

PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA INFORMÁTICA

Arquitectura domótica utilizando dispositivos X10 y
comunicación mediante Web Services

Claudio Silvano Jara Ferrada

INFORME FINAL DEL PROYECTO
PARA OPTAR AL TÍTULO PROFESIONAL DE
INGENIERO CIVIL EN INFORMÁTICA

Julio 2009

Pontificia Universidad Católica de Valparaíso
Facultad de Ingeniería
Escuela de Ingeniería Informática

Arquitectura domótica utilizando dispositivos X10 y
comunicación mediante Web Services

Claudio Silvano Jara Ferrada

Profesor Guía: **Claudio Cubillos Figueroa**

Profesor Co-referente: **Silvana Roncagliolo de la Horra**

Carrera: **Ingeniería Civil en Informática**

Julio 2009

Dedicatoria

“A mis queridos padres: Silvano y Mirtina, por su constante apoyo y dedicación. Por celebrar conmigo cada logro y también por enseñarme a levantarme tras cada tropiezo y mirar hacia adelante con confianza. Ustedes me enseñaron y guiaron en cada paso importante, siendo siempre un pilar fundamental en mi vida.”

Agradecimientos

“Primeramente, agradezco a Dios por regalarme la vida y llenarme de bendiciones a lo largo de ella.

A mi amada esposa Jennifer, fuente inagotable de alegrías. Su amor constante me ha alentado para dar este gran paso. Gracias por dejarme ser parte de tu vida.

A mis padres por sus sabios consejos y ánimo cuando lo necesitaba.

A mis hermanos Ginno y Elías, compañeros de juegos y amigos entrañables. Gracias por todos los momentos compartidos en casa.

A mis profesores, por ser una constante guía a lo largo de todos estos años de enseñanza.

Gracias a mis amigos y a todos quienes de una u otra manera me han acompañado durante este proceso.”

Resumen

En tiempos modernos dedicarle un tiempo a la seguridad vale la pena. Cuidar el hogar de accidentes o de intrusos no es un tema para dejar de lado. Es importante por lo tanto contar con las medidas necesarias para resguardarse de estos inconvenientes.

El presente trabajo describe una arquitectura domótica capaz de dotar al hogar de algún grado de inteligencia, al poder responder de forma satisfactoria ante eventos indeseados pero previstos. Dicha arquitectura permite generar acciones oportunas frente a situaciones de riesgo, utilizando para esto, sensores que detectan factores no deseados tanto internos (incendio, fugas de agua o gas, etc.) como externos (robos).

La arquitectura posee además herramientas dedicadas al confort y ahorro energético, dotando al hogar de dispositivos capaces de controlar distintos componentes como la luz, calefacción, persianas, etc.

Se proponen, como elemento primordial para lograr estos objetivos de seguridad y confort, los dispositivos domóticos provistos del protocolo de comunicación X10, ya que su fácil manejo, bajo costo y modularidad los hacen una herramienta potente y sencilla. Es importante destacar que el medio principal de transmisión de dichos dispositivos es el cableado eléctrico del hogar.

Para la comunicación entre distintas viviendas, que han sido afectadas por un evento o a las que se les desea notificar, se utiliza la tecnología de Web Services por su habilidad de comunicarse con distintas plataformas, lenguajes de programación y otras bondades.

Finalmente se comentan dos escenarios en los cuales es probada la arquitectura.

Abstract

Nowadays spending time in security is worth. Caring about house accidents or intruders is not a topic to leave unattended. In this way, it's important to have a good plan to keep away from these problems.

The following work describes a domotic architecture which is able to give anyone's home some degree of intelligence by being able to react successfully on undesired but expectable events. This architecture lets create timely actions in the case of risky situations by using sensors which detect unwanted factors being either inner (fires, water or gas leaks, etc.) or outer ones (burglary).

The architecture also has tools meant for comfort and energy saving, giving home devices capable of controlling different components such as light, central heating, blinds, etc.

To achieve these goals, it is proposed that domotic devices should be used as the main tool, provided with the X10 protocol because its easy handling, low cost and modularity makes them a powerful and single tool. It is important to mention that the main transmission mean of that devices is the electrical wiring.

For the communication between different dwellings which have been affected by an event of which we want to notify, the Web Services technology is used because the can communicate with different platforms, programming languages, operating systems and so on.

At the end, two scenarios are shown in which the architecture is tested.

Índice de Contenido

Resumen	5
Abstract	6
Índice de Contenido	7
Índice de Ilustraciones	9
Índice de Tablas	11
Lista de Abreviaturas	12
1 Introducción	13
1.1 Descripción del problema	14
1.2 Objetivos	14
1.3 Metodología	15
1.4 Estructura del documento	16
2 Domótica y X10	17
2.1 Estándares domóticos:	18
2.2 X10	24
2.3 Historia del X10	24
2.4 Tipos de dispositivos	25
2.5 Comunicación entre plataformas:	26
2.6 Tecnología X10 (Powerline Carrier (PLC)).....	26
2.7 Elementos básicos de los sistemas X10.....	30
2.8 Filtrado de ruidos y señales.....	33
2.9 Consideraciones de implementación	36
2.10 Mercado actual de software X10.....	38
3 Web Services	40
3.1 XML	41
3.2 WSDL: Web Service Description Language	42
3.3 UDDI: Universal Description, Discovery and Integration	43
3.4 SOAP (Simple Object Access Protocol).....	44
3.5 Arquitectura de Web Services.....	45
3.6 SOA	46
3.7 JAX-WS	47
3.8 JAXB	48
3.9 Implementación de Web Service.....	49
3.10 FrameWorks para Web Services	51
4 Arquitectura domótica	52
4.1 Aplicaciones de la arquitectura	53
4.2 Diseño de la arquitectura	56
4.3 Diseño de interfaz aplicación para Iphone.....	67
4.4 Herramientas y entorno de desarrollo.....	71

4.5	Implementación.....	71
4.6	Implantación.....	75
5	Casos de estudio.....	78
5.1	Ejecución de acciones remotas.....	78
5.2	Alarma por sensor de movimiento	84
6	Conclusiones.....	87
	Referencias	89

Índice de Ilustraciones

Figura 1: Ejemplo de áreas de gestión en un sistema doméstico	17
Figura 2: Modulo X10.....	24
Figura 3: Sincronización de la señal X10 con la corriente alterna	27
Figura 4: Forma de onda en transmisión X10.	27
Figura 5: Composición de un comando X10.....	28
Figura 6: Composición de un comando X10.....	28
Figura 7: Sincronismo entre la señal X10 y la corriente alterna	29
Figura 8: Controladores hasta 256 posibles códigos.....	30
Figura 9: Controladores manuales PHC01 y PHC02.....	31
Figura 10: Controladores PHC05	31
Figura 11: Interfaz PSC01	33
Figura 12: Dispositivos para detección de ruido. Izquierda: XPTR, derecha: XPPT.....	34
Figura 13: Dimensiones del filtro XPPF para uso residencial1"1	35
Figura 14: Filtro XPNR para reducir ruido	35
Figura 15: Principales tecnologías de Web Service.....	41
Figura 16: Estructura de UDDI	44
Figura 17: Arquitectura Web Service	46
Figura 18: Mecanismo de funcionamiento JAXB	49
Figura 19: Creación Web Service mediante NetBeans	50
Figura 20: Edición de parámetros Web Service	50
Figura 21: Modelo de despliegue de arquitectura.....	53
Figura 22: Caso de uso general.....	57
Figura 23: Caso de uso "Ejecución de acciones"	58
Figura 24: Caso de uso "Gestión de acciones"	58
Figura 25: Caso de uso "Gestión de grupos"	59
Figura 26: Caso de uso "Gestionar informes"	60
Figura 27: Modelo de datos.....	61
Figura 28: Modelo de base de datos de una estación.....	62
Figura 29: Modelo de base de datos del sgc.....	63
Figura 30: D. Secuencia - Envío de alarma.....	63
Figura 31: D. Secuencia - Ingreso de estación	64
Figura 32: D. de colaboración - Evento sensor	64
Figura 33: Diagrama de estado "Estación en espera de evento (sensor)"	65
Figura 34: Diagrama de estado "Estación ejecutando acción"	66
Figura 35: Diagrama de actividad "Agregar Estación"	66
Figura 36: Diagrama de actividad "Eventos SGC"	67
Figura 37: Sistema de Logueo	68

Figura 38: Menú Administrador	Figura 39: Menú Usuario	68
Figura 40: Formulario de ingreso de nuevo dispositivo		69
Figura 41: Nuevo actuador ingresado (código A3)		69
Figura 42: Opciones de prueba de actuador		70
Figura 43: Lámpara antes de encenderla remotamente		70
Figura 44: Lámpara encendida remotamente		70
Figura 45: Como recibir un SMS desde un sitio Web		74
Figura 46: Capas del API		76
Figura 47: Lectura/Escritura		76
Figura 48: Diagrama despliegue "Escenario de prueba 1a"		79
Figura 49: Diagrama de secuencia "Escenario de prueba 1a"		81
Figura 50: Diagrama de despliegue "Escenario de prueba 1b"		82
Figura 51: Diagrama de secuencia "Escenario de prueba 1b"		83
Figura 52: Diagrama de despliegue "Escenario 2"		85
Figura 53: Diagrama Secuencia "Ejecutar acción"		86

Índice de Tablas

Tabla 1- Plan de trabajo	16
Tabla 2: Códigos X10 en bits. (House Codes y Key Codes)	29
Tabla 3: Caso de uso "Operacion sobre actuador mediante Web"	80
Tabla 4: Caso de uso "Operacion sobre actuador mediante SMS"	83
Tabla 5: Caso de uso "Ejecutar acción"	86

Lista de Abreviaturas

- WSO: Web Service Oriented.
- API: Application programming interface.
- IDE: Integrated development environment.
- UML: Unified Modeling Language.
- CORBA: Common Object Request Broker Architecture.
- RMI: Java Remote Method Invocation.
- SOAP: Simple Object Access Protocol.
- WSDL: Web Services Description Language.
- UDDI: Universal Description, Discovery and Integration.
- XML: Extensible Markup Language.
- HTTP: Hypertext Transfer Protocol.
- WS: Web Services

1 Introducción

La seguridad y el confort son aspectos muy importantes en el ser humano. El tener un hogar protegido, optimizando en recursos con una buena gestión siempre ha estado en su mira, y conforme crece la tecnología, crecen las posibilidades de lograrlo de una manera eficiente.

Se desea obtener una ganancia en principalmente cuatro aspectos: seguridad, confort, ahorro energético y comunicación. Una de las iniciativas para lograr esta meta, es el concepto de “Hogar inteligente”. Su origen se remonta a los años 1976 y 1978 cuando se crea el protocolo de comunicación X10 en escocia.

En este trabajo se desarrolla una arquitectura domótica que permita lograr una interrelación y gestión en los 4 conceptos generales de domótica anteriormente para lo cual se construyo un sistema que es capaz de generar alarmas ante eventos indeseados y obtener una respuesta que logren satisfacer las necesidades de confort y seguridad.

Una buena alternativa y de bajo costo para lograr un hogar inteligente es utilizar los dispositivos X10 [03], que poseen funciones básicas pero muy poderosas gestionan de una manera adecuada. Los dispositivos X10 utilizan como principal medio de transmisión la red eléctrica, por lo que se puede utilizar inmediatamente una vez adquiridas sin necesidad se crear cableado extra. Estos dispositivos son estandartes por lo que existen muchos proveedores que los construyen creando cada vez más aparatos que pueden irse mezclando y extendiéndose sin problemas, a la vez son modulares y pueden ser agregados o desligados en una cierta configuración sin tener que realizar grandes modificaciones.

Por otro lado tenemos que un “Web Service” es “una aplicación de software identificado por un URI, cuyas interfaces y Vinculantes son capaces de ser definidos, descritos y descubiertos por artefactos XML y apoya directamente la interacción con otras aplicaciones de software usando mensajes XML a través de protocolos estándares basados en Internet” [06] y proporcionan la posibilidad de integrar dinámicamente componentes, que puedan comunicarse e interactuar con otros componentes que hayan publicado sus servicios e interactuar con diversos clientes móviles.

La arquitectura que se presenta en este trabajo utiliza en gran medida dispositivos X10 (basado en la red eléctrica), que poseen un protocolo de comunicación, también denominado X10, para la interacción entre para sensores, actuadores y el pc, y la tecnología de Web Services, para publicar los dispositivos sensoriales instalados en una vivienda y lograr la interacción con otros dispositivos que gatillarán a diversos actuadores en respuesta a eventos detectados.

1.1 Descripción del problema

El problema radica en la imposibilidad de interactuar con una vivienda remotamente distante, operar elementos de una vivienda con el fin de ahorro de energía y confortabilidad y prestar seguridad.

Una de las posibilidades de contrarrestar la imposibilidad de prestar seguridad a un lugar en el que no se está presente es, mediante dispositivos sensoriales que hagan las veces de mecanismos receptores de eventos indeseados tanto internos en una vivienda como externos como los robos. En el ámbito de la seguridad, se propone como método de seguridad, que cada vivienda pertenezca a un grupo de viviendas, y en la medida que ocurra algún evento indeseado, se le envíe un mensaje a cada vivienda del grupo con la información relevante para cada una tome una acción física (prender una alarma, las luces exteriores de su casa) o un envío de mensajes (email y/o mensaje de texto).

En el presente trabajo se diseña y construye una arquitectura domótica que cuenta con los dispositivos físicos que son capaces de detectar los eventos indeseados y actuadores capaces de reaccionar ante un estímulo definido con tal de lograr un método de disuasión de un agente extraño o una aminoración de los daños. También cuenta con métodos para dar aviso de los eventos detectados y una tecnología de comunicación basada en estándares de Internet para interactuar con las otras viviendas del grupo. Posee además mecanismos para la interacción con los dispositivos y electrodomésticos de una vivienda para lograr un grado extra de comodidad (encender calefacción, activar sistema de riego), y lograr un ahorro energético (apagar luces cuando estas hallan quedado encendidas, cerrar persianas a determinada hora, etc.)

1.2 Objetivos

En esta sección se presentan los objetivos del proyecto.

1.1.1 Objetivo general

Desarrollar una arquitectura domótica utilizando dispositivos X10, sensores genéricos y cámaras Web, que pueda ser utilizada en el contexto de seguridad y confort para un set de viviendas cuya comunicación sea realizada a través de Web Services.

1.2.1 Objetivos específicos

Para el cumplimiento del objetivo General del proyecto, se deben lograr los siguientes objetivos específicos:

- Estudio de las diferentes tecnologías asociadas al uso de los dispositivos X10 y Web Services
- Desarrollar una aplicación Web y una aplicación de escritorio capaz de gestionar el envío de alarmas, reportes y administración de las configuraciones del sistema.
- Validar la aplicación en dos escenarios utilizando actuadores X10, sensores de movimientos y cámaras web en dos PCs remotos.

1.3 Metodología

A continuación se presenta la metodología con la que se hará el desarrollo de éste proyecto

1.3.1 Metodología propuesta y plan de trabajo

- Estudio teórico
- Generación de una arquitectura de solución
- Estudio de Herramientas para el desarrollo de las aplicaciones
- Desarrollo de escenarios
- Desarrollo de interfaz para Iphone/Ipod touch
- Configuración y Ejecución de pruebas experimentales

Para la consecución de los objetivos anteriormente planteados, se consideran a continuación cuatro etapas necesarias para la efectiva realización del proyecto:

1. Recolección de información y posterior análisis de la investigación: Estudio en profundidad del protocolo X10, implementaciones existentes, herramientas disponibles para la interacción entre SW y la Red eléctrica (protocolos, IDEs). Investigación de los distintos dispositivos tanto de sensores como de actuadores. Estudio de los “Web Services”, protocolos y metodologías de aplicación. Estudios de arquitecturas descentralizadas para el tratamiento de las interacciones.
2. Construcción y desarrollo de prototipos modulares: Una vez que se halla recolectado la información suficiente, se realizaran pequeños desarrollos incrementales y modulares, para probar la funcionalidad tanto de comunicación entres “Web Services”, comunicaron entre el SW en cuestión y los distintos dispositivos X10
3. Implementación final e integración: Integración final de pequeños módulos, logrando una comunicación fluida y fiable mediante protocolos y metodologías establecidas previamente. Se aplicaran medidas para lograr un SW robusto y escalable, con el fin de lograr que puedan anexarse nuevos dispositivos a la aplicación sin malograrla en ninguna medida sino que pueda verse beneficiada de dicha agregación.

4. Validación y pruebas de la aplicación: Con el sistema final completo se procederá a la ejecución de pruebas exhaustivas al sistema, y una validación con los implementos antes descritos para tal hecho.

La siguiente tabla muestra el plan de trabajo para la realización del proyecto.

Estudio de las tecnologías X10
Estudio de Web Services
Estudio de frameworks para Iphone /Ipod Touch
Construcción del modelo
Prueba y selección de herramientas para el Desarrollo
Pruebas de interacción entre dispositivos X10 y WS
Desarrollo y validación del primer incremento
Redefinición del modelo
Segundo incremento, buscar mejoras
Pruebas del incremento, analizar mejoras
Implementación Final del sistema
Pruebas integrales y validación

Tabla 1- Plan de trabajo

1.4 Estructura del documento

El documento se encuentra dividido en 4 secciones. La primera parte comprende sólo al capítulo 1 donde se hace una introducción al tema del proyecto, se especifica la problemática, los objetivos y la metodología a usar para la obtención final de la arquitectura domótica.

La segunda parte comprende los capítulos 2 y 3 y corresponden al estado del arte en cuanto domótica, dispositivos y protocolo X10 con una definición exhaustiva sobre los componentes que forman parte del set herramientas basados en el protocolo X10. También se interioriza en el tema de los Servicios Web (Web Services), utilizados como principal herramienta de comunicación para la arquitectura diseñada.

En la tercera parte se encuentra el diseño de la arquitectura domótica, se identifican las partes que la componen, los servicios ofrecidos, sensores utilizados y prestaciones posibles. Para ejemplificar el uso de la arquitectura se detallan dos escenarios de estudio utilizando un set de herramientas X10 y una cámara Web como un tipo de sensor de movimiento

La cuarta parte y final detalla las conclusiones obtenidas de la investigación y ejecución de varios escenarios utilizando la arquitectura, haciendo énfasis en los usos en cuanto a seguridad y confort.

2 Domótica y X10

El concepto domótica se refiere a la automatización y control de una vivienda mediante la incorporación de equipamiento especializado que sea capaz de lograr una comunicación e integración entre múltiples componentes existentes en el hogar y su interacción con el ser humano. Las tendencias en domótica apuntan hacia un mayor grado de seguridad en el hogar, ahorros de energía y confort. Estos sistemas pueden estar integrados por redes interiores y/o redes exteriores de comunicación, pudiendo ser estas cableadas o inalámbricas

Se podría por lo tanto, dividir la domótica en cuatro áreas fundamentales:

- Energía, confort, seguridad y comunicación.

Los beneficios otorgados por la domótica son variados. Algunos de los más importantes son:

- El ahorro energético gracias a una gestión "inteligente" de los sistemas y consumos.
- La potenciación y enriquecimiento de la propia red de comunicaciones.
- La más contundente seguridad personal y patrimonial.
- La tele asistencia.
- La gestión remota (teléfono, radio, Internet, etc.) de instalaciones y equipos domésticos.

Como consecuencia de todos los anteriores apartados se consigue un nivel de confort muy superior. La calidad de vida aumenta considerablemente, como se ve en la Figura 1.

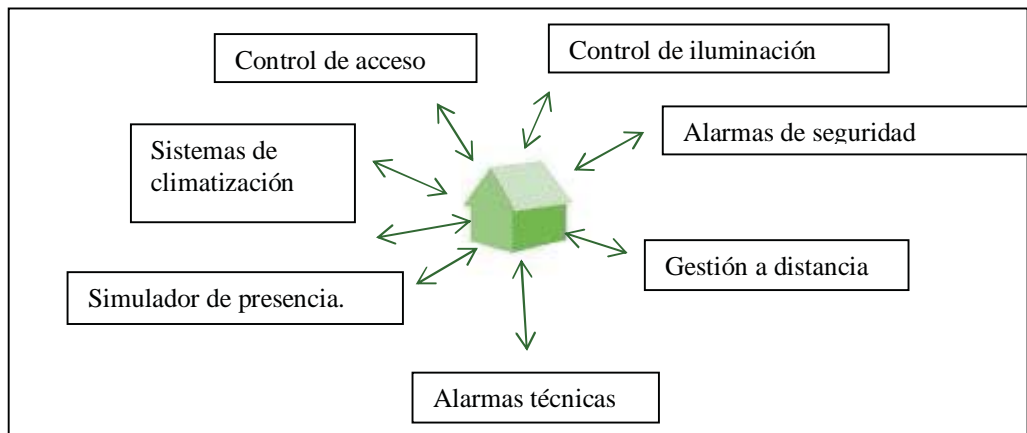


Figura 1: Ejemplo de áreas de gestión en un sistema domótico

Otro concepto que encierra la domótica es la INMOTICA. Los sistemas inmóticos se basan en la integración de elementos tecnológicos en edificios, y persiguen la automatización y control de las diversas funciones de los mismos destacando las ventajas en seguridad y gestión energética que aportan

2.1 Estándares domóticos:

A continuación se clasifican algunos de los estándares más importantes en los sistemas domóticos, separados por donde su uso es más fuerte.

2.1.1 Europeos:

a) EIB

El European Installation Bus o EIB es un sistema domótico desarrollado bajo los auspicios de la Unión Europea con el objetivo de contrarrestar las importaciones de productos similares que se estaban produciendo desde el mercado japonés y el norteamericano donde ésta tecnologías se han desarrollado antes que en Europa.

El objetivo era crear un estándar europeo, con el suficiente número de fabricantes, instaladores y usuarios, que permita comunicarse a todos los dispositivos de una instalación eléctrica como: contadores, equipos de climatización, de custodia y seguridad, de gestión energética y los electrodomésticos.

El EIB está basado en la estructura de niveles OSI y tiene una arquitectura descentralizada. Este estándar europeo define una relación extremo-a- extremo entre dispositivos que permite distribuir la inteligencia entre los sensores y los actuadores instalados en la vivienda.

I. Nivel Físico

Aunque en un principio sólo se contempló usar un cable de dos hilos como soporte físico de las comunicaciones, se pretendía que el nivel EIB.MAC (Médium Access Control) pudiera funcionar sobre los siguientes medios físicos:

- EIB.TP: Sobre par trenzado a 9600 bps. Además por estos dos hilos se suministra 24 Vdc para la tele alimentación de los dispositivos EIB. Usa la técnica CSMA con arbitraje positivo del bus que evita las colisiones evitando así los reintentos y maximizando el ancho de banda disponible.
- EIB.PL: Corrientes portadoras sobre 230 Vac/50 Hz (powerline) a 1200/2400 bps. Usa la modulación SFSK (Spread Frequency Shift Keying) similar a la FSK pero con las portadoras más separadas. La distancia máxima que se puede lograr sin repetidor es de 600 metros.

- EIB.net: Usa el estándar Ethernet a 10 Mbps (IEC 802-2). Sirve de backbone entre segmentos EIB además de permitir la transferencia de telegramas EIB a través del protocolo IP a viviendas o edificios remotos.
- EIB.RF: Radiofrecuencia: usando varias portadoras, se consiguen distancias de hasta 300 metros en campo abierto. Para mayores distancias o edificios con múltiples estancias se pueden usar repetidores.
- EIB.IR: Infrarrojo: Para el uso con mandos a distancia en salas o salones donde se pretenda controlar los dispositivos EIB instalados.
- En la práctica, sólo el par trenzado ha conseguido una implantación masiva mientras que los demás apenas han conseguido una presencia testimonial.

EIBA: La EIBA es una asociación de 113 empresas europeas, líderes en el mercado eléctrico, que se unieron en 1990 para impulsar el uso e implantación del sistema domótico EIB.

La EIBA tiene su sede en Bruselas y todos sus miembros cubren el 80 % de la demanda de equipamiento eléctrico en Europa. Las tareas principales de esta asociación son:

- Fijar las directrices técnicas para el sistema y los productos EIB, así como establecer los procedimientos de ensayo y certificación de calidad.
- Distribuye el conocimiento y las experiencias de las empresas que trabajan sobre el EIB. Encarga a laboratorios de ensayo las pertinentes pruebas de calidad.
- Concede a los productos EIB y a los fabricantes de estos una licencia de marca EIB con la que se podrán distribuir los productos.
- Colabora activamente con otros organismos europeos o internacionales en todas las fases de la normalización y adapta el sistema EIB a las normas vigentes.
- Lidera el proceso de convergencia (ver Konnex Association en este mismo apartado) de los tres buses europeos de más amplia difusión como son el propio EIB, el Batibus y el EHS (European Home System).

Según la EIBA (EIB Association) hay unos 10 millones de dispositivos EIB instalados por todo el mundo, unas 70.000 instalaciones, una gama de 4.500 productos diferentes, 113 empresas asociadas a la EIBA, y 70.000 instaladores cualificados.

b) BatiBUS

Este protocolo de domótica está totalmente abierto, esto es, al contrario de los que sucede con el protocolo LonTalk de la tecnología Lonworks, el protocolo del BatiBUS lo puede implementar cualquier empresa interesada en introducirlo en su cartera de productos.

A nivel de acceso, este protocolo usa la técnica CSMA-CA, (Carrier Sense Multiple Access with Collision Avoidance) similar a Ethernet pero con resolución positiva de las colisiones. Esto es, si dos dispositivos intentan acceder al mismo tiempo al bus ambos detectan que se está produciendo una colisión, pero sólo el que tiene más prioridad continua transmitiendo el otro deja de poner señal en el bus. Esta técnica es muy similar a la usada en el bus europeo EIB y también el en bus del sector del automóvil llamado CAN (Controller Area Network).

La filosofía es que todos los dispositivos BatiBUS escuchen lo que han enviado cualquier otro, todos procesan la información recibida, pero sólo aquellos que hayan sido programados para ello, filtrarán la trama y la subirán a la aplicación empotrada en cada dispositivo.

Al igual que los dispositivos X-10, todos los dispositivos BatiBUS disponen de unos micro-interruptores circulares o miniteclados que permiten asignar una dirección física y lógica que identifican unívocamente a cada dispositivo conectado al bus.

- **Tecnología:** La velocidad binaria es única (4800 bps) la cual es mas que suficiente para la mayoría de las aplicaciones de control distribuido.
- **Instalación:** La instalación de este cable se puede hacer en diversas topologías: bus, estrella, anillo, árbol o cualquier combinación de estas. Lo único que hay que respetar es no asignar direcciones idénticas a dos dispositivos de la misma instalación.
- **Estandarización:** BatiBUS ha conseguido la certificación como estándar europeo CENELEC. Existen una serie de procedimientos y especificaciones que sirven para homologar cualquier producto que use esta tecnología como compatible con el resto de productos que cumplen este estándar. A su vez, la propia asociación BCI ha creado un conjunto de herramientas para facilitar el desarrollo de productos que cumplan esta especificación.

c) EHS

El estándar EHS (European Home System) ha sido otro de los intentos que la industria europea (año 1984), auspiciada por la Comisión Europea, de crear una tecnología que permitiera la implantación de la domótica en el mercado residencial de forma masiva. El resultado fue la especificación del EHS en el año 1992. Esta basada en una topología de niveles OSI (Open Standard Interconnection), y se especifican los niveles: físico, de enlace de datos, de red y de aplicación. Desde su inicio han estado involucrados los fabricantes europeos más importantes de electrodomésticos de línea marrón y blanca, las empresas

eléctricas, las operadoras de telecomunicaciones y los fabricantes de equipamiento eléctrico. La idea... crear un protocolo abierto que permitiera cubrir las necesidades de interconexión de los productos de todos estos fabricantes y proveedores de servicios.

Tal y como fue pensado, el objetivo de la EHS es cubrir las necesidades de automatización de la mayoría de las viviendas europeas cuyos propietarios que no se pueden permitir el lujo de usar sistemas más potentes pero también más caros (como Lonworks, EIB o Batibus) debido a la mano de obra especializada que exige su instalación.

El EHS viene a cubrir, por prestaciones y objetivos, la parcela que tienen el CEBus norteamericano y el HBS japonés y rebasa las prestaciones del X-10 que tanta difusión ha conseguido en EEUU.

I. Nivel Físico

Durante los años 1992 al 1995 la EHS auspició el desarrollo de componentes electrónicos que implementaran la primera especificación. Como resultado nació un circuito integrado de ST-Microelectronics (ST7537HS1) que permitía transmitir datos por una canal serie asíncrono a través de las líneas de baja tensión de las viviendas (ondas portadoras o "powerline communications"). Esta tecnología, basada en modulación FSK, consigue velocidades de hasta 2400 bps y además también puede utilizar cables de pares trenzados como soporte de la señal.

En la actualidad, se están usando o se están desarrollando los siguientes medios físicos:

- PL-2400: Ondas Portadoras a 2400 bps.
- TP0: Par Trenzado a 4800 bps (idéntico a nivel físico del BatiBUS).
- TP1: Par Trenzado/Coaxial a 9600 bps.
- TP2: Par Trenzado a 64 Kbps.
- IR-1200: Infrarrojo a 1200 bps.
- RF-1100: Radiofrecuencia a 1100 bps.

II. Protocolo

Este protocolo está totalmente abierto, esto es, cualquier fabricante asociado a la EHS puede desarrollar sus propios productos y dispositivos que implementen el EHS.

Con una filosofía Plug&Play, se pretende aportar las siguientes ventajas a los usuarios finales:

- Compatibilidad total entre dispositivos EHS.
- Configuración automática de los dispositivos, movilidad de los mismos (poder conectarlo en diferentes emplazamientos) y ampliación sencilla de las instalaciones.
- Compartir un mismo medio físico entre diferentes aplicaciones sin interferirse entre ellas.

Cada dispositivo EHS tiene asociada una subdirección única dentro del mismo segmento de red que además de identificar unívocamente a un nodo también lleva asociada información para el enrutado de los telegramas por diferentes segmentos de red EHS.

EHSA: La asociación EHSA (EHS Association) es la encargada de emprender y llevar a cabo diversas iniciativas para aumentar el uso de esta tecnología en las viviendas europeas. Además se ocupa de la evolución y mejora tecnológica del EHS y de asegurar la compatibilidad total entre fabricantes de productos con interfase EHS.

d) Konnex

En abril de 1999 nueve compañías europeas establecieron una nueva asociación industrial, Konnex (KNX), para trabajar en el desarrollo de un nuevo estándar resultante de la convergencia de otros tres: Batibus, EIB y EHS.

El estándar KNX se basa en la tecnología EIB, y expande su funcionalidad añadiendo nuevos medios físicos a dicho estándar y los modos de configuración de BatiBUS y EHS.

Los objetivos de esta iniciativa, con el nombre de "Convergencia", son:

- Crear un único estándar para la domótica e inmótica que cubra todas las necesidades y requisitos de las instalaciones profesionales y residenciales de ámbito europeo.
- Aumentar la presencia de estos buses domóticos en áreas como la climatización o HVAC.
- Mejorar las prestaciones de los diversos medios físicos de comunicación sobretodo en la tecnología de radiofrecuencia.
- Introducir nuevos modos de funcionamiento que permitan aplicar una filosofía Plug&Play a muchos de dispositivos típicos de una vivienda.
- Contactar con empresas proveedoras de servicios como las telecomunicaciones y las eléctricas con el objeto de potenciar las instalaciones de telegestión técnica de las viviendas o domótica.

Aunque puede utilizar distintos medios físicos; pares trenzado, línea eléctrica, cableado Ethernet o radio-frecuencia, lo más habitual es que las instalaciones KNX utilicen cableado propio de par trenzado.

La versión 1.0 del estándar KNX proporciona una solución con tres modos de configuración:

- Modo-S (modo sistema). La configuración del sistema usa la misma filosofía que el EIB actual, esto es, los diversos dispositivos o nodos de la red son instalados y

configurados por profesionales con ayuda de una aplicación software especialmente diseñada para este propósito.

- **Modo-E (Modo Easy).** En la configuración sencilla los dispositivos son programados en fábrica para realizar una función concreta. Aún así algunos detalles deben ser configurados en la instalación, ya sea con el uso de un controlador central (como una pasarela residencial o similar) o mediante unos micro interruptores alojados en el mismo dispositivo (similar a muchos dispositivos X-10 que hay en el mercado).
- **Modo-A (Modo Automático).** En la configuración automática, con una filosofía Plug&Play ni el instalador ni el usuario final tienen que configurar el dispositivo. Este modo está especialmente indicado para ser usado en electrodomésticos, equipos de entretenimiento (consolas, set-top boxes, HIFI,...) y proveedores de servicios. Es el objetivo al que tienden muchos productos informáticos y de uso cotidiano. Con la filosofía Plug&Play, el usuario final no tiene que preocuparse de leer complicados manuales de instalación o perderse en un mar de referencias o especificaciones.

2.1.2 Americano:

a) CEBus

En 1984 varios miembros de la EIA norteamericana (Electronics Industry Association) llegaron a la conclusión de la necesidad de un bus domótico que aportara más funciones que las que aportaban sistemas de aquella época (ON, OFF, DIMMER xx, ALL OFF, etc.). Especificaron y desarrollaron un estándar llamado CEBus (Consumer Electronic Bus).

En 1992 fue presentada la primera especificación. Se trata de un protocolo, para entornos distribuidos de control, que está definido en un conjunto de documentos (en total unas 1000 páginas). Como es una especificación abierta cualquier empresa puede conseguir estos documentos y fabricar productos que implementen este estándar.

En Europa una iniciativa similar en prestaciones, y en el mercado al que va dirigido, es el protocolo EHS (European Home System).

b) LonWorks

LonWorks es una tecnología de control domótico propietaria de la compañía americana Echelon Corp. (<http://www.echelon.com>).

Al igual que KNX, LonWorks puede utilizar una gran variedad de medios de transmisión: aire, par trenzado, coaxial, fibra, o red eléctrica. Requiere la instalación de “nodos” a lo largo de la red que gestionan los distintos sensores y actuadores. La instalación y configuración de estos nodos debe ser realizada por profesionales utilizando las herramientas informáticas apropiadas.

LonWorks es una tecnología muy robusta y fiable por lo que está especialmente indicada para la automatización industrial, ámbito del que procede.

2.2 X10

X10 es un protocolo de comunicación que funciona a través de la red eléctrica de la vivienda. Mediante este protocolo se pueden interconectar dispositivos compatibles X10 y hacer que transmitan información por los cables eléctricos, creando una red de dispositivos. Los dispositivos X10 son modulares por lo que pueden extraerse o insertarse en el sistema con una mínima configuración adicional.

La configuración de un sistema X-10 es sencilla pues basta con asignar a cada uno de los dispositivos un código de vivienda (A-P) y un código de unidad (1-16), con lo que se posibilita un total de 256 combinaciones distintas. Estos códigos se seleccionan de forma manual en cada dispositivo. Ver ejemplo de un modulo en Figura 2.

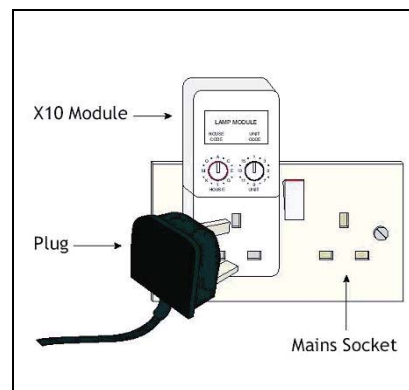


Figura 2: Modulo X10

Originalmente X10 se desarrollo para una comunicación unidireccional, pero luego también se añadió la capacidad de tener una comunicación bidireccional. De todas maneras la gran mayoría de las comunicaciones son unidireccionales.

2.3 Historia del X10

En 1970 un grupo de ingenieros inició una compañía llamada Pico Electronics en Glenrothes, Escocia. Pico revolucionó la industria de las calculadoras desarrollando el primer chip que funcionaba solo. (La mayoría de las calculadoras de la época usaban al menos cinco, éstos eran los conocidos circuitos integrados, ICs) [02].

En 1974 los ingenieros de Pico se unieron al desarrollo de un cambiador de registros (record changer) que seleccionaba pistas de un vinilo de LP regular, éste fue denominado Accutrac y podía ser manejado a distancia con un control remoto basado en un dispositivo desarrollado por Pico usando señales ultrasónicas. Esto condujo directamente a la idea de controlar remotamente luces y aparatos. En 1975 el proyecto X10 fue concebido. (Correspondía al décimo proyecto en que Pico trabajaba. Existían 8 diferentes proyectos de

calculadoras IC y el Accutrac fue el proyecto X-9). El concepto de usar el cableado de corriente alterna existente para transmitir señales para controlar luces y aparatos nació.

En 1978, luego de varios años refinando la tecnología, los productos X10 comenzaron a aparecer en las tiendas de Radio Shack. Rápidamente aparecieron en las tiendas de Sears. Se formó una sociedad con BSR, conocida como X10 Ltd, y el sistema X10 BSR nació. Aquel sistema consistía de una consola de comando de 16 canales, un módulo Lámpara, y un módulo Aparato. Pronto apareció el módulo Interruptor de pared y el primer temporizador X10.

En 1989, X10 introdujo el primer sistema de seguridad inalámbrico de bajo costo y de fácil instalación. Que contenía un discador por voz, sistema de monitoreo, entre otras características. En 1995, X10 configuró su propia estación de monitoreo llamado Orca en Seattle, Washington. Hoy este sistema monitorea los sistemas de seguridad desarrollados y manufