cual, invoca al despachador correspondiente que designara la tarea. En la factoría de acciones se encapsulan los datos, que serán los parámetros que el Data Access Object (DAO) utilizara para actuar sobre la base de datos. Un objeto es devuelto hasta el despachador el cual lo retorna junto con una dirección de una pagina JSP al controlador frontal. El controlador devuelve la respuesta (response) al cliente.

#### Servlet de la aplicación midlet

El diagrama de software de la aplicación midlet utiliza la misma estructura de la aplicación Web. La única diferencia se muestra en la figura 26; en vez de retornar una pagina jsp al controlador frontal, retorna un XML.

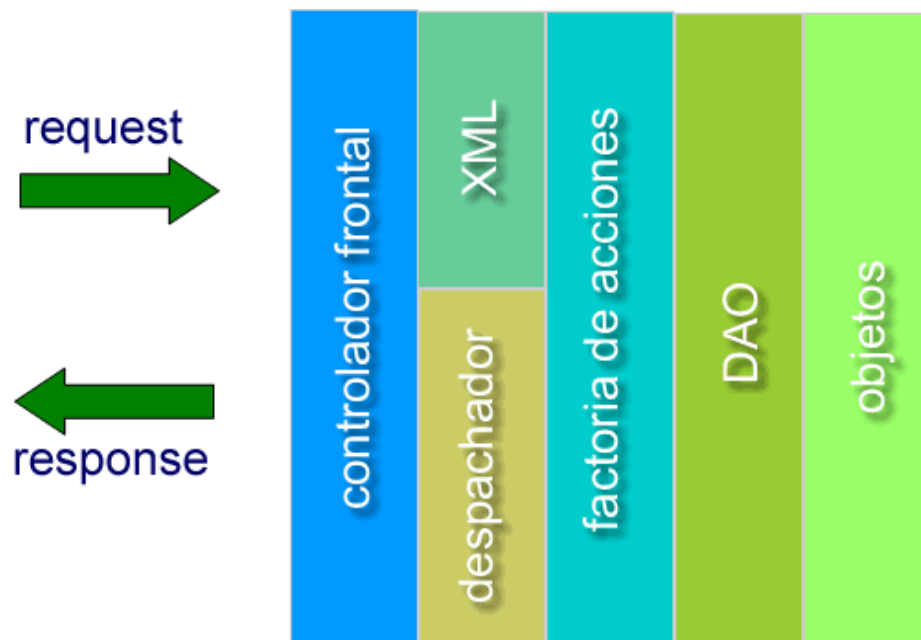


Figura 26. Diagrama de software del servlet de la aplicación Midlet

### 5.3 ALTERNATIVAS DE SOLUCION

A continuación se relacionan las alternativas de software que se pueden usar para el desarrollo de la aplicación Web y la aplicación midlet las cuales se analizan teniendo en cuenta los requerimientos funcionales y no funcionales descritos en la sección 3.1 y 3.2. La tabla 37 compara las opciones de dos programas de desarrollo escogidos: Java y .Net

Tabla 3. Comparación de los programas de desarrollo que se pueden utilizar para el desarrollo de la aplicación

Tecnología	Java	.Net
Aplicación		
Desarrollo aplicación dispositivos móviles	SI	SI
Manejo de bases de datos	SI	SI
Software libre	SI	NO
Aplicación cliente-servidor	SI	SI
Desarrollo de aplicación página Web	SI	SI
Compatibilidad entre	SI	SI

aplicación Web y aplicación móvil		
Orientado a objetos	SI	SI
Conocimiento del lenguaje	70%	20%

Java y .Net dentro de su paquete de desarrollo ofrecen la alternativa de desarrollar aplicaciones para dispositivos móviles. Sin embargo, el celular debe soportar archivos .JAD los cuales son únicamente generados por Java (J2EE). Así mismo, la aplicación Web debe ser diseñada en páginas JSP por lo que no se considerará desarrollarlas en ASPX, tecnología de .Net. Otra característica fundamental, es el costo de la licencia de Java comparada con la de .Net; mientras la primer herramienta es gratis, la segunda tiene costo.

Por otra parte, Java es compatible con MySQL, el manejador de bases de datos que se puede descargar gratis de Internet y las aplicaciones pueden ser alojadas en servidores Web configurados con apache-tomcat, el software que permite alojar paginas y aplicaciones Web.

Por último, el conocimiento del lenguaje Java es mayor en porcentaje a .Net; esto ahorra en recursos y tiempo el desarrollo del proyecto.

Por las razones anteriormente expuestas, el presente proyecto se desarrollará en Java, con la herramienta J2ME para la aplicación Midlet y J2EE para la aplicación Web alojada en un servidor Apache-Tomcat y con MySQL como manejador de base de datos.



## 6. DISEÑO DETALLADO

### 6.1. DIAGRAMA DE CLASES SIN PATRONES

La Figura 27 muestra el diagrama de clases sin patrones de la aplicación, en donde una clase llamada *Midlet* es la encargada de enviar órdenes al dispositivo domótico y otra llamada *Página Web*, envía los datos para configurar la aplicación. La “puerta” por donde entran y salen las órdenes y los datos es el *Controlador Principal* quien decide si utiliza el *Manejador Usuario*, el *Manejador Dispositivo* o el *Controlador Domótico* para llevar a cabo las tareas asignadas.

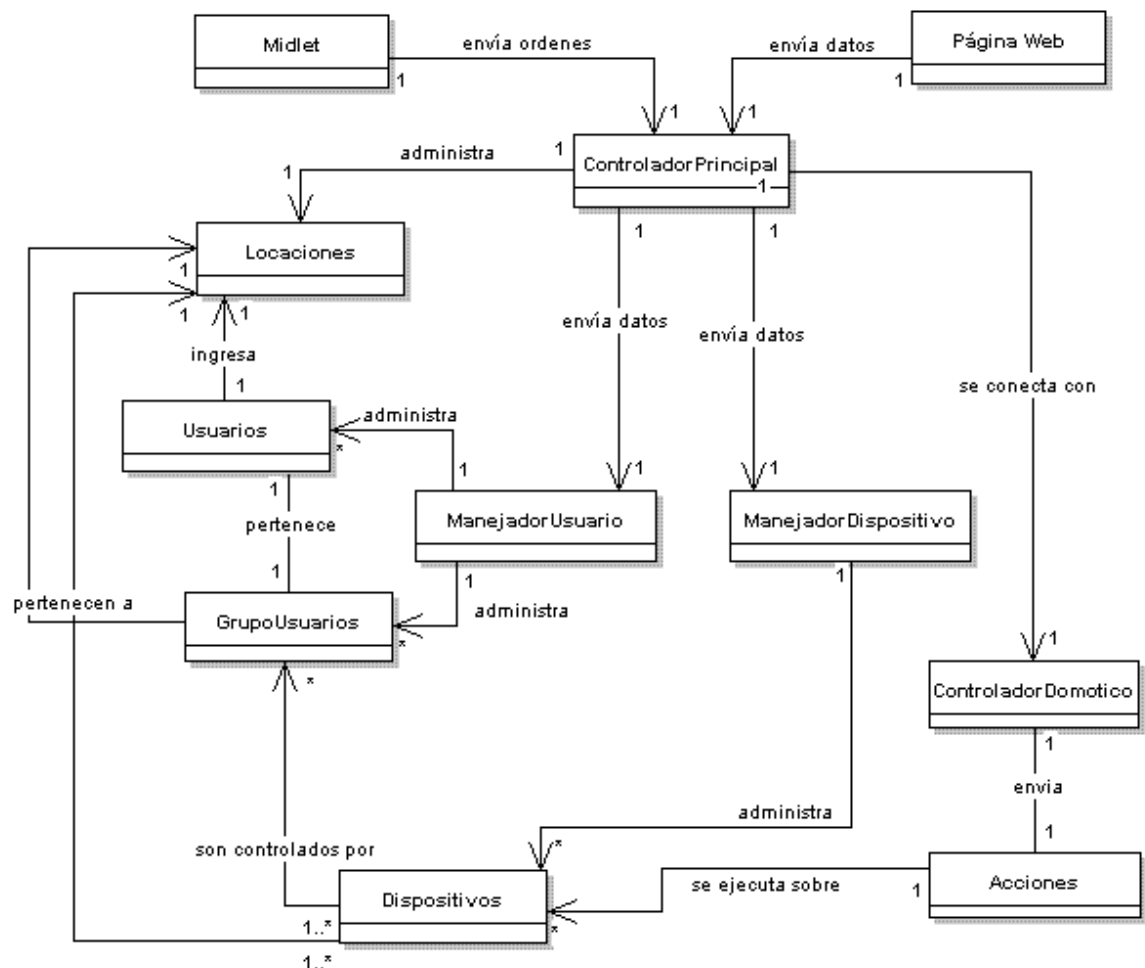


Figura 27. Diagrama de clases sin patrones

## 6.2 DIAGRAMA DE CLASES CON PATRONES

Los patrones son soluciones que se han dado a diferentes problemas encontrados al momento de desarrollar e implementar una aplicación en Java. En la página Web <http://java.sun.com/blueprints/corej2eepatterns/Patterns/index.html>, se explican en detalle todos los patrones publicados hasta el momento para J2EE. Para el desarrollo del presente proyecto se utilizarán cuatro de ellos: Front Controller, View Helper, Composite View y Data Access Object

### Front Controller

La estrategia de este patrón es utilizar un controlador frontal al que lleguen todas las peticiones (requests) del cliente; la clase recibe dichas peticiones y las encamina a un *despachador* el cual está encargado de reenviarlas a la vista correspondiente y de recibir de parte de esta los datos para retornarlos al controlador como una página JSP o un XML. La figura 28 muestra la estructura de este patrón.

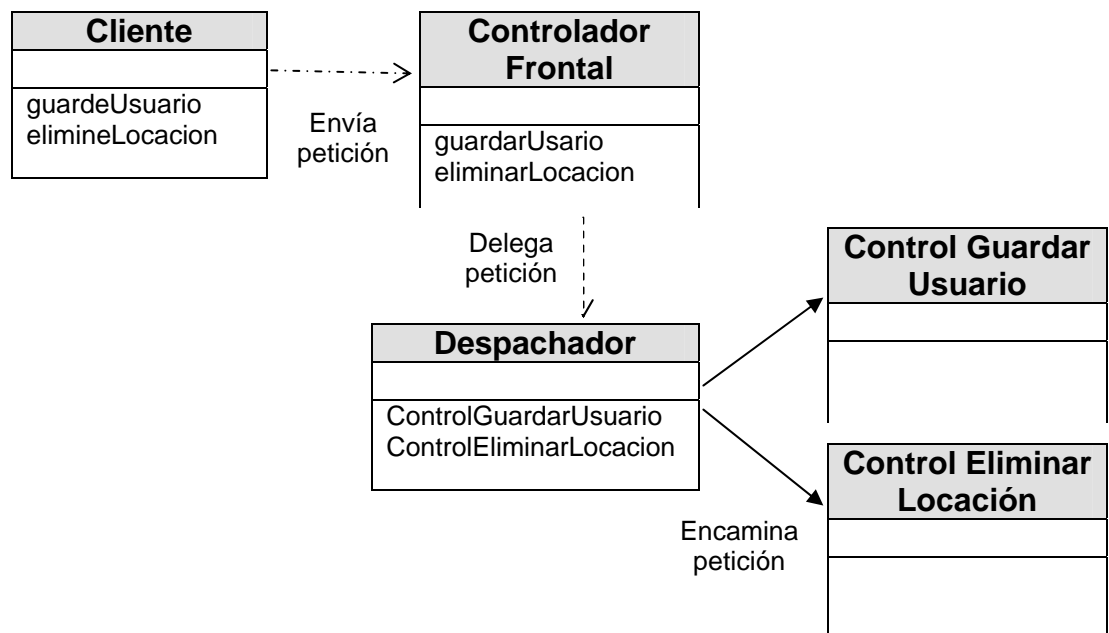


Figura 28. Patrón Front Controller

## View Helper y Composite View

La estrategia que utilizan estos dos patrones es el invocar los diferentes beans en la página JSP para el caso de la aplicación Web y XML para el midlet, para que puedan ser formateados y mostrados al usuario. Combinan etiquetas html con jsp para generar tablas o listas dinámicas. La figura 29 muestra la estructura de este patrón.

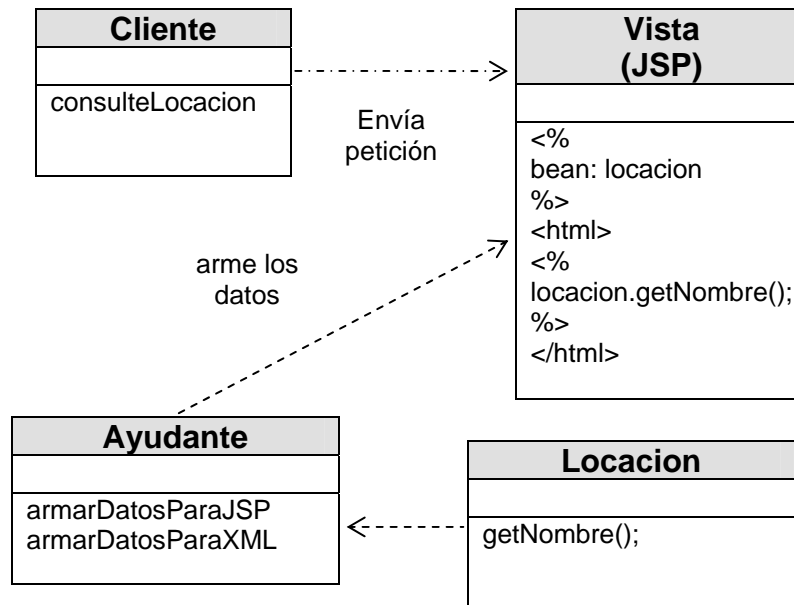


Figura 29. Patrón View Helper

## Data Access Object

Este patrón encapsula y abstrae los datos de la base de datos. Su estructura consiste en separar las instrucciones SQL del resto de componentes del código creando una factoría que se encargue de invocar los métodos que llevan a cabo esta tarea. La figura 30 muestra la estructura de este patrón.

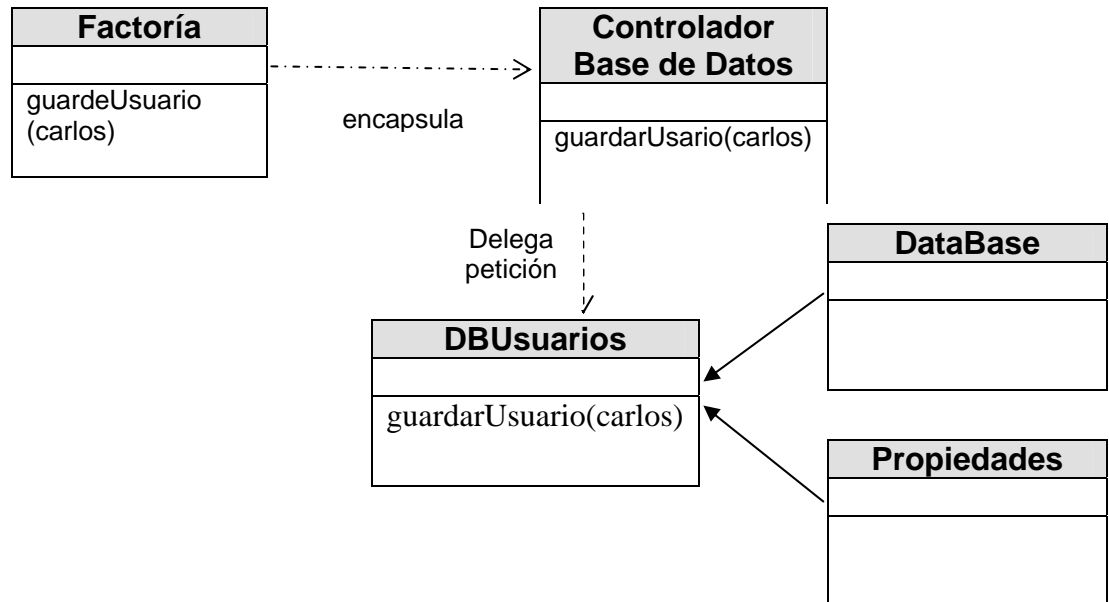


Figura 30. Patrón Data Access Object

Hecha la definición de los patrones que se implementarán en la aplicación, a continuación se describen los diagramas de clases. En primer lugar, se mostrará la estructura de los paquetes del servlet de la aplicación Web seguido por los diagramas de clase de las partes que intervienen en el sistema como son el usuario, el administrador, las locaciones, los grupos de usuario y los dispositivos de iluminación. Posteriormente, se mostrará la estructura de paquetes del servlet de la aplicación midlet junto con los diagramas de clase para las locaciones, los usuarios y los dispositivos.

La figura 31 muestra la estructura de los paquetes del servlet de la aplicación Web. El paquete controladorFrontal, guarda la clase que controla las peticiones desde el cliente junto con la interfaz JSPControlador, la cual es referenciada por los despachadores; estos se encuentran dentro de cada uno de los paquetes controlador. Los paquetes acción, contienen las clases factoría las cuales invocan cada uno de los DAO alojados en el paquete accionDB. Los demás paquetes contienen las clases donde son creados los objetos.

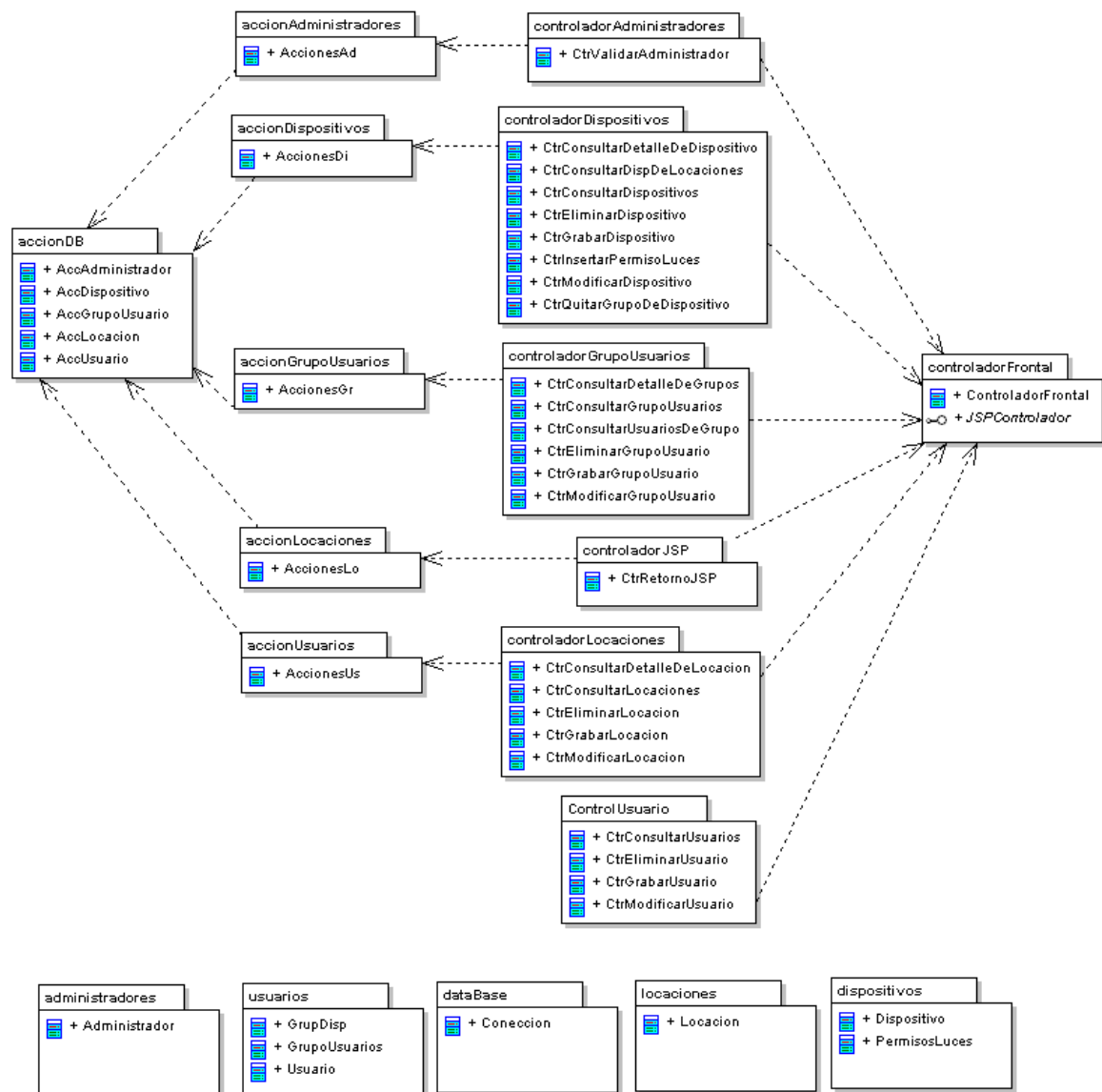


Figura 31. Estructura de los paquetes usados en la aplicación Web

La figura 32 muestra el diagrama de clases de la aplicación Web para la parte de *usuario*. Existen cuatro despachadores los cuales reciben los parámetros para que puedan ser enviados a la factoría. Estos mismos despachadores, retornan al controlador frontal la dirección de una página en JSP junto con los objetos necesarios para generar la vista correspondiente.

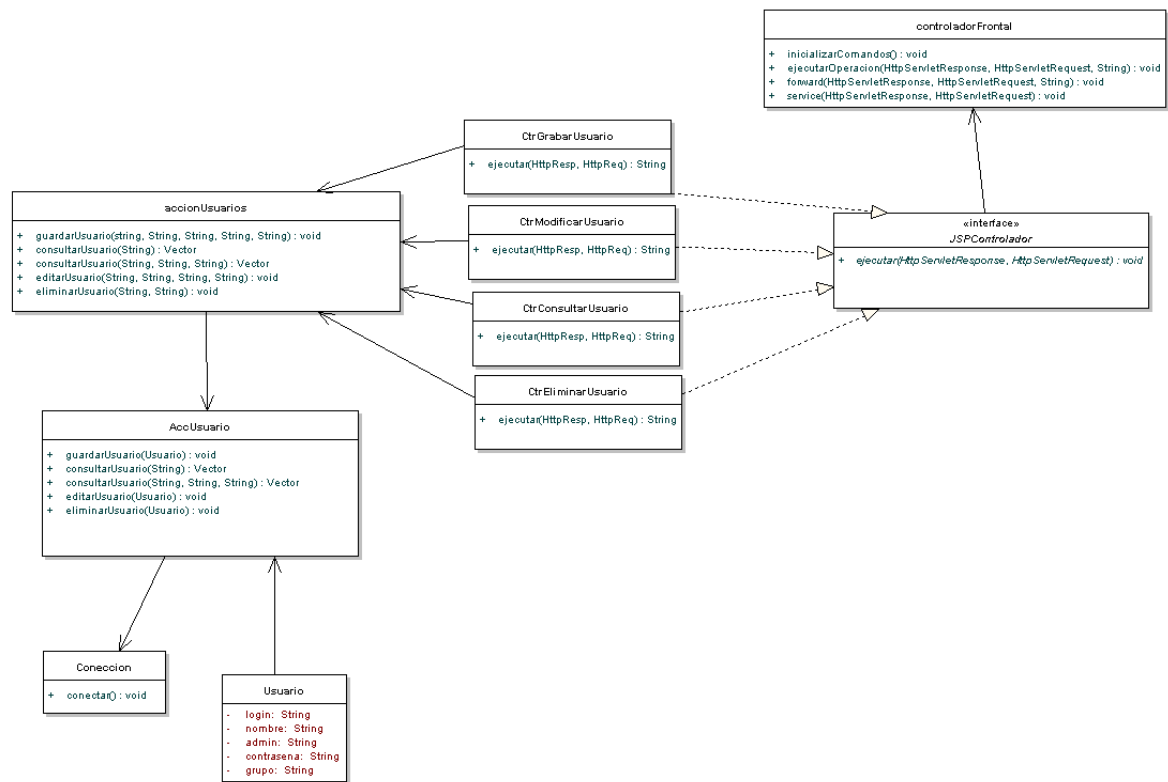


Figura 32. Diagrama de clases del usuario

La figura 33 muestra el diagrama de clases de la aplicación Web para la parte del *dispositivo*. En este diagrama se observa mayor cantidad de despachadores que en las clases del usuario debido a que se necesita realizar mas acciones sobre los dispositivos de iluminación. Se observa además que, a diferencia del usuario, los métodos de la factoría superan en número a los despachadores, esto es porque un mismo despachador puede invocar no solo uno sino varios métodos y retornar al controlador frontal varios objetos.

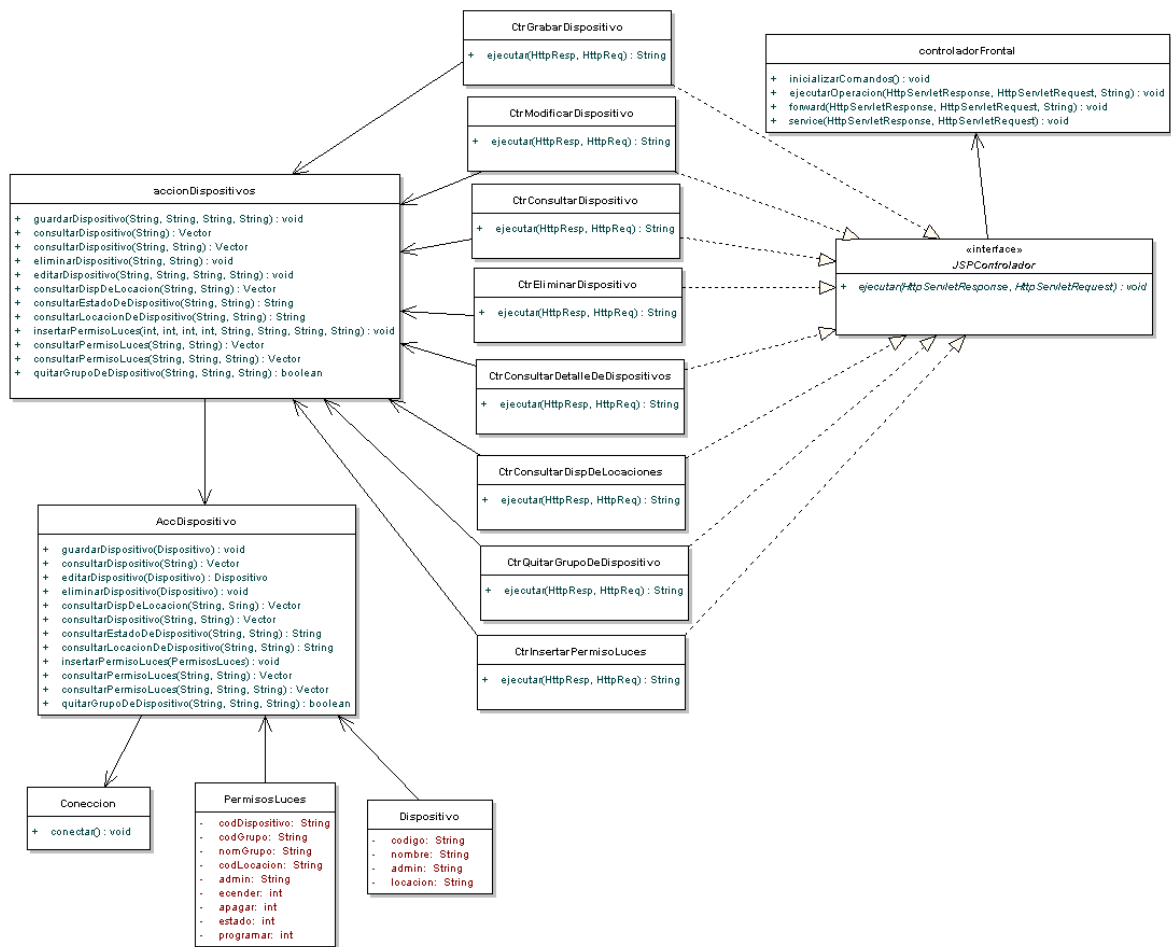


Figura 33. Diagrama de clases del dispositivo

Al igual que los diagramas anteriores, para la parte de la *locación* se emplea la misma estructura. La figura 34 muestra cinco despachadores que invocan sus respectivos métodos de la factoría. Cabe resaltar que cada método creado en la factoría debe tener su respectiva “pareja” en el DAO. En algunas de las acciones que se llevan a cabo, la factoría encapsula los datos en un objeto y son devueltos en un vector al despachador.

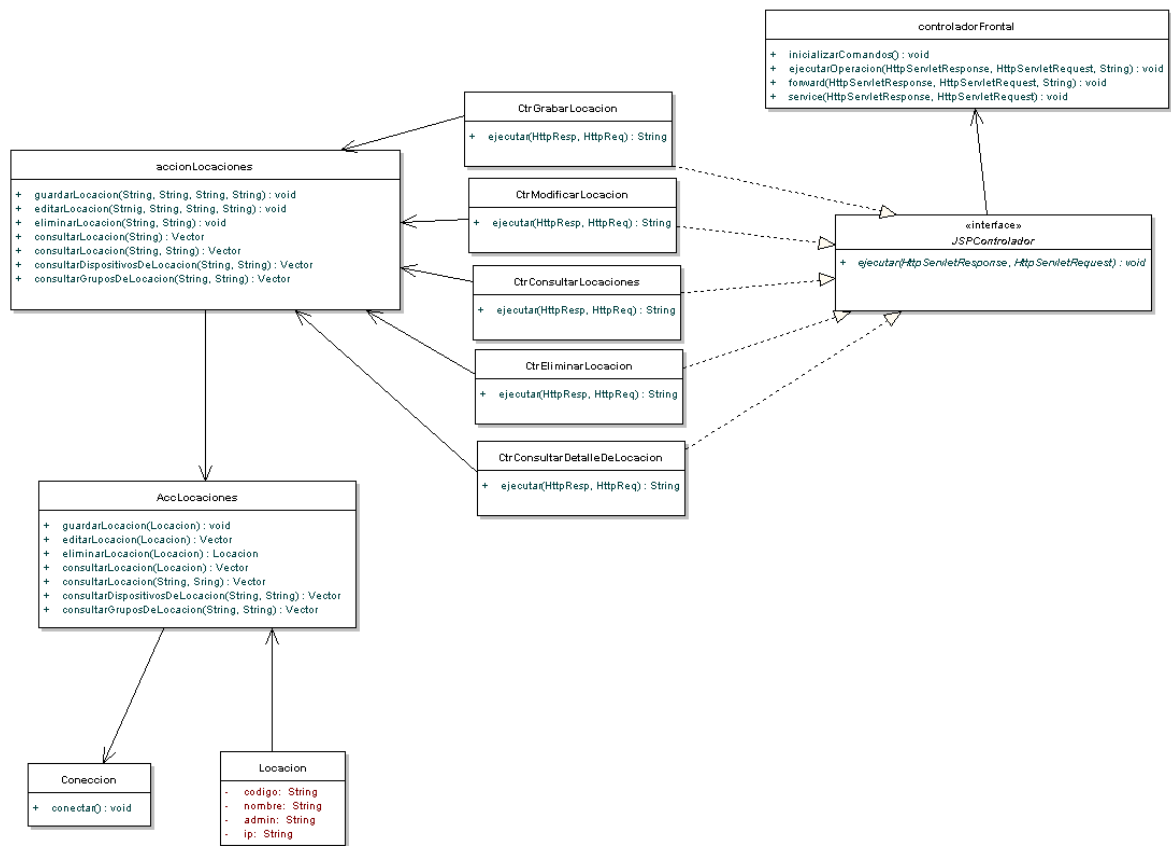


Figura 34. Diagrama de clases de la locación

Por último, en la figura 35 se muestra el diagrama de clases para los grupos de *usuario*. En este diagrama se observa que intervienen tres objetos: *grupoUsuarios*, *Usuario* y *GrupDisp*; cada uno de ellos es referenciado dependiendo del tipo de acción que se lleve a cabo. Estos mismos objetos serán invocados por la página JSP para generar las vistas dinámicas al momento de consultar un grupo de usuario o un dispositivo.



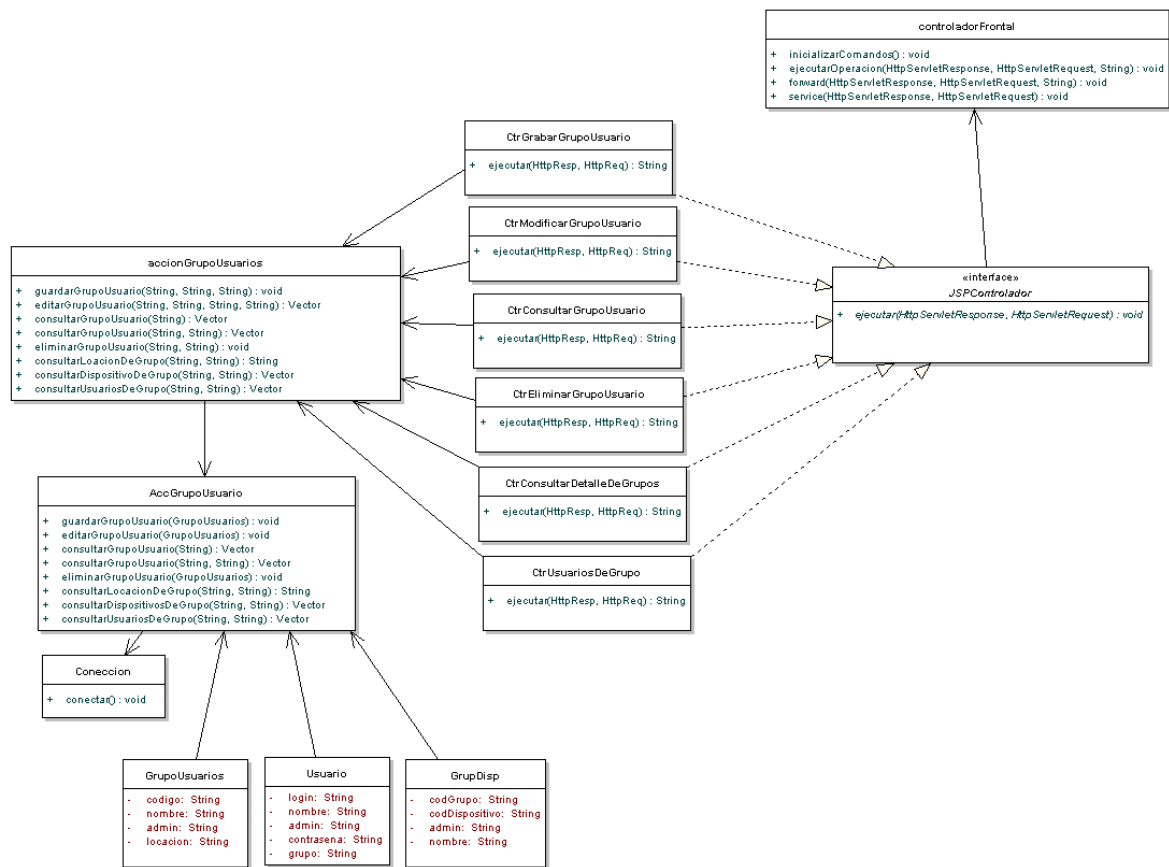


Figura 35. Diagrama de clases de grupo de usuarios

Los diagramas de clases que se presentan a continuación, muestran la estructura del servlet de la aplicación midlet. En la figura 36 se ilustra los paquetes que conforman este servlet. Un primer paquete llamado ControladorFrontal recibe las peticiones desde el midlet y las encamina al respectivo despachador. Para este servlet solo habrá tres despachadores: el controlador de dispositivos, el de usuarios y el de locaciones. El servlet del midlet sigue la misma lógica del servlet de la aplicación Web, es decir, una factoría que encapsula los datos que van a acceder a la base de datos. La diferencia radica en el retorno del despachador al controlador frontal pues en vez de retornar una dirección de una página JSP, devuelve un XML.

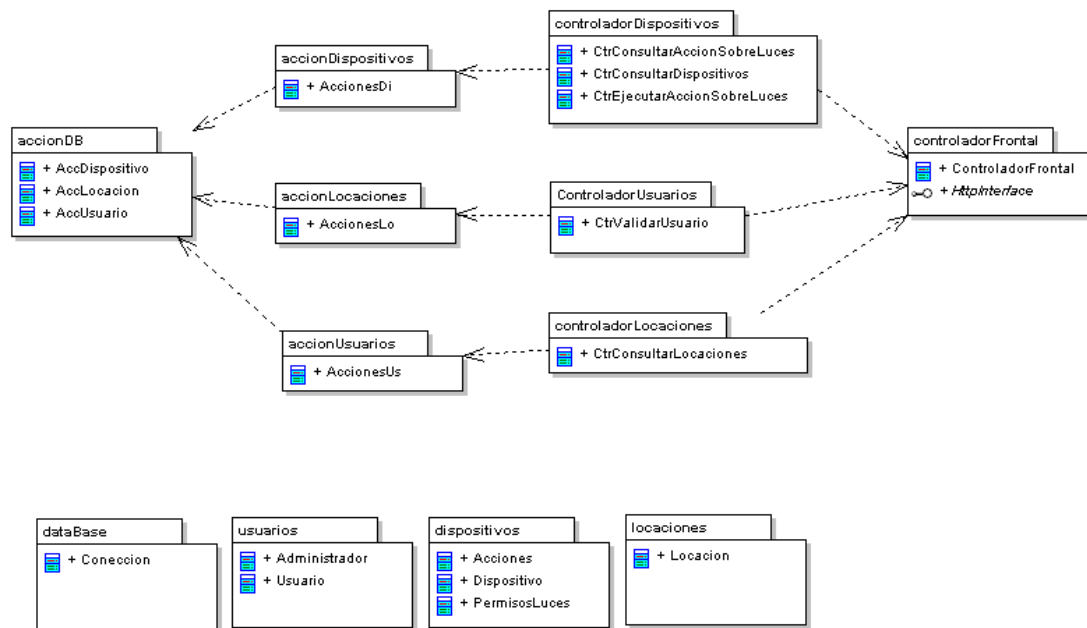


Figura 36. Diagrama de clases servlet controlador domótico

La figura 37 muestra el diagrama de clases de la parte de *dispositivos*. Consta de tres despachadores los cuales son los encargados de enviar a la fachada las órdenes que el usuario envía a través del midlet como apagar o encender el dispositivo de iluminación. La fachada *accionDispositivos* encapsula estas instrucciones y las envía para que sean guardadas por las clases del DAO (paquete *AccDispositivo*). Posteriormente, una aplicación tomará estos datos y los enviará al controlador domótico para que efectúe la acción sobre el hardware domótico.

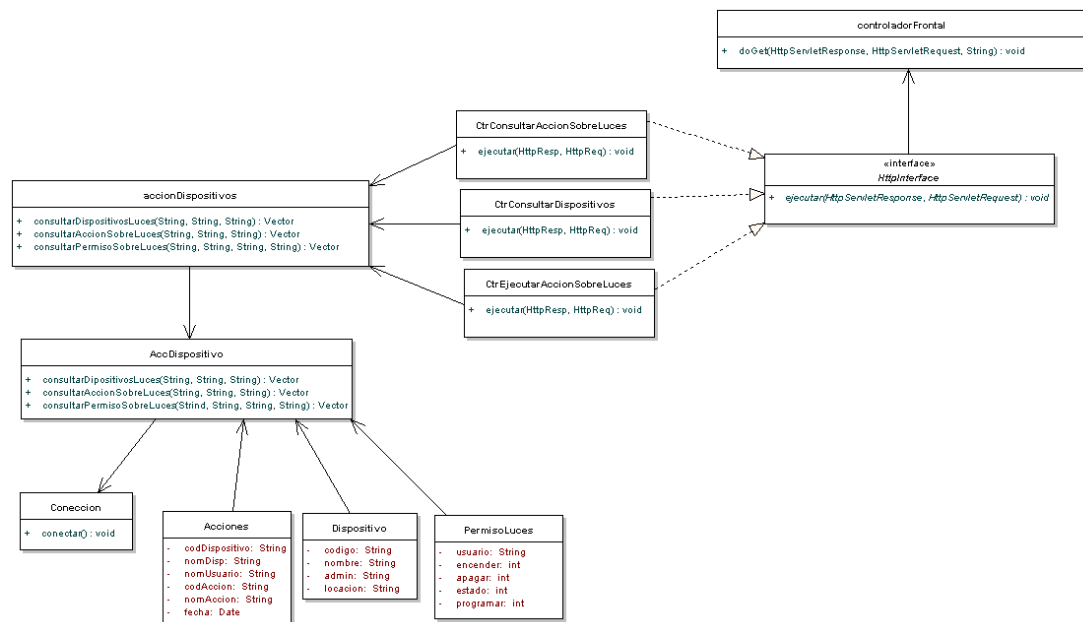
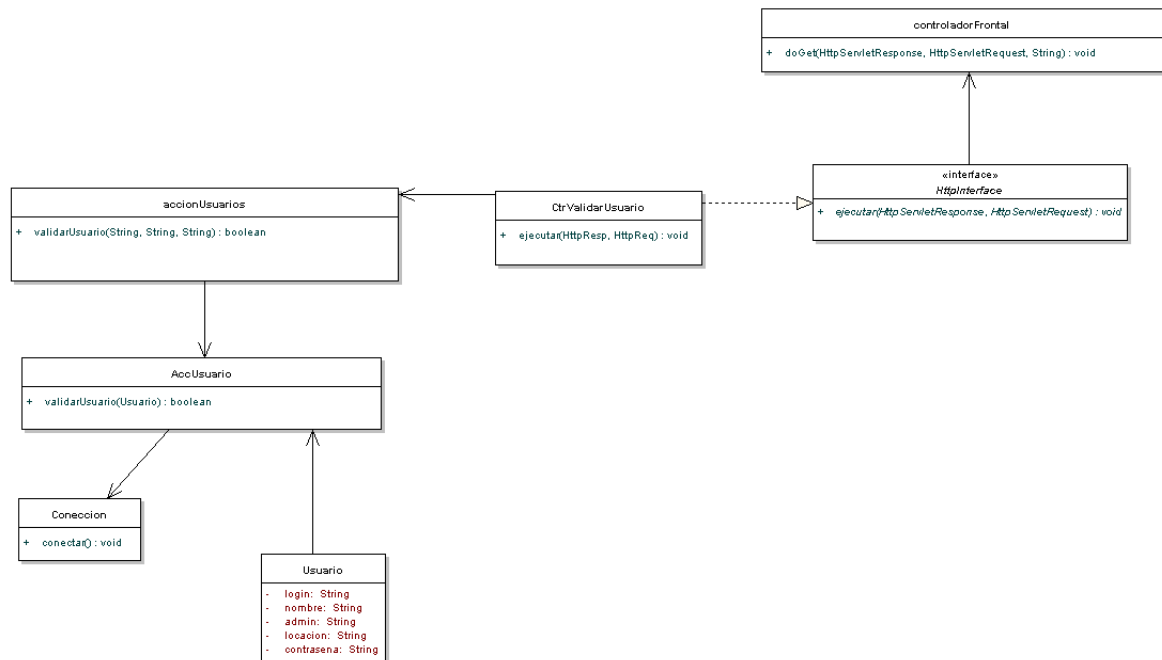


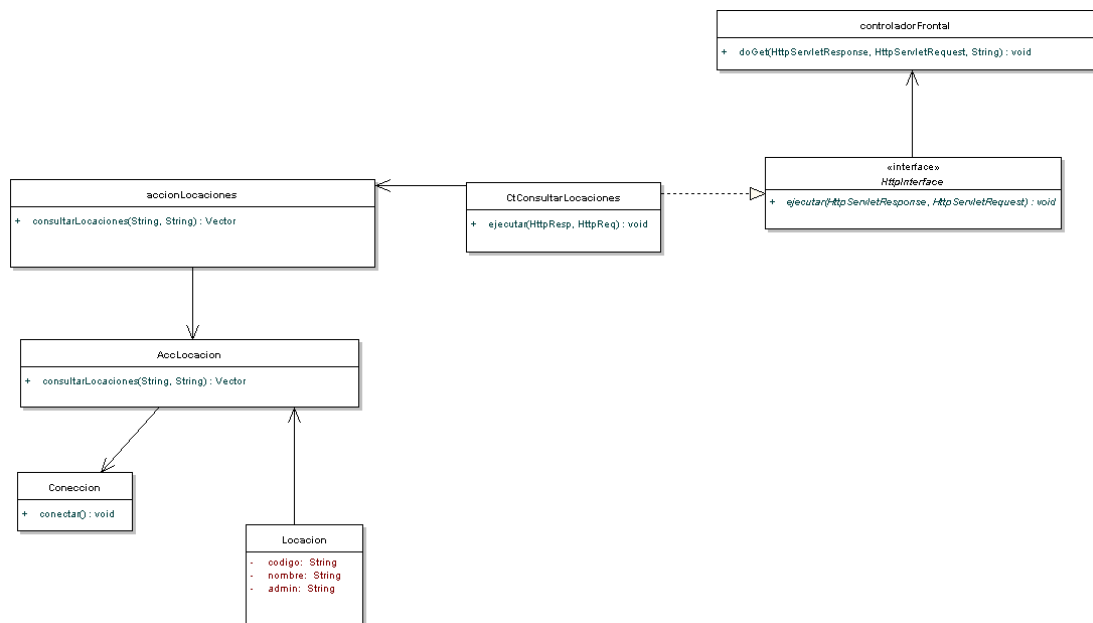
Figura 37. Diagrama de clases Dispositivos Servlet controlador Domótico

El diagrama que se muestra en la figura 38 pertenece al usuario; de observa que solo interviene un despachador debido a que la única función de este es validar al usuario y devolver un XML con la respuesta afirmativa o negativa.



**Figura 38. Diagrama de clases Usuarios Servlet controlador Domótico**

Por último se muestra en la figura 39 el diagrama de clases de la aplicación midlet para la parte de locaciones. Al igual que el diagrama de usuario, este tiene un solo despachador encargado de enviar a la factoría un parámetro para que consulte las locaciones a las cuales el usuario que hizo la petición tiene acceso. La lista es retornada en un XML al controlador frontal.



**Figura 39. Diagrama de clases Locaciones Servlet controlador Domótico**

## 7. IMPLEMENTACION

Luego de la etapa de diseño, se procedió con la implementación del sistema, la cual se dividió en cuatro partes. La primera consistió en desarrollar el Servlet de la aplicación Web, por medio de la cual, el administrador configura el sistema. En la segunda parte se desarrolló el Servlet del Midlet, al cual llegan las órdenes y peticiones desde el celular. La tercera parte consistió en desarrollar el código del Midlet, desde donde el usuario interactúa con el hardware domótico. La ultima parte, consistió en desarrollar la parte RMI en donde el controlador domótico monitorea periódicamente si hay instrucciones a ejecutar.

Para integrar estas cuatro partes en un solo sistema de control domótico se siguieron los siguientes pasos:

En primer lugar se bajaron los programas de desarrollo NetBeans 5.5 para desarrollar los dos Servlets y DreamWeaver MX para las paginas JSP, este ultimo con una versión de prueba de 30 días. El midlet fue creado utilizando el componente MobilityPack de NetBeans, especial para desarrollos de esta naturaleza.

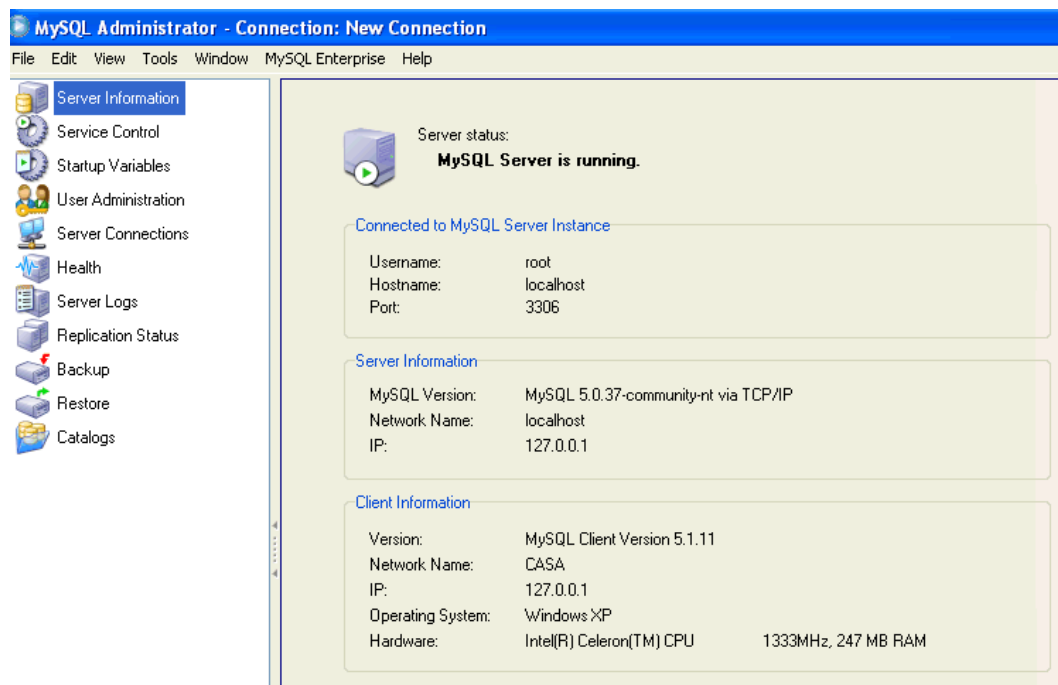


Figura 40. Interfaz de MySQL Administrator

Posteriormente se creo la base de datos del sistema. Para esto se descargo de Internet MySQL 5.0 junto con MySQL Administrador (figura 41), interfaz que permite crear de manera grafica bases de datos y administrarlas. A la base de datos se le llamó “domótica” en donde fueron creadas las diferentes tablas las cuales se explican en detalle en el diagrama entidad relación en el siguiente capitulo.

Las páginas JSP y las clases de los Servlets fueron alojadas el servidor de Internet y aplicaciones apache-tomcat 5.0; para que por medio de una dirección URL tomcat pueda encontrar el Servlet correspondiente, es necesario crear un archivo XML llamado *web.xml*; al momento de recibir una petición, tomcat busca las clases en la ruta que se especifique en este archivo. La figura 42 muestra el código XML que se debe generar para que tomcat encuentre el Servlet correspondiente.

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.4"
  xmlns="http://java.sun.com/xml/ns/j2ee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
    http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">

  <servlet>
    <servlet-name>ControladorFrontal</servlet-name>
    <servlet-class>controladorFrontal.ControladorFrontal</servlet-class>
  </servlet>

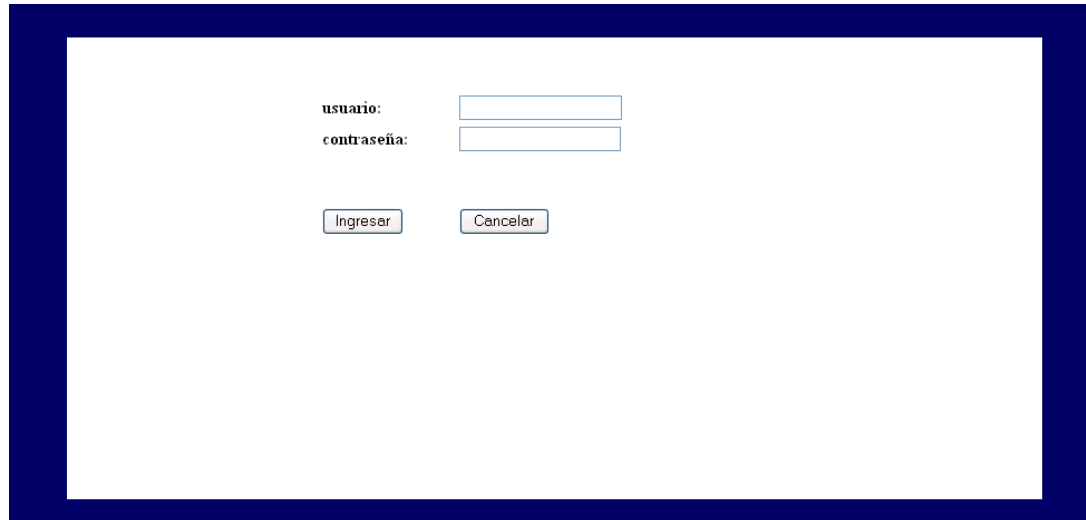
  <servlet-mapping>
    <servlet-name>ControladorFrontal</servlet-name>
    <url-pattern>/ControladorFrontal</url-pattern>
  </servlet-mapping>

</web-app>
```

**Figura 41. Archivo web.xml**

En este caso, cuando en una dirección URL se ingrese /ControladorFrontal, tomcat buscará el Servlet en una clase llamada ControladorFrontal.class dentro de un paquete llamado controladorFrontal en la carpeta WEB\_INF/classes, en la cual se guardan las clases por defecto.

Luego de los pasos descritos anteriormente, el administrador puede comenzar a interactuar con el sistema. Para ingresar, desde un navegador se ingresa la dirección <http://localhost:8080/domotica/JSP/login.jsp>



usuario:

contraseña:

Figura 42. Página de inicio.

La figura 43 muestra la página inicial en donde se digita la cuenta del administrador y la contraseña. Luego, si es validado correctamente, se muestra una página (figura 44) con tres opciones: *Administración de Dispositivos*, donde el administrador puede crear, eliminar, editar y asignar permisos a los dispositivos de iluminación, *Administración de Usuarios*, donde el administrador crea los usuarios y los grupos de usuario y *Administración de Locaciones*, donde el administrador crea, edita y elimina las locaciones a donde pertenecerán los dispositivos de iluminación.



Figura 43. Administración del sistema.

Cuando el administrador selecciona los detalles de un determinado dispositivo, la página muestra los grupos de usuario que tienen acceso a el, y los permisos que estos tienen sobre el dispositivo de iluminación. Se han creado cuatro permisos: encender, apagar, ver estado y programar, los cuales son diferenciados por colores. Por ejemplo, en la figura 45, el dispositivo luz cuarto de abuelita Fabiola tiene tres grupos con permisos entre los que está el grupo hijos e hijas; este grupo tiene todos los permisos menos el de apagar la luz (color negro).



Figura 44. Detalle de dispositivos.

El usuario por su parte, ingresa al sistema a través del midlet; sin embargo, es necesario que previamente el administrador haya registrado su cuenta en la aplicación. Debido a que un administrador puede tener a su cargo varios usuarios, al momento de enviar un dato al Servlet, este identificará qué usuario está realizando la petición y a qué administrador pertenece.

Todas las respuestas desde el Servlet retornan al midlet en forma de XML; las etiquetas son leídas e interpretadas por una librería de java llamada kxml2 la cual se descarga de [www.kxml.org](http://www.kxml.org).

En cualquier momento en el midlet, se puede cambiar de sesión de administrador; esto se logra utilizando RMS, la base de datos propia de las aplicaciones para dispositivos celulares, la cual, almacena los datos del administrador actual.

La figura 46 (a) muestra las locaciones a las cuales un usuario tiene acceso. Al seleccionar una de ellas, el midlet envía al Servlet una petición de buscar los dispositivos que pertenecen a dicha locación y las cuales, el usuario tiene permiso



de ver. El Servlet retorna un XML que es interpretado y mostrado en la pantalla del celular como se muestra en la figura 46 (b).



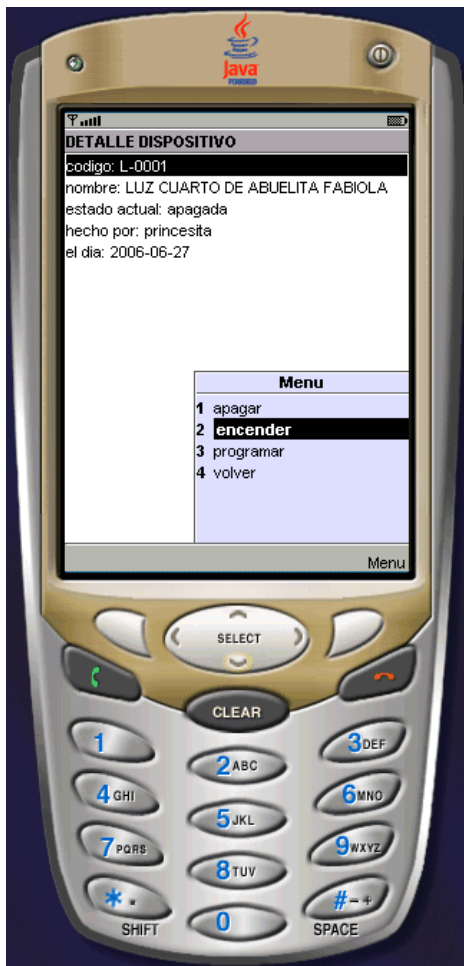
(a)



(b)

**Figura 45. Midlet domótico.**

Cuando el usuario selecciona ver detalles, se muestra el estado actual del dispositivo de iluminación, el último usuario que realizó una acción sobre el y la fecha en que se hizo (figura 47 a). Dependiendo de los permisos que el administrador haya dado al grupo al que pertenece el usuario, se puede apagar, encender o programar el dispositivo de iluminación. De tener permiso para ejecutar la acción, el usuario recibirá un mensaje de confirmación, de lo contrario recibirá un mensaje de negación de la acción (figura 47 b).



(a)



(b)

**Figura 46. Envío de orden de encender a través del midlet**

Para la ejecución de la orden enviada desde el midlet, se conectó el hardware domótico a un computador por medio del puerto serial. Una instrucción en java envía un código ASCII, el cual es recibido por el hardware al que previamente se ha programado con el software JDSTechnology. Las instrucciones son enviadas y leídas hacia y desde el puerto serial utilizando la librería javacomm, la cual se descarga gratis de Internet.

## 8. PRUEBAS

Durante la etapa de desarrollo e implementación, se realizaron diferentes pruebas al sistema para analizar qué errores se presentaban y cómo corregirlos. Estas pruebas fueron hechas sobre el hardware domótico, el midlet, la aplicación Web y los dos Servlets. Los resultados de las pruebas más importantes y que influyeron en cambios puntuales en el sistema se muestran a continuación.

### 8.1 PRUEBAS UNITARIAS

Tabla 4. Pruebas Unitarias

<b>Id Prueba</b>	<b>Descripción</b>	<b>Datos de entrada</b>	<b>Datos de salida</b>	<b>Resultado de prueba</b>
P-0001	Validación de administrador en la página Web	Login de administrador y contraseña	Página principal	Satisfactorio
P-0002	Verificación del controlador frontal	http://localhost:8080/domotica/ControladorFrontal?accion=nombre del Control & Parametros	Retorno de las consultas en JSP	Satisfactorio
P-0003	Insertar nuevo usuario	Login, contraseña, nombre del usuario, código del administrador, código de locación	Confirmación de éxito de la ejecución	Satisfactorio
P-0004	Editar, eliminar usuario	Login de usuario, código de locación, código de administrador	Confirmación de éxito de la ejecución	Satisfactorio
P-0005	Ver los grupos a los que pertenece el usuario	Login de usuario, código de locación, código de administrador	Código y nombre de los grupos de usuario a los que pertenece el usuario	Satisfactorio
P-0006	Ver los dispositivos que tiene asignados el usuario	Login de usuario, código de locación, código de administrador	Código y nombre de los dispositivos a los que pertenece el	Satisfactorio

P-0007	Insertar nuevo dispositivo	Código de dispositivo, código administrador, código locación	usuario Confirmación de éxito de la ejecución	Satisfactorio
P-0008	Editar, eliminar dispositivo	Código de dispositivo, código administrador, código locación	Confirmación de éxito de la ejecución	Satisfactorio
P-0009	Ver usuarios que tienen permisos sobre ese dispositivo	Login de usuario, código dispositivo, código administrador, código locación	Código de usuario y nombre de usuario	Satisfactorio
P-0010	Ver grupos de dispositivos a los que pertenece el dispositivo	código dispositivo, código administrador, código locación	Código de grupo, nombre de grupo	Satisfactorio
P-0011	Consulta de locaciones	Código del administrador	Código y nombre de las locaciones	Satisfactorio
P-0012	Consulta de los usuarios que pertenecen a esa locación	Código de locación, código de administrador	Código y nombre de usuario	Satisfactorio
P-0013	Consulta de los dispositivos que pertenecen a esa locación	Código de locación, código de administrador	Código y nombre del dispositivo	Satisfactorio
P-0014	Validación de nombre de usuario en el midlet	Nombre de usuario y contraseña	Locaciones a las que pertenece el usuario	Satisfactorio, fue necesario emplear XML para traer los datos desde el Servlet al midlet.
P-0015	Ver lista de dispositivos domóticos	Código de usuario, código de locación, código de administrador	Lista de dispositivos a los que el usuario tiene acceso	Satisfactorio, fue necesario emplear XML para traer los datos desde el Servlet al midlet.

P-0016	Envío de datos al puerto serial	Bits de datos	Retorno de mensaje de comunicación satisfactoria	Se descargaron los archivos comm.jar, win32comm.dll, javax.comm.properties y funcionó correctamente
P-0017	Validación del usuario a través del Midlet	Login de usuario y contraseña	Locación a la cual pertenece el usuario.	Satisfactoria.
P-0018	Ver lista de dispositivos	Locación, usuario, administrador	Lista de dispositivos asignados al usuario.	Satisfactoria
P-0019	Enviar orden de encendido y apagado al hardware domótico.	Código de dispositivo, código de orden.	Mensaje de confirmación.	Se presentaron inconsistencias de datos; fue necesario hacer adecuaciones en el código del Servlet.
P-0020	Programar dispositivos.	Código de dispositivo, usuario, fecha, hora, acción.	Mensaje de confirmación.	Fue necesario crear un programa en el servidor para que monitoree cada minuto si hay acciones pendientes por realizar.
P-021	Monitoreo de acciones pendientes por parte del controlador domótico.	Código de locación.	Código a ejecutar sobre el hardware domótico.	Se presentaron errores de permisos de acceso a la base de datos y de puerto. Fue necesario dar permisos a la dirección IP pública del controlador domótico y tomar una red sin firewalls

para el puerto  
3306

## 8.2 PRUEBAS SOBRE EL HARDWARE DOMÓTICO

El hardware domótico recibe códigos ASCII enviados a través del puerto serial del computador. Para probarlo, se utilizó el programa JDSTechnology en el cual se puede programar el hardware para que determinado código realice una acción. El código que se programó fue el 5 el cual, hacía que el hardware emitiera un sonido. La prueba fue satisfactoria.

Se necesitaba diseñar entonces un pequeño programa en Java que enviara el 5 a través del puerto. Java tiene una librería llamada comm.jar la cual envía cualquier dato al puerto serial. La instrucción para el envío de un dato al puerto serial es la siguiente:

```
port=CommPortIdentifier.getPortIdentifier("COM1"); //abre el puerto COM1
CommPort p=port.open("Vlpport1",30000 );//30000 son los segundos
SerialPort sp=(SerialPort)p;
os = new DataOutputStream(p.getOutputStream()); //crea un flujo de datos
int i=5;
Number l=new Integer(i);
os.write(l.byteValue()); //envia el dato al puerto COM1
System.out.println(i);//muestra el dato en pantalla
os.flush();
```

Al momento de ejecutar el código, el hardware domótico no emitía sonido alguno.

Para saber cuál era el problema exacto, se diseñó un programa en java que recibiera los datos enviados a través del puerto. Esto se conectando dos computadores por medio de un cable serial. Cuando se ejecuto de nuevo el programa que enviaba el 5, el otro computador no recibía el dato. Se pensó entonces que el cable serial era el origen de la falla.

Para comprobar esa teoría, se bajó de Internet un emulador de cable null-modem, llamado virtual serial port; este software crea puertos COM virtuales en el computador así físicamente no exista. Se crearon los puertos COM3 y COM4 y se adaptó el software para que enviara el 5 por el puerto COMM3 y lo recibiera por el puerto COM4. La prueba fue satisfactoria.

Esta prueba confirmó que el programa que envía datos al puerto funcionaba correctamente, pero no resolvía aún el problema, pues no estaba ejerciendo ninguna acción sobre el hardware domótico.

Se pensó que el problema era el código enviado. En el programa JDSTechnology al enviar un 5 se ejecuta una acción en el hardware; en el programa de java al

enviar el mismo 5 no sucede nada. Se hizo una nueva prueba que consistía en enviar desde el programa JDSTechnology el 5 a través del puerto virtual COM3 y que lo recibiera el programa de java a través del puerto COM4. El resultado arrojó que el programa de java estaba recibiendo el numero 53 y el número 13. Esto es porque 53 es el código ascci del numero 5 y 53 es el de retorno. Se acondiciono el programa de Java para que enviara el 53 y el 13. La prueba fue satisfactoria.

Se concluyó que el hardware domótico debe recibir sólo códigos ASCII seguido del número 13.

### 8.3 PRUEBAS SOBRE EL MIDLET Y EL SERVLET DEL MIDLET

Estas pruebas consistían en poder mostrar en la pantalla los datos retornados por el controlador frontal. Al principio estaba planteado enviar dichos datos en un String. Sin embargo, era muy engorroso crear una función que los leyera y que identificara por ejemplo, cuantos caracteres confirmaban el código de la locación, cuantos el nombre, etc. Se pensó en la alternativa de enviarlos utilizando XML. Para que el celular pudiera leer un XML era necesario importar una librería llamada kxml2, la cual fue bajada de Internet.

Se creo un archivo xml con la siguiente estructura:

```
<?xml version="1.0"?>
<domotica>
<codigo>L-0001</codigo>
<nombre>luz cuarto principal</nombre>
</domótica>
```

El archivo fue leído correctamente por el midlet y mostrado en pantalla. Sin embargo, fue necesario modificar la estructura el XML para que cualquier mensaje pudiera ser enviado y para que el midlet pudiera saber si se estaba enviando un código de un dispositivo, o uno de un usuario, o uno de una locación, etc. Esto se hizo adhiriendo una etiqueta. La nueva estructura XML quedo de la siguiente manera:

```
<?xml version="1.0"?>
<domotica>
<dispositivo>
<codigo>L-0001</codigo>
<nombre>luz cuarto principal</nombre>
</dispositivo>
</domótica>
```

De esta forma el midlet sabía que los datos enviados eran de un dispositivo. Para enviar un dato de una locación, bastaba con poner la etiqueta <locacion>.

### 8.4 PRUEBAS SOBRE LA APLICACIÓN WEB Y EL SERVLET

Las pruebas hechas sobre el servlet de la aplicación Web presentaron inconvenientes iniciales los cuales se describen a continuación.

El primer error se presentó al momento de ingresar datos o extraerlos de la base de datos:

```
org.apache.jasper.JasperException: No se puede compilar la clase para JSP  
Unable to find a javac compiler;
```

Este error fue corregido copiando el archivo tools.jar desde la carpeta de la instalación de java: \lib\ hasta \common\lib dentro de la carpeta de instalación de apache. Así mismo, se presentó el error:

```
Exception: java.lang.NullPointerException
```

La excepción se corrigió copiando el archivo mysql-connector-java.jar desde la carpeta de instalación de MySQL en \MySQL Tools for 5.0\java\lib\ a la carpeta de instalación de apache en \common\lib\

Otro error común ocurrió al momento de compilar las clases creadas en java. La librería javax.servlet.http.HttpServletRequest, que funciona para que se puedan enviar y recibir requests y responses, no era reconocida por el compilador. Para solucionar el problema se copió el archivo servlet-api.jar desde la carpeta de instalación de apache en \common\lib\ hasta la carpeta de instalación de java dentro de \lib\.

Una de las pruebas más importantes en la aplicación Web se llevó a cabo para establecer si era posible que el Servlet actuara como cliente RMI y que el controlador domótico actuara como servidor. Para esto se comenzó haciendo pruebas locales sin intervención del Servlet las cuales consistieron en crear un servidor y un cliente RMI; el cliente enviaba un dato al servidor, éste lo recibía y lo mostraba en pantalla; esta prueba fue satisfactoria. Sin embargo, al momento de hacer lo mismo con el Servlet, arrojó un error:

```
java.lang.ClassCastException: $Proxy0
```

Este error sucede cuando la interfaz no extiende la clase java.rmi.Remote; al mirar cómo estaba creada la interfaz, se concluyó que sí extendía a Remote. La posible causa de este error es que el Stub que se genera al compilar la clase RMI, guarda el paquete donde ésta reside. Por ejemplo, la clase cliente.class está dentro del paquete Clientes. Al generar el Stub con rmic, se crea clienteStub.class; cada vez que se invoca el método remoto, java busca en el servidor una clase llamada clienteStub.class; este le dirá que la clase se llama cliente y que está dentro de un paquete llamado Clientes; localmente no hay problemas, pero cuando se copia este Stub en la carpeta WEB-APPS de apache-tomcat, que es donde se guardan los servlets, se presenta la excepción. Al parecer, al momento en que el Servlet llama el Stub clienteStub.class, no reconoce que está dentro del paquete Clientes, sino dentro del paquete localhost. Se buscó documentación en Internet y en libros



Diseño de un modulo para el control de iluminación de un hardware domótico a través de un teléfono celular en un ambiente distribuido

Julian Bojacá – Angela Ramirez

de Java pero no se encontraron aplicaciones que implementaran un Servlet como cliente RMI.

## CONCLUSIONES

Con la realización de este proyecto se pudo evidenciar que gracias a la nueva tecnología que ofrecen los celulares de última generación, se pueden controlar las luces de un hardware domótico y que, siguiendo los patrones de programación y metodología expuestos en este documento, se podrían también controlar los dispositivos domóticos de una casa a escala normal.

Netbeans es una plataforma confiable y amigable al momento de hacer aplicaciones en Java. No se necesita crear un CLASSPATH y su compilador marca una gran diferencia con JCreator o Eclipse debido a que corrige o muestra errores en el mismo instante en que se está escribiendo el código, no es necesario compilar cada paquete por separado sino que genera los archivos .class o .jar en una misma carpeta de proyecto, entre otros; así mismo, Netbeans trae el paquete MobilityPack para el desarrollo de aplicaciones midlet y que en éste también se pueden crear paquetes, incluir librerías, simular en vivo y exportar a un celular real.

El motor de la base de datos MySQL es muy eficiente debido a la velocidad con que busca los datos en las tablas; así mismo, MySQL mostró una estabilidad del 100% en todas las transacciones que se hicieron durante las pruebas y la implementación del software; esta base de datos ofrece muchas de las funciones y portabilidad que tienen otras más robustas como Oracle, pues se pueden emplear procedimientos almacenados, administración remota, control de pool de conexiones, permisos, triggers, entre otros.

El uso de XML para los datos que viajan desde el servidor al celular es una de las mejores maneras de enviar información a través de la Web. Básicamente un archivo XML lo puede leer cualquier plataforma lo que le da escalabilidad y portabilidad al programa. El formato con que los archivos XML manejan la información, le da más orden y estructura a los datos lo que hace que un celular cuando lee un XML, los datos contenidos en el sean mucho más manejables y fáciles de decodificar que si llegaran en una cadena de texto.

El puerto serial puede considerarse como una buena alternativa para enviar datos a un hardware. Las librerías de Java que se utilizaron para llevar a cabo esta tarea no presentaron inconvenientes en un sistema operativo como Windows XP. Sin embargo, los computadores actuales, sobretodo los portátiles, no están incluyendo en los nuevos modelos un puerto serial y las librerías no fueron probadas sobre Windows Vista; a pesar de esto, existen conversores de USB a puerto serial que se comportan de la misma manera y que no presentan conflictos con el hardware domótico; estos conversores dieron la misma utilidad que un puerto serial normal y Java no generó errores al enviar datos al hardware domótico.

Se hace necesario que si el cliente está detrás de un firewall, los puertos 8080, 3306 estén abiertos de lo contrario la aplicación que corre en el controlador domótico presentará excepciones de código pues estos son los puertos usados por Internet y la base de datos.

Luego de probar toda la aplicación, se puede concluir que no se necesita tener un servidor de grandes características de hardware pues el que se uso fue un Pentium III de 380Mhz con 256Mb en RAM y una tarjeta 10/100; así mismo, el celular no requiere un espacio en disco significativo pues la aplicación solo ocupa 73Kb. El nokia6080 donde corrió la aplicación Midlet, presentó tiempos de respuesta rápidos y en ninguna de las pruebas se notó congelamiento o caída del programa lo que demuestra que cualquier celular con características similares puede ser empleado.

Las librerías de Java para leer documentos XML funcionaron muy bien al ser empleadas; en el caso de los celulares que soportan aplicaciones .JAR, es totalmente compatible el uso de la librería kxml2 para leer dichos documentos XML.

Más de el 90% de la investigación sobre el código utilizado en este proyecto fue tomado de referencias de Internet, sobretodo en foros de Java y en el idioma Inglés; los errores que se presentaron en el transcurso del desarrollo del código ya habían sido comunes a otras personas. Sin embargo, los libros de programación no contemplan esta clase de errores por lo que Internet es una herramienta mucho más enriquecedora y con más información.

## RECOMENDACIONES

Se debe tener en cuenta que realizar aplicaciones para celulares que requieran acceso a Internet implica configurar un servidor apache. Muchos de los proveedores de Internet asignan una IP pública a sus clientes pero ésta no es fija; así mismo, se debe configurar el módem de banda ancha o fibra óptica para que pueda recibir peticiones a través de los diferentes puertos como el 8080 de Internet.

Las aplicaciones cliente-servidor funcionan muy bien cuando tanto el cliente como el servidor corren en la misma máquina. Sin embargo, se debe ser muy cuidadoso pues al correrlos en diferentes máquinas, se deben considerar los puertos que se utilizan, pues por ejemplo, MySQL no acepta conexiones remotas a menos que se den esos permisos en la base de datos; igualmente existen firewalls que bloquean la mayoría de estos puertos por lo que es una consideración bien importante a tener en cuenta antes de realizar un proyecto que implique conexiones remotas.

No siempre un programa que corra en el simulador de MobilityPack funciona igual cuando se prueba en un celular real, por lo que se recomienda en lo posible no limitarse solo al simulador.

Muchos de los errores que se presentan al momento de programar en Java se encuentran documentados y solucionados en foros en el idioma inglés.

## **PROYECCIONES**

El hardware domótico solo recibe datos a través del puerto serial; sin embargo, está en la capacidad de adaptar nuevas interfaces como reconocimiento de voz, por lo que realizar una investigación relacionada con esta nueva interfaz se podría considerar como la siguiente etapa en el desarrollo de aplicaciones para domótica.

El sistema está estructurado de tal forma que se puedan controlar, a parte de las luces, más dispositivos domóticos. El hardware domótico puede ser adaptado para que puedan controlarse dispositivos reales como bombillas, puertas, alarmas, entre otros. De esta forma, se puede desarrollar un Midlet que controle esta clase de dispositivos.

El uso de Servlets es una buena alternativa para trabajar con aplicaciones Midlet. A pesar de su gran potencial, los Servlets solo están enfocados a Java; para poder desarrollar aplicaciones en otros lenguajes una buena solución podría ser el uso de Web Services con la ventaja que utilizan el mismo protocolo http lo que solucionaría el problema de los firewalls. Así mismo, los Web Services son independientes del lenguaje de programación con que se desarrollen.

## **BIBLIOGRAFIA**

### **LIBROS**

BATES, Regis J. (Bud). Comunicaciones inalámbricas de banda ancha. Madrid: McGraw-Hill, 2003. 345 p.

KEOGH, Jim. J2EE. Manual de Referencia. Madrid: McGraw-Hill, 2003. 803 p.

PRIETO MARTIN, Manuel Jesús. Desarrollo de juegos con J2ME. Java 2 Micro Edition. México: Alfaomega Grupo Editor, 2005. 333 p.

QUINTAS, Agustín Froufe y CARDENES, Patricia Jorge. J2ME Java 2 Micro Edition. Manual de Usuario y tutorial. México: Alfaomega Grupo Editor, 2004. 572 p.

TANENBAUM, Andrew S. Sistemas Operativos Distribuidos. México: Prentice-Hall. 1996. 617 p.

### **TESIS DE GRADO**

ORTEGA, Jimmy y CARDENAS, Eduardo. Aplicación para consultas de sitios de entretenimiento desde dispositivos móviles. 2006. 137p. Trabajo de Grado (Ingeniero de Sistemas) Universidad El Bosque. Facultad de Ingeniería de Sistemas

RODRIGUEZ, Julio y CALVACHE Miguel. Simulación de la transmisión de datos en una red gprs por medio de ns2. 2006. 108p. Trabajo de Grado (Ingeniero de Sistemas) Universidad El Bosque. Facultad de Ingeniería de Sistemas

### **INTERNET**

AGUIRRE, José Eduardo. Redes Inalámbricas. [en línea]. (Consultado en Abril de 2007). <http://www.monografias.com/trabajos/redesinalam/redesinalam.shtml>.

AIELLO, M. The role of web services at home. University of Trento. [en línea]. (Consultado en Abril de 2007). <http://eprints.biblio.unitn.it/archive/00000886>

ALTADIL A., FÉLIZ M. y MOLINA M.. Será el hogar digital un lugar sostenible? Telefónica de España. [en línea]. (Consultado en Febrero de 2007). <http://www.tid.es/documentos/sostenibilidad/hogar.pdf>

Asociación de la Industria Celular en Colombia. [en línea]. (Consultado en Febrero de 2007). <http://www.asocel.org.co/prensa.php>

Bluetooth. [en línea]. (Consultado en Junio de 2007). <http://es.wikipedia.org/wiki/Bluetooth>

Casadomo. [en línea]. (Consultado en Marzo de 2007). <http://www.casadomo.com/promocionesDetalle.aspx?id=7&idm=122&pat=122&prov=&sec=&year.>

Casadomo. [en línea]. (Consultado en Marzo de 2007). <http://www.casadomo.com/noticiasDetalle.aspx?id=8628&c=6&idm=10&pat=10>

Comisión Reguladora de Telecomunicaciones. [en línea]. (Consultado en Abril de 2007). [http://www.crt.gov.co/crt\\_2001-2004/paginas/internas/biblioteca/regulatorio\\_a.htm](http://www.crt.gov.co/crt_2001-2004/paginas/internas/biblioteca/regulatorio_a.htm)

GPRS Arquitectura. [en línea]. (Consultado en Mayo de 2007). [www.mailxmail.com/curso/informatica/gsm/capitulo6.htm+arquitectura%2Bgsm&hl=es&ct=clnk&cd=2&gl=us](http://www.mailxmail.com/curso/informatica/gsm/capitulo6.htm+arquitectura%2Bgsm&hl=es&ct=clnk&cd=2&gl=us)

Grun Technik. [en línea]. (Consultado en Abril de 2006). <http://www.gruntechnik.com/estandares.htm>

Historia del Teléfono móvil. [en línea]. (Consultado en Marzo de 2007). [http://es.wikipedia.org/wiki/Historia\\_del\\_tel%C3%A9fono\\_m%C3%B3vil](http://es.wikipedia.org/wiki/Historia_del_tel%C3%A9fono_m%C3%B3vil)

Homewireless E.U. [en línea]. (Consultado en Abril de 2007). <http://www.catalogodetecnologia.com/procesos/link.aspx?ie=4624919&idi=5&type=1&web=www.tuhogardigital.com>

JINI. [en línea]. (Consultado en Mayo de 2007). [http://www.jini.org/wiki/Main\\_Page](http://www.jini.org/wiki/Main_Page)

KLEINDIENST, Jan., MACEK, Tomáš., SERÉDI, Ladislav y ŠEDIVÝ, Jan. Vision-Enhanced Multi-Modal Interactions in Domotic Environments. [en línea]. (Consultado en Marzo de 2007). [http://ui4all.ics.forth.gr/workshop2004/files/ui4all\\_proceedings/adjunct/interactive\\_applications/103.pdf](http://ui4all.ics.forth.gr/workshop2004/files/ui4all_proceedings/adjunct/interactive_applications/103.pdf)

LSB. [en línea]. (Consultado en Abril de 2007). [http://www.lsb.es/subhome\\_hogar.asp](http://www.lsb.es/subhome_hogar.asp)

Lonwork. [en línea]. (Consultado en Abril de 2006). <http://odisea.ii.uam.es/esp/recursos/Lonwork.htm>

Medios de Transmisión/Componentes. [en línea]. (Consultado en Abril de 2006). <http://tecnologias.gio.etsit.upm.es/domotica/>

Microsoft Mobile. [en línea]. (Consultado en Abril de 2007). <http://www.microsoft.com/windowsmobile/default.mspx>

Proxima Systems. [en línea]. (Consultado en Enero de 2007). <http://www.proximasystems.net/es/productos/lincegprs.php>

R. JIMENO, Z. SALVADOR, A. LAFUENTE, M. LARREA, A. URIBARREN. An architecture for the personalized control of domotic resources. Noviembre 2004. Eindhoven, The Netherlands. [en línea]. (Consultado en Febrero de 2007). <http://portal.acm.org/citation.cfm?id=1031432>.

Red Inalámbrica. [en línea]. (Consultado en Abril de 2007). [http://es.wikipedia.org/wiki/Red\\_inal%C3%A1mbrica](http://es.wikipedia.org/wiki/Red_inal%C3%A1mbrica)

RIM. [en línea]. (Consultado en Abril de 2007). <http://www.rim.com>

Sistemas Distribuidos. [en línea]. (Consultado en Mayo de 2007). <http://www.monografias.com/trabajos16/sistemas-distribuidos/sistemas-distribuidos.shtml>

Superintendencia de Industria y Comercio. [en línea]. (Consultado en Abril de 2007). [http://www.sic.gov.co/Articulos\\_Pagina\\_Principal/Noticias/2006/Telefonia/Operadores\\_telefonia.php](http://www.sic.gov.co/Articulos_Pagina_Principal/Noticias/2006/Telefonia/Operadores_telefonia.php)

Symbian. [en línea]. (Consultado en Abril de 2007). <http://www.symbian.com/>

TechnoImport. [en línea]. (Consultado en Marzo de 2007). [http://www.latinpyme.com.co/asp\\_noticia.asp?ite\\_id=8907&pla\\_id=2&cat\\_id=5474&cat\\_nom=Actualidad](http://www.latinpyme.com.co/asp_noticia.asp?ite_id=8907&pla_id=2&cat_id=5474&cat_nom=Actualidad)

Technoimport. [en línea]. (Consultado en Abril de 2007). <http://www.technoimport.com.co/>

Tecnología de acceso celular. [en línea]. (Consultado en Marzo de 2007). <http://www.yucatan.com.mx/especiales/celular/tecnologiascelulares.asp>

Tecnología GSM. [en línea]. (Consultado en Junio de 2007). <http://www.casadomo.com/noticiasDetalle.aspx?c=61&m=164&idm=73&pat=148&n2=148>



Telefonía Celular. [en línea]. (Consultado en Febrero de 2007).  
<http://www.monografias.com/trabajos34/telefonía-celular/telefonía-celular.shtml>

Telefonía móvil 2G. [en línea]. (Consultado en Junio de 2007).  
[http://es.wikipedia.org/wiki/Telefon%C3%ADa\\_m%C3%B3vil\\_2G](http://es.wikipedia.org/wiki/Telefon%C3%ADa_m%C3%B3vil_2G)

Telefonía móvil 3G. [en línea]. (Consultado en Junio de 2007).  
[http://es.wikipedia.org/wiki/Telefon%C3%ADa\\_m%C3%B3vil\\_3G](http://es.wikipedia.org/wiki/Telefon%C3%ADa_m%C3%B3vil_3G)

Telefonía móvil 4G. [en línea]. (Consultado en Junio de 2007).  
[http://es.wikipedia.org/wiki/Telefon%C3%ADa\\_m%C3%B3vil\\_4G](http://es.wikipedia.org/wiki/Telefon%C3%ADa_m%C3%B3vil_4G)

WIFI. [en línea]. (Consultado en Junio de 2007).  
<http://72.14.253.104/search?q=cache:IDRayDVChhQJ:es.wikipedia.org/wiki/Wi-Fi+redes%2Binalambricas&hl=es&ct=clnk&cd=3&gl=us>

WiMAX. [en línea]. (Consultado en Junio de 2007).  
<http://es.wikipedia.org/wiki/WiMAX>

Wireless Application Protocol. [en línea]. (Consultado en Abril de 2007).  
<http://es.wikipedia.org/wiki/WAP>

## APÉNDICE A DOCUMENTACIÓN CASOS DE USO

La descripción de cada uno de los casos de uso definidos anteriormente en los requerimientos funcionales están a continuación:

*Actor Administrador* escenario *Administración de Locaciones*

**Tabla 1. Caso de uso crear locación**

Identificador del caso de uso:	CUA1	
Nombre caso de uso:	Crear locación.	
Prioridad y Tipo:	Alta, necesario	
Descripción:	El administrador crea las diferentes locaciones a donde pertenecerán los dispositivos de iluminación y los grupos de usuario.	
Curso Básico Eventos:	<b>ACTOR</b>	<b>SISTEMA</b>
		1. El sistema muestra el formulario de crear nueva locación.
	2. El administrador llena los campos CODIGO y NOMBRE.	
		3. El sistema guarda la nueva locación.
Caminos de Excepción:	El administrador cancela la acción de crear locación.	
Suposiciones:	La sesión del administrador está activa. El administrador escoge la opción AGREGAR NUEVA LOCACION.	
Pre-condiciones:	La locación no existe creada en el sistema para el administrador.	
Post-condiciones:	El sistema muestra confirmación de creación. El sistema muestra la lista de locaciones creadas.	
Autor:	Julian Bojacá – Angela Ramírez	

**Tabla 2. Caso de uso consultar locación**

Identificador del caso de uso:	CUA2	
Nombre caso de uso:	Consultar locación.	
Prioridad y Tipo:	Media, deseable.	
Descripción:	El administrador puede consultar las locaciones creadas en el sistema.	
Curso Básico Eventos:	<b>ACTOR</b>	<b>SISTEMA</b> 1. El sistema muestra la lista de locaciones creadas.
Caminos de Excepción:		
Suposiciones:	La sesión del administrador está activa. El administrador escoge la opción ADMINISTRAR LOCACIONES.	
Pre-condiciones:	Existe al menos una locación en el sistema creada.	
Post-condiciones:	El sistema da la opción de ver los detalles de las locaciones listadas.	
Autor:	Julian Bojacá – Angela Ramírez	

**Tabla 3. Caso de uso editar locación**

Identificador del caso de uso:	CUA3	
Nombre caso de uso:	Editar locación.	
Prioridad y Tipo:	Media, deseable.	
Descripción:	El administrador puede editar el nombre de las locaciones creadas.	
Curso Básico Eventos:	<b>ACTOR</b>	<b>SISTEMA</b> 1. El sistema muestra el campo de texto NOMBRE para que se pueda modificar. 2. El administrador cambia

	<table border="1"> <tr> <td>el nombre de la locación.</td><td></td></tr> <tr> <td></td><td>3. El sistema guarda el nuevo nombre de locación.</td></tr> </table>	el nombre de la locación.			3. El sistema guarda el nuevo nombre de locación.
el nombre de la locación.					
	3. El sistema guarda el nuevo nombre de locación.				
Caminos de Excepción:	El usuario cancela la acción.				
Suposiciones:	La sesión del administrador está activa. El administrador escoge una locación y ve los DETALLES. El administrador selecciona la opción EDITAR.				
Pre-condiciones:					
Post-condiciones:	El sistema muestra confirmación de edición. El sistema muestra el nuevo nombre de la locación.				
Autor:	Julian Bojacá – Angela Ramírez				

**Tabla 4. Caso de uso eliminar locación**

Identificador del caso de uso:	CUA4	
Nombre caso de uso:	Eliminar locación.	
Prioridad y Tipo:	Media, deseable.	
Descripción:	El administrador puede eliminar las locaciones creadas en el sistema.	
Curso Básico Eventos:	<b>ACTOR</b>	<b>SISTEMA</b>
	1. El administrador selecciona una locación a eliminar.	
		3. El sistema elimina la locación del sistema.
Caminos de Excepción:	El usuario cancela la acción de eliminar locación. El sistema encuentra dispositivos o grupos de usuarios en esa locación.	
Suposiciones:	La sesión del administrador está activa.	
Pre-condiciones:	No deben existir dispositivos en la locación. No deben existir grupos de usuarios en la locación.	
Post-condiciones:	El sistema muestra la nueva lista de locaciones.	

Autor:	Julian Bojacá – Angela Ramírez
--------	--------------------------------

Actor *Administrador* escenario *Administración de Dispositivos*

**Tabla 5. Caso de uso crear dispositivo**

Identificador del caso de uso:	CUA5	
Nombre caso de uso:	Crear dispositivo.	
Prioridad y Tipo:	Alta, necesario	
Descripción:	El administrador crea las diferentes dispositivos de iluminación los cuales serán controlados desde el midlet.	
Curso Básico Eventos:	<b>ACTOR</b>	<b>SISTEMA</b>
		1. El sistema muestra el formulario de crear nuevo dispositivo.
	2. El administrador llena los campos CODIGO, NOMBRE y LOCACIÓN.	
		3. El sistema guarda el nuevo dispositivo.
Caminos de Excepción:	El administrador cancela la acción de crear dispositivo.	
Suposiciones:	La sesión del administrador está activa. El administrador escoge la opción AGREGAR NUEVO DISPOSITIVO.	
Pre-condiciones:	El dispositivo no existe creado en el sistema para el administrador. Existen locaciones creadas.	
Post-condiciones:	El sistema muestra confirmación de creación. El sistema muestra la lista de dispositivos creados.	
Autor:	Julian Bojacá – Angela Ramírez	

**Tabla 6. Caso de uso consultar dispositivo**

Identificador del caso de uso:	CUA6
Nombre caso de uso:	Consultar dispositivo.

uso:		
Prioridad y Tipo:	Media, deseable.	
Descripción:	El administrador puede consultar los dispositivos creados en el sistema.	
Curso Básico Eventos:	<b>ACTOR</b>	<b>SISTEMA</b>
		1. El sistema muestra la lista de dispositivos creados.
Caminos de Excepción:		
Suposiciones:	La sesión del administrador está activa. El administrador hace escoge la opción ADMINISTRAR DISPOSITIVOS.	
Pre-condiciones:	Existe al menos un dispositivo en el sistema creado.	
Post-condiciones:	El sistema da la opción de ver los detalles de los dispositivos creados en el sistema.	
Autor:	Julian Bojacá – Angela Ramírez	

**Tabla 7. Caso de uso editar dispositivo**

Identificador del caso de uso:	CUA7	
Nombre caso de uso:	Editar dispositivo.	
Prioridad y Tipo:	Media, deseable.	
Descripción:	El administrador puede editar el nombre de los dispositivos creados o la locación a la que pertenecen.	
Curso Básico Eventos:	<b>ACTOR</b>	<b>SISTEMA</b>
		1. El sistema muestra el campo de texto NOMBRE y LOCACION para que se puedan modificar.
	2. El administrador cambia el nombre del dispositivo o la locación a la que pertenece	
		3. El sistema guarda los cambios.

Caminos de Excepción:	El usuario cancela la acción.
Suposiciones:	La sesión del administrador está activa. El administrador escoge un dispositivo y ve los DETALLES. El administrador escoge la opción EDITAR.
Pre-condiciones:	
Post-condiciones:	El sistema muestra confirmación de edición. El sistema muestra los cambios hechos al dispositivo.
Autor:	Julian Bojacá – Angela Ramírez

**Tabla 8. Caso de uso eliminar dispositivo**

Identificador del caso de uso:	CUA8							
Nombre caso de uso:	Eliminar dispositivo.							
Prioridad y Tipo:	Media, deseable.							
Descripción:	El administrador puede eliminar los dispositivos creados en el sistema.							
Curso Básico Eventos:	<table><tr><td><b>ACTOR</b></td><td><b>SISTEMA</b></td></tr><tr><td>1. El administrador selecciona un dispositivo a eliminar.</td><td></td></tr><tr><td></td><td>3. El sistema elimina el dispositivo del sistema.</td></tr></table>		<b>ACTOR</b>	<b>SISTEMA</b>	1. El administrador selecciona un dispositivo a eliminar.			3. El sistema elimina el dispositivo del sistema.
<b>ACTOR</b>	<b>SISTEMA</b>							
1. El administrador selecciona un dispositivo a eliminar.								
	3. El sistema elimina el dispositivo del sistema.							
Caminos de Excepción:	El usuario cancela la acción de eliminar dispositivo. El sistema encuentra grupos de usuarios asignados a ese dispositivo.							
Suposiciones:	La sesión del administrador está activa.							
Pre-condiciones:	No deben existir grupos de usuarios asignados a ese dispositivo.							
Post-condiciones:	El sistema elimina los permisos que tengan los grupos de usuario sobre el dispositivo. El sistema muestra la nueva lista de dispositivos.							
Autor:	Julian Bojacá – Angela Ramírez							

Actor *Administrador* escenario *Administración de Usuarios*

**Tabla 9. Caso de uso crear grupo de usuario**

Identificador del caso de uso:	CUA9	
Nombre caso de uso:	Crear grupo de usuario.	
Prioridad y Tipo:	Alta, necesario	
Descripción:	El administrador crea los grupos de usuario a los cuales pertenecerán los usuarios.	
Curso Básico Eventos:	<b>ACTOR</b>	<b>SISTEMA</b>
		1. El sistema muestra el formulario de crear nuevo grupo de usuario.
	2. El administrador llena los campos CODIGO, NOMBRE y LOCACIÓN.	
		3. El sistema guarda el nuevo grupo de usuario.
Caminos de Excepción:	El administrador cancela la acción de crear grupo de usuario.	
Suposiciones:	La sesión del administrador está activa. El administrador escoge la opción AGREGAR NUEVO GRUPO DE USUARIO.	
Pre-condiciones:	El grupo de usuario no existe creado en el sistema para el administrador. Existen locaciones creadas.	
Post-condiciones:	El sistema muestra confirmación de creación. El sistema muestra la lista de grupos de usuarios creados.	
Autor:	Julian Bojacá – Angela Ramírez	

**Tabla 10. Caso de uso consultar grupo de usuario**

Identificador del caso de uso:	CUA10
Nombre caso de	Consultar grupo de usuario.



uso:		
Prioridad y Tipo:	Media, deseable.	
Descripción:	El administrador puede consultar los grupos de usuario creados en el sistema.	
Curso Básico Eventos:	<b>ACTOR</b>	<b>SISTEMA</b> 1. El sistema muestra la lista de grupos de usuario creados.
Caminos de Excepción:		
Suposiciones:	La sesión del administrador está activa. El administrador escoge la opción ADMINISTRAR USUARIOS.	
Pre-condiciones:	Existe al menos un grupo de usuario en el sistema creado.	
Post-condiciones:	El sistema da la opción de ver los detalles de los grupos de usuario creados en el sistema.	
Autor:	Julian Bojacá – Angela Ramírez	

**Tabla 11. Caso de uso editar grupo de usuario**

Identificador del caso de uso:	CUA11	
Nombre caso de uso:	Editar grupo de usuario.	
Prioridad y Tipo:	Media, deseable.	
Descripción:	El administrador puede editar el nombre de los grupos de usuario creados en el sistema o la locación a la que pertenecen.	
Curso Básico Eventos:	<b>ACTOR</b>	<b>SISTEMA</b> 1. El sistema muestra el campo de texto NOMBRE y LOCACION para que se puedan modificar. 2. El administrador cambia el nombre del grupo o la locación a la que pertenece.

		3. El sistema guarda los cambios.
Caminos de Excepción:	El usuario cancela la acción.	
Suposiciones:	La sesión del administrador está activa. El administrador escoge un grupo de usuario y ve los DETALLES. El administrador hace clic en EDITAR.	
Pre-condiciones:		
Post-condiciones:	El sistema muestra confirmación de edición. El sistema muestra los cambios hechos al grupo de usuario.	
Autor:	Julian Bojacá – Angela Ramírez	

**Tabla12. Caso de uso eliminar grupo de usuario**

Identificador del caso de uso:	CUA12	
Nombre caso de uso:	Eliminar grupo de usuario.	
Prioridad y Tipo:	Media, deseable.	
Descripción:	El administrador puede eliminar los grupos de usuario creados en el sistema.	
Curso Básico Eventos:	<b>ACTOR</b>	<b>SISTEMA</b>
	1. El administrador selecciona un grupo de usuario a eliminar.	
		3. El sistema elimina el grupo de usuario del sistema.
Caminos de Excepción:	El administrador cancela la acción de eliminar grupo de usuario. El sistema encuentra usuarios asignados a ese dispositivo.	
Suposiciones:	La sesión del administrador está activa.	
Pre-condiciones:	No deben existir usuarios asignados al grupo de usuarios.	
Post-condiciones:	El sistema elimina los permisos que tiene el grupo sobre los dispositivos de iluminación.	

	El sistema muestra la nueva lista de grupo de usuario.
Autor:	Julian Bojacá – Angela Ramírez

**Tabla 13. Caso de uso crear usuario**

Identificador del caso de uso:	CUA13	
Nombre caso de uso:	Crear usuario.	
Prioridad y Tipo:	Alta, necesario	
Descripción:	El administrador crea los usuarios que controlarán los dispositivos de iluminación.	
Curso Básico Eventos:	<b>ACTOR</b>	<b>SISTEMA</b>
		1. El sistema muestra el formulario de crear nuevo usuario.
	2. El administrador llena los campos NOMBRE, LOGIN, CONTRASEÑA y GRUPO	
		3. El sistema guarda el nuevo usuario.
Caminos de Excepción:	El administrador cancela la acción de crear usuario.	
Suposiciones:	La sesión del administrador está activa. El administrador ha ingresado al modulo de ADMINISTRAR USUARIOS. El administrador escoge la opción AGREGAR NUEVO USUARIO.	
Pre-condiciones:	El usuario no existe creado en el sistema para el administrador. Existen grupos de usuarios creados.	
Post-condiciones:	El sistema muestra confirmación de creación. El sistema muestra la lista de usuarios creados en le grupo.	
Autor:	Julian Bojacá – Angela Ramírez	

**Tabla 14. Caso de uso consultar usuario**

Identificador del caso de uso:	CUA14
--------------------------------	-------

Nombre caso de uso:	Consultar usuario.	
Prioridad y Tipo:	Media, deseable.	
Descripción:	El administrador puede consultar los usuarios creados en el sistema.	
Curso Básico Eventos:	<b>ACTOR</b>	<b>SISTEMA</b> 1. El sistema muestra la lista de usuarios creados.
Caminos de Excepción:		
Suposiciones:	La sesión del administrador está activa. El administrador ha ingresado a ADMINISTRAR USUARIOS. Existe al menos un grupo de usuario en el sistema creado.	
Pre-condiciones:	Existe al menos un usuario creado en el grupo.	
Post-condiciones:		
Autor:	Julian Bojacá – Angela Ramírez	

**Tabla 15. Caso de uso editar usuario**

Identificador del caso de uso:	CUA15	
Nombre caso de uso:	Editar usuario.	
Prioridad y Tipo:	Media, deseable.	
Descripción:	El administrador puede editar el nombre de los usuarios, la contraseña o el grupo al que pertenece.	
Curso Básico Eventos:	<b>ACTOR</b>	<b>SISTEMA</b> 1. El sistema muestra el campo de texto NOMBRE, CONTRASEÑA y GRUPO para que se puedan modificar. 2. El administrador cambia el nombre, la contraseña o el grupo al que pertenece el usuario.

		3. El sistema guarda los cambios.
Caminos de Excepción:	El usuario cancela la acción.	
Suposiciones:	La sesión del administrador está activa. El administrador escoge la opción ADMINISTRAR USUARIOS. El administrador escoge un grupo de usuario y ve los DETALLES.	
Pre-condiciones:	El administrador escoge un usuario.	
Post-condiciones:	El sistema muestra confirmación de edición. El sistema muestra los cambios hechos al usuario.	
Autor:	Julian Bojacá – Angela Ramírez	

**Tabla16. Caso de uso eliminar usuario**

Identificador del caso de uso:	CUA16	
Nombre caso de uso:	Eliminar usuario.	
Prioridad y Tipo:	Media, deseable.	
Descripción:	El administrador puede eliminar usuarios creados en el sistema.	
Curso Básico Eventos:	<b>ACTOR</b>	<b>SISTEMA</b>
	1. El administrador selecciona un usuario a eliminar.	
		3. El sistema elimina el usuario del sistema.
Caminos de Excepción:	El administrador cancela la acción de eliminar usuario.	
Suposiciones:	La sesión del administrador está activa.	
Pre-condiciones:		
Post-condiciones:	El sistema muestra la nueva lista de usuarios del grupo seleccionado.	

Autor:	Julian Bojacá – Angela Ramírez
--------	--------------------------------

Actor *Administrador* escenario *Administración de Permisos sobre Dispositivos*

**Tabla 17. Caso de uso otorgar permiso sobre dispositivo**

Identificador del caso de uso:	CUA17	
Nombre caso de uso:	Otorgar permiso sobre dispositivo.	
Prioridad y Tipo:	Alta, necesario	
Descripción:	El administrador otorga a los grupos de usuario los permisos sobre los dispositivos de iluminación.	
Curso Básico Eventos:	<b>ACTOR</b>	<b>SISTEMA</b>
		1. El sistema muestra el formulario de asignación de permisos sobre dispositivo con los campos: apagar, encender, ver estado y programar.
	2. El administrador selecciona los permisos a otorgar al grupo de usuarios.	
		3. El sistema guarda los permisos.
Caminos de Excepción:	El administrador cancela la acción de otorgar permiso sobre dispositivo.	
Suposiciones:	La sesión del administrador está activa. El administrador ha ingresado al modulo de ADMINISTRAR DISPOSITIVOS. El administrador ha escogido ver los DETALLES de un dispositivo.	
Pre-condiciones:	Debe existir un grupo de usuario creado.	
Post-condiciones:	El sistema muestra confirmación de otorgamiento de permisos. El sistema muestra la lista de grupos y los permisos sobre el dispositivo seleccionado.	
Autor:	Julian Bojacá – Angela Ramírez	

**Tabla 18. Caso de uso editar permiso sobre dispositivo**

Identificador del caso de uso:	CUA18	
Nombre caso de uso:	Editar permiso sobre dispositivo.	
Prioridad y Tipo:	Media, necesario	
Descripción:	El administrador modifica los permisos de los grupos de usuario sobre los dispositivos de iluminación.	
Curso Básico Eventos:	<b>ACTOR</b>	<b>SISTEMA</b>
		1. El sistema muestra el formulario de asignación de permisos sobre dispositivo con los campos: apagar, encender, ver estado y programar.
	2. El administrador selecciona los permisos a otorgar o quitar.	
		3. El sistema guarda las modificaciones.
Caminos de Excepción:	El administrador cancela la acción de editar permiso sobre dispositivo.	
Suposiciones:	La sesión del administrador está activa. El administrador ha ingresado al modulo de ADMINISTRAR DISPOSITIVOS. El administrador ha escogido ver los DETALLES de un dispositivo.	
Pre-condiciones:	Al grupo de usuarios previamente se le dieron permisos sobre el dispositivo.	
Post-condiciones:	El sistema muestra confirmación de modificación de permisos. El sistema muestra la lista de grupos y los permisos sobre el dispositivo seleccionado.	
Autor:	Julian Bojacá – Angela Ramírez	

**Tabla 19. Caso de uso eliminar permiso sobre dispositivo**

Identificador del caso de uso:	CUA19	
Nombre caso de uso:	Eliminar permiso sobre dispositivo.	
Prioridad y Tipo:	Media, necesario	
Descripción:	El administrador quita los permisos de los grupos de usuario sobre los dispositivos de iluminación.	
Curso Básico Eventos:	<b>ACTOR</b>	<b>SISTEMA</b>
	1. El administrador escoge el grupo de usuario al cual quiere quitarle los permisos sobre el dispositivo.	
		2. El sistema elimina los permisos sobre el dispositivo del grupo de usuarios seleccionado.
Caminos de Excepción:	El administrador cancela la acción de eliminar permiso sobre dispositivo.	
Suposiciones:	La sesión del administrador está activa. El administrador ha ingresado al modulo de ADMINISTRAR DISPOSITIVOS. El administrador ha escogido ver los DETALLES de un dispositivo.	
Pre-condiciones:	Al grupo de usuarios previamente se le dieron permisos sobre el dispositivo.	
Post-condiciones:	El sistema muestra confirmación de eliminación de permisos. El sistema muestra la nueva lista de grupos de usuarios con permisos sobre el dispositivo.	
Autor:	Julian Bojacá – Angela Ramírez	

Actor *Usuario* escenario *Administración de Interacción con el hardware domótico*

**Tabla 20. Caso de uso ver detalles de locación**

Identificador del caso de uso:	CUU1
--------------------------------	------



Nombre caso de uso:	Ver detalles de locación.									
Prioridad y Tipo:	Baja, opcional									
Descripción:	El usuario ve los detalles de la locación.									
Curso Básico Eventos:	<table><tr><th>ACTOR</th><th>SISTEMA</th></tr><tr><td></td><td>1. El sistema muestra la lista de locaciones a las cuales el usuario tiene acceso.</td></tr><tr><td>2. El usuario selecciona la locación.</td><td></td></tr><tr><td></td><td>3. El sistema muestra los detalles de la locación seleccionada.</td></tr></table>		ACTOR	SISTEMA		1. El sistema muestra la lista de locaciones a las cuales el usuario tiene acceso.	2. El usuario selecciona la locación.			3. El sistema muestra los detalles de la locación seleccionada.
ACTOR	SISTEMA									
	1. El sistema muestra la lista de locaciones a las cuales el usuario tiene acceso.									
2. El usuario selecciona la locación.										
	3. El sistema muestra los detalles de la locación seleccionada.									
Caminos de Excepción:	No se establece conexión con el servidor.									
Suposiciones:	El usuario se ha validado en el servidor. El administrador ha creado previamente la locación.									
Pre-condiciones:	El usuario ha seleccionado una locación de la lista. El grupo de usuarios al que pertenece el usuario debe estar asignado a la locación.									
Post-condiciones:										
Autor:	Julian Bojacá – Angela Ramírez									

**Tabla 21. Caso de uso ver lista de dispositivos**

Identificador del caso de uso:	CUU2		
Nombre caso de uso:	Ver lista de dispositivos.		
Prioridad y Tipo:	Alta, necesario		
Descripción:	El usuario ve la lista de los dispositivos asignados a su grupo.		
Curso Básico Eventos:	<b>ACTOR</b>		<b>SISTEMA</b>
			1. El sistema muestra la lista de dispositivos a las cuales el usuario tiene acceso.
	2. El usuario selecciona el dispositivo.		

Caminos de Excepción:	No se establece conexión con el servidor.
Suposiciones:	El usuario se ha validado en el servidor. El administrador ha creado previamente la locación.
Pre-condiciones:	El grupo de usuarios al que pertenece el usuario debe estar asignado a la locación.
Post-condiciones:	
Autor:	Julian Bojacá – Angela Ramírez

**Tabla 22. Caso de uso ver detalles de dispositivo**

Identificador del caso de uso:	CUU3	
Nombre caso de uso:	Ver detalles de dispositivo.	
Prioridad y Tipo:	Alta, necesario	
Descripción:	El usuario ve los detalles del dispositivo seleccionado como el estado actual, el usuario que realizó la última acción sobre este, etc.	
Curso Básico Eventos:	<b>ACTOR</b>	<b>SISTEMA</b>
		1. El sistema muestra la lista de dispositivos a las cuales el usuario tiene acceso.
	2. El usuario selecciona el dispositivo.	
		3. El sistema muestra los detalles del dispositivo como código del dispositivo, nombre del dispositivo, estado actual, usuario que realizó la ultima acción sobre el dispositivo y fecha de la última acción.
Caminos de Excepción:	No se establece conexión con el servidor.	
Suposiciones:	El usuario se ha validado en el servidor.	
Pre-	El grupo de usuarios al que pertenece el usuario tiene	

condiciones:	permisos de ver estado del dispositivo.
Post- condiciones:	El sistema muestra las opciones de ejecutar acción sobre el dispositivo.
Autor:	Julian Bojacá – Angela Ramírez

**Tabla 23. Caso de uso enviar orden a dispositivo**

Identificador del caso de uso:	CUU4	
Nombre caso de uso:	Enviar orden a dispositivo.	
Prioridad y Tipo:	Alta, necesario	
Descripción:	El usuario envía una orden de encender, apagar o programar al dispositivo de iluminación.	
Curso Básico Eventos:	<b>ACTOR</b>	<b>SISTEMA</b>
		1. El sistema muestra la lista de órdenes que se pueden enviar al dispositivo de iluminación.
	2. El usuario selecciona la orden a enviar.	
		3. El sistema ejecuta la orden.
Caminos de Excepción:	El hardware domótico está apagado.	
Suposiciones:	El usuario se ha validado en el servidor. El usuario tiene permiso de ver el dispositivo de iluminación.	
Pre-condiciones:	El grupo de usuarios al que pertenece el usuario tiene permisos de ejecutar la orden sobre el dispositivo de iluminación.	
Post-condiciones:	El sistema de control del hardware domótico se notifica que hay una orden de ejecución pendiente y la lleva a cabo.	
Autor:	Julian Bojacá – Angela Ramírez	

**Tabla 24. Caso de uso Programar dispositivo**

Identificador del caso de uso:	CUU5
Nombre caso de	Programar dispositivo.

uso:		
Prioridad y Tipo:	Medio, deseable	
Descripción:	El usuario puede programar las luces del hardware domótico para que se enciendan o se apaguen en determinado horario	
Curso Básico Eventos:	<b>ACTOR</b>	<b>SISTEMA</b>
		1. El sistema muestra días de la semana y horas
	2. El usuario selecciona el día y hora.	
		3. El sistema muestra la lista de acciones a ejecutar.
	4. El usuario selecciona una acción.	
		5. El sistema guarda la programación del dispositivo.
Caminos de Excepción:		
Suposiciones:	El grupo del usuario tiene permisos para ejecutar la orden programada.	
Pre-condiciones:		
Post-condiciones:	El hardware domótico y el controlador deben estar encendidos en el día y hora de la programación.	
Autor:	Julian Bojacá – Angela Ramírez	

## APENDICE B MANUALES PÁGINA WEB Y MIDLET

### 1. MANUAL PÁGINA WEB

Para que un usuario pueda empezar a utilizar esta página, deberá ingresar desde un navegador con la dirección `http://localhost:8080/domotica/JSP/login.jsp` Luego de hacer esto, lo primero que va a mostrar la página Web es una pantalla en la que se indica el nombre del usuario y la contraseña de la persona que desea entrar a administrar los dispositivos, locaciones y usuarios. Se muestra lo siguiente:

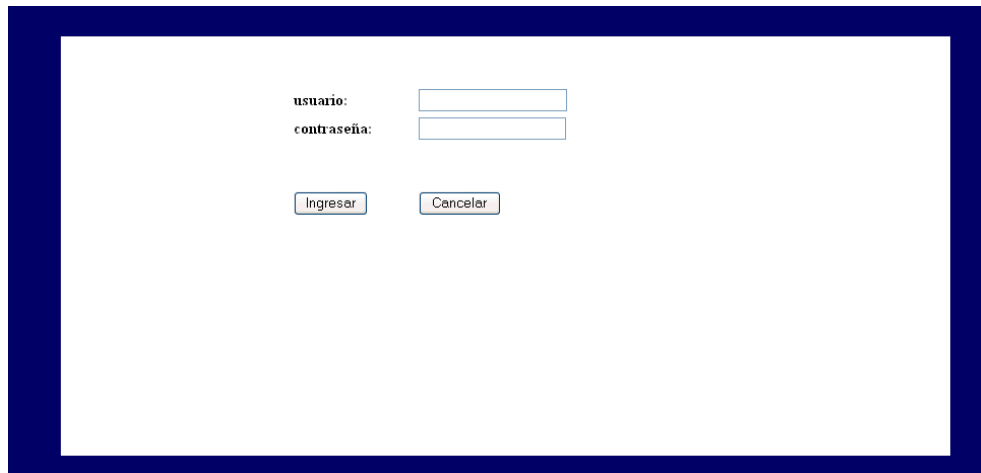
A screenshot of a web form for user authentication. The form is enclosed in a dark blue rectangular border. It contains two labels, 'usuario:' and 'contraseña:', each followed by a text input field. Below the input fields are two buttons: 'Ingresar' and 'Cancelar'.

Figura 1. Autenticación de usuario

Si la contraseña dada es incorrecta, nos mostrara una pantalla en la que se nos indica que se ha colocado mal el usuario o la contraseña (figura 2.).

A screenshot of the same web form as in Figure 1, but with an error message. The error message, 'usuario o contraseña invalidos...', is displayed in red text below the password input field. The 'Ingresar' and 'Cancelar' buttons are still present at the bottom.

Figura 2. Validación incorrecta

Si la contraseña dada es correcta, nos muestra la pantalla mostrada en la figura 3, la cual permitirá que se pueda empezar a administrar los diferentes usuarios, dispositivos y locaciones con las que podrán contar los demás usuarios que deseen interactuar con el sistema domótico.



Figura 3. Validación correcta. Menu de administración de dispositivos, usuarios y locaciones.

## 1.1 ADMINISTRACION DE DISPOSITIVOS

Según la opción que desee escoger aparecerá una nueva ventana. Si desea escoger la opción de Administración de Dispositivos, mostrara la pantalla de la figura 4., en la que estarán los dispositivos que hay en el momento implementados con sus características, además de la posibilidad de insertar un nuevo dispositivo y de volver a la página anterior, es decir, a la que se muestra en la figura 3.



Figura 4. Consulta de Dispositivos

Si se desea ver con detalle los dispositivos que hay hasta el momento lo puede hacer de dos maneras: haciendo clic en detalles o bien en el código del dispositivo. A continuación mostrara la pantalla de la figura 5., en la que además se podrá editar la información del dispositivo, eliminar el dispositivo, agregar un grupo a este dispositivo y volver a la página anterior, es decir, la mostrada en la figura 4.

**Detalle Dispositivo**

Código: L-0001  
Nombre: LUZ CUARTO DE ABUELITA FABIOLA

Estado actual: apagada  
Locacion: CASA ABUELITOS

Grupos asignados

codigo	nombre	E	A	V	P	+	X
GU-0001	HIJOS E HIJAS	●	●	●	●	+	X
GU-0002	PRIMOS	●	●	●	●	+	X
GU-0003	PAPAS	●	●	●	●	+	X

» agregar grupo a este dispositivo

« volver

Figura 5. Detalles de los dispositivos

Si se escoge la opción de editar esta información aparecerá la pantalla mostrada en la figura 6., en la que se puede modificar el nombre del dispositivo, el código de encendido, el código de apagado y la locación. Luego se le da clic en modificar y aparecerá una confirmación de que se ha modificado el dispositivo y aceptar que lo llevara a la pantalla de consulta de dispositivos, es decir, a la de la figura 4.

codigo: L-0001  
nombre: LUZ DE LA SALA  
cod encender: 57  
cod apagar: 48  
locacion: 0001

Modificar

Figura 6. Editar información del dispositivo.

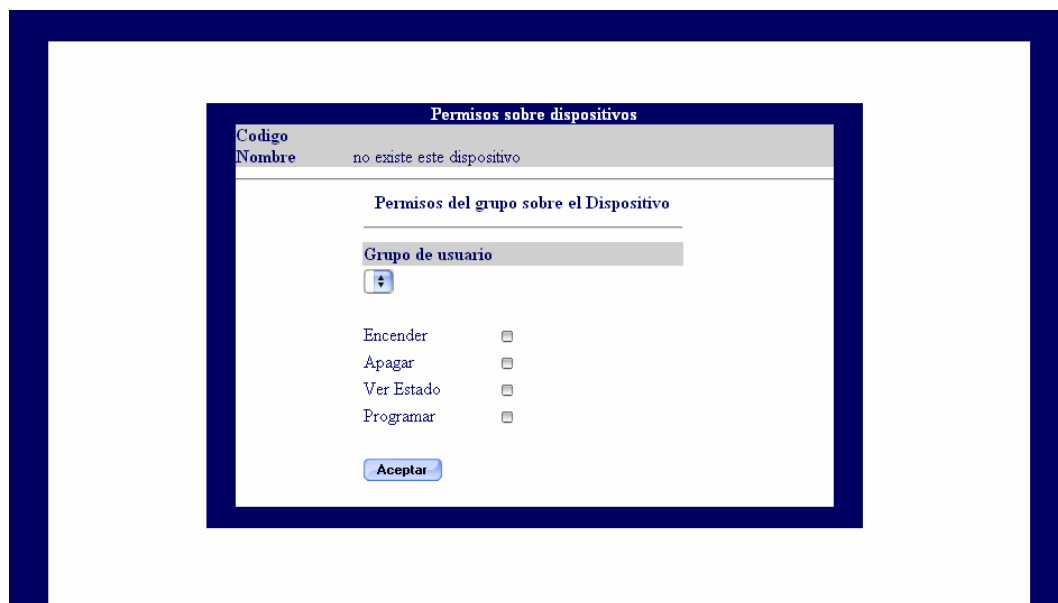
Si desea eliminar el dispositivo le mostrara la pantalla de la figura 7., en la que le piden confirmación al eliminar el dispositivo, luego eliminar y aparecerá una confirmación de que el dispositivo ha sido eliminado.



The screenshot shows a confirmation dialog box with a dark blue border. At the top, a title bar reads "Confirma que desea eliminar el dispositivo?". Below this, the device details are listed: "codigo: L-0001" and "nombre: LUZ DE LA SALA". At the bottom, there is a single button labeled "Eliminar".

Figura 7. Eliminar dispositivo

Al escoger la opción de agregar grupos a este dispositivo, va a mostrar la pantalla de la figura 8., la cual de una vez nos da la opción de asignarle los permisos al grupo de usuarios, como lo son encendido, apagado, ver el estado o programar los dispositivos.

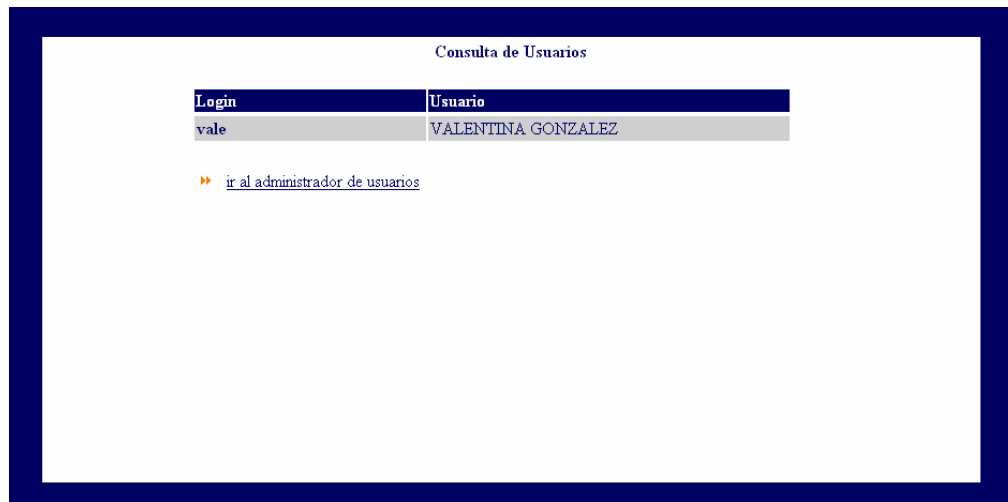


The screenshot displays a "Permisos sobre dispositivos" (Permissions over devices) screen. It features a header with "Codigo" and "Nombre" fields, where the name is "no existe este dispositivo". Below the header, a section titled "Permisos del grupo sobre el Dispositivo" contains a "Grupo de usuario" dropdown menu. Underneath, there are four permission options, each with an unchecked checkbox: "Encender", "Apagar", "Ver Estado", and "Programar". An "Aceptar" (Accept) button is located at the bottom of the screen.

Figura 8. Permisos sobre los dispositivos



Además de esto, en el detalle de los dispositivos (figura 5) también se puede ver los usuarios que están asignados al grupo y que pueden acceder a este dispositivo haciéndole clic en el código del grupo. En la figura 9 podemos ver la pantalla que nos muestra al elegirla y además podemos por medio de esta, ingresar inmediatamente a la opción de administración de usuarios. Más adelante se describe como es el proceso de administración de usuarios.



Login	Usuario
vale	VALENTINA GONZALEZ

» ir al administrador de usuarios

Figura 9. Consulta de usuarios.

Si en la administración de los dispositivos no desea ver los detalles de cada dispositivo, sino que quiere insertar un nuevo dispositivo se va a atrás, y atrás hasta que llegue a la primera página, la de consulta de dispositivos (figura 4). En esta página le da clic en insertar nuevo dispositivo y le mostrara una ventana como la de la figura 10., en la que podrá insertar el código del nuevo dispositivo, nombre, código de encendido, código de apagado, locación y luego se guarda la información. Después de hacer esto sale una confirmación de que ha insertado un nuevo dispositivo y se devuelve al menú anterior.



Insertar nuevo Dispositivo

codigo:

nombre:

cod encender:

cod apagar:

Locacion:

« volver

Figura 10. Insertar un nuevo dispositivo

## 1.2 ADMINISTRACION DE USUARIOS

En la pagina inicial (figura 3) está la opción de administración de usuarios en la que uno puede consultar los grupos de usuarios. Recuerde que no se manejan usuarios independientes sino grupos de usuarios. Al entrar en esta opción podemos ver la página de consulta de los grupos de usuarios que están creados actualmente al igual que podemos ver los detalles, insertar un nuevo grupo de usuarios, insertar un nuevo usuario o volver a la página anterior. Esto se indica en la figura 11.



Figura 11. Consulta de grupos de usuario.

Si se desea insertar un nuevo grupo de usuario, esta opción le mostrara una página como la que se muestra a continuación en la figura, en donde podrá introducir el código, nombre, locación, guardar la información introducida del nuevo grupo de usuarios, limpiar y volver a la página anterior.

Insertar nuevo Grupo de Usuario

codigo:

nombre:

locacion:

« volver

Figura 12. Insertar un nuevo Grupo de usuarios.

Si por el contrario solo se desea insertar un nuevo usuario, esta opción le mostrara una pantalla como la que aparece en la figura 13., en la que podrá insertar el nombre, la identificación de usuario, contraseña, grupo de usuario al que va a pertenecer, guardar los cambios realizados y limpiar. Luego de hacer esto le aparecerá una confirmación de que se ha agregado un nuevo usuario.



Formulario para insertar un nuevo usuario. El formulario contiene los siguientes campos:

- nombre:
- id de usuario:
- contraseña:
- confirme contraseña:
- Grupo de usuario:

En la parte inferior del formulario hay dos botones: "Guardar" y "Limpiar".

Figura 13. Insertar un nuevo usuario.

Si se desea ver los detalles de cada grupo de usuarios, hay que hacer clic en detalles o en el código de cada grupo de usuarios. Luego de esto le aparecerá una página como la de la figura 14., en la que además de los detalles se podrá insertar un nuevo usuario, editar los permisos de cada grupo de usuario o eliminar un grupo de usuario. Cualquier opción que escoja le mostrara una confirmación de lo que hará. Más abajo esta la opción de volver a la página anterior, es decir, la de la figura 11.



Detalle Grupo Usuarios

Codigo: GU-0001  
Nombre: HIJOS

Locacion: CASA ABUELITA  
Usuarios: 3  
Dispositivos: 3

Usuarios asignados

login	nombre	+	X
gpsa	gpsa	+	X
princesita	princesita	+	X
vale	VALENTINA GONZALEZ	+	X

» insertar un nuevo usuario

Dispositivos asignados

codigo	nombre
L-0001	LUZ DE LA SALA
L-0002	LUZ DE LOS NINOS
L-0003	BANO

» editar » eliminar

« volver

Figura 14. Detalles de los Grupos de usuarios.

### 1.3 ADMINISTRACION DE LOCACIONES

Si desea ir solamente a la administración de las locaciones, este le mostrará la página de la figura 15., en la que se podrá ver los detalles, insertar locaciones y volver a la página anterior, es decir, al menú principal (figura 3).



Figura 15. Administración de locaciones.

Si escoge la opción de insertar una nueva localización este le mostrara una pantalla como la de la figura 16., en la que podrá introducir el código, nombre, dirección ip, guardar la información de la nueva locación, limpiar y volver a la página anterior. Cuando introduzca la nueva localización le aparecerá una confirmación de que ha introducido una nueva localización.

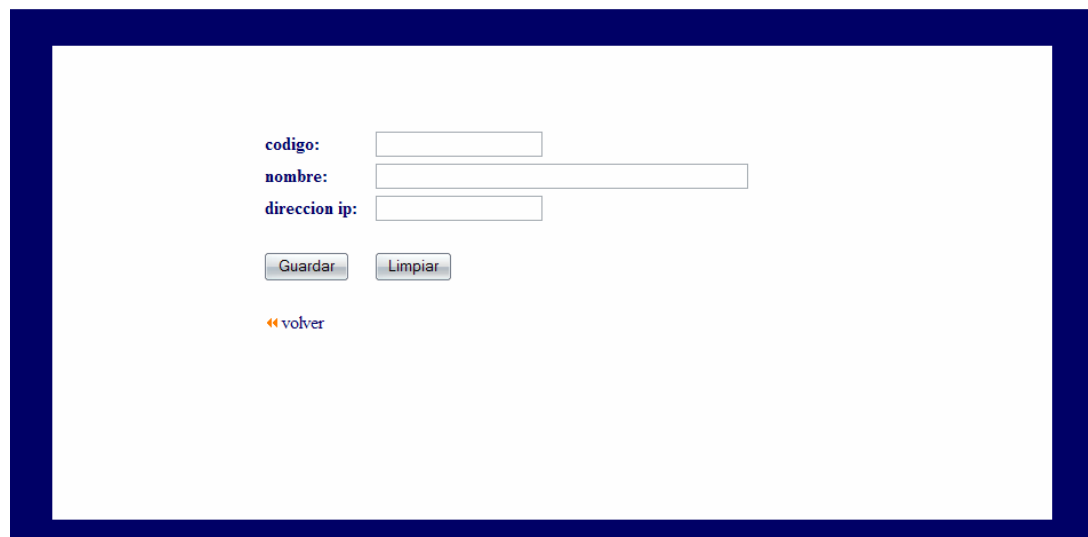


Figura 16. Insertar nueva localización.

Si desea ver los detalles haga clic en detalles o en el código de la locación que desea ver. A continuación aparecerá una pantalla como la de la figura 17., en la que aparte de ver los detalles, se podrá ver los detalles de los diferentes dispositivos y grupos de usuarios que están asignados a la locación. Además de esto podrá editar la información o eliminarla y podrá volver a la página anterior.

The screenshot shows a web application interface titled "Detalle de Locaciones". At the top right, there are two links: "editar" and "eliminar". The main content area displays the following information:

- Codigo:** 0001
- Nombre:** CASA ABUELITA
- Direccion IP:** 444
- Estado:** (empty field)
- Dispositivos:** 3
- Grupos de Usuarios:** 1

To the right of these fields is an icon of a house with a person inside.

Below this information are two tables:

**Dispositivos asignados**

codigo	nombre	det
L-0001	LUZ DE LA SALA	
L-0002	LUZ DE LOS NINOS	
L-0003	BANO	

**Grupos de usuarios asignados**

codigo	nombre	det
GU-0001	HIJOS	

At the bottom left, there is a link: "volver".

Figura 17. Detalles de las Locaciones.

Si desea ver los detalles de los dispositivos asignados a la locación le mostrara una página como la de la figura 5., que ya se describió anteriormente. Si quiere ver los detalles de los grupos de usuarios que están asignados a la locación lo podrá hacer escogiendo los detalles o el código del grupo y a continuación le mostrara la página de la figura 14., que ya fue descrita anteriormente.

## 2. MANUAL DE LA APLICACIÓN CELULAR (MIDLET)



Figura 18. Validación usuario.

Al empezar la aplicación por primera vez aparecerá en el celular una pantalla en la que se deberá introducir el código del administrador para empezar la aplicación. Luego de esto aparecerá una nueva pantalla con el mensaje de bienvenida al sistema domótico y luego se le da ingresar. En seguida aparece una pantalla en la cual se le introduce el usuario y contraseña como se indica en la figura 18. Luego se selecciona en menú la opción de login y va a mostrar que esta conectando.



Figura 19. Locaciones disponibles.

Luego de hacer esto aparecerán las locaciones a las cuales puede tener acceso. Selección la locación que le interesa y luego en menú, podrá ver los dispositivos que están asignados a esa locación y a la cual le podrá hacer los cambios. Además está la opción de volver a la pantalla anterior. La pantalla que le aparecerá es como la de la figura 19.



Figura 20. Dispositivos de la locación.



Figura 21. Detalles de dispositivos.

Si selecciona la opción de ver dispositivos le aparecerá una pantalla como la de la figura 20., en la que podrá ver los diferentes dispositivos con que cuenta la locación y sobre los cuales podrá ejecutar una acción determinada. También está la opción de volver a la pantalla anterior.

Si desea ver los detalles del dispositivo aparecerá una pantalla como la de la figura 21., en la que podrá ver en qué estado se encuentra este dispositivo, si está apagado o encendido y además indica quien fue la última persona que tuvo acceso a este dispositivo y a qué horas. Si selecciona el menú podrá escoger la acción que desea ejecutar sobre este. Cuando haya escogido la acción le aparecerá una confirmación de que ya fue ejecutada la orden.



**Figura 22. Frecuencia de la programación sobre el dispositivo.**

Si lo que desea es programar el dispositivo a una determinada hora, le mostrara una pantalla como esta, en la que podrá escoger la frecuencia con que desea hacerlo, ya sea de lunes a viernes o los sábados y domingos. Luego selecciona programar hora y aparecerá una nueva pantalla en la que podrá colocar la hora y minutos a la cual desea ejecutar esa orden.



## APENDICE C HARDWARE DOMÓTICO

El hardware domótico con el que se realizo este proyecto, se encuentra disponible en la facultad de Ingeniería de Sistemas en la Universidad el Bosque. Este hardware esta compuesto de dos tarjetas que permiten que se pueda tener la funcionalidad a escala de una casa inteligente, permitiendo que se pueda incorporar en ella, cualquier clase de dispositivo que se desee.



**Figura 1. Hardware domótico**

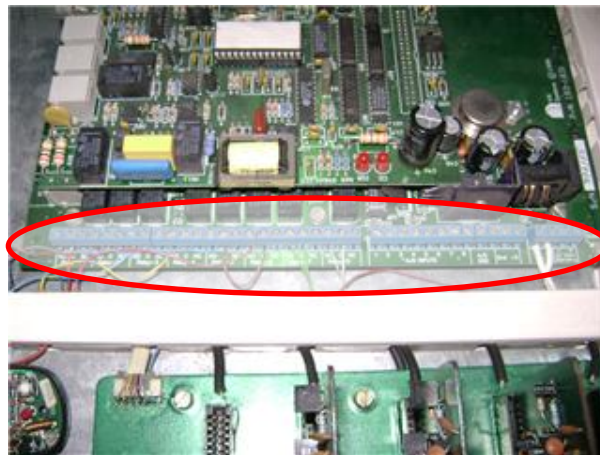
Este hardware (Figura 1) contiene la programación de los dispositivos que en este caso, son las luces, la cual se hace por medio del programa llamado StarGate el cual se explicara mas adelante.



**Figura 2. Puertos**

Para poder comunicar el hardware domótico con la computadora, hay que hacer la comunicación por medio de un cable serial, el cual va desde el puerto del computador hasta una de las entradas que se ven en la Figura 2, en la que se puede ver los puertos de COM1, COM2, COM3, así como para una conexión telefónica, entre otras. En este caso la comunicación se va a hacer por el puerto COM1.

Para simular las luces de la casa se utilizaron leds los cuales van conectados a hardware por medio de una canaleta como se muestra en la Figura 3., en donde se hace la comunicación con la tarjeta controladora y que esta encargada de recibir la programación de los leds y demás dispositivos. La canaleta se utiliza como protección para que los cables no se suelten.



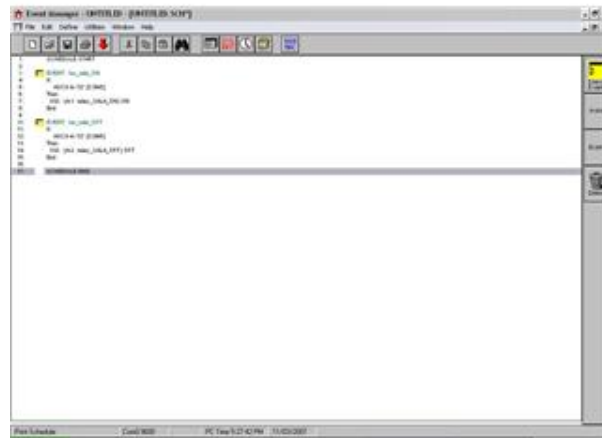
**Figura 3. Conexión de leds con la tarjeta controladora.**

Además de los dispositivos, el hardware también tiene incorporado sonido, el cual va confirmando lo que se va haciendo en el. Para que este hardware funcione tiene que estar conectado a la corriente y para que los leds funcionen hay que poner en el hardware una pila.



**Figura 4. Software StarGate**

Este software es muy fácil de manejar ya que hacer el código para este no es tan complicado. Solo basta con especificar el tipo de dispositivo al cual se le quiere hacer la programación, y luego detallar cuando se tiene que apagar o encender. Por ultimo se manda esa información por medio de un comando al hardware domótico, que lo recibe y lo ejecuta.



Para el caso de este proyecto se hace la programación de los dispositivos validando un ASCII que va a ser el código del dispositivo cuando se apaga o se enciende, para que así el sistema sepa que es lo que va a ejecutar.

124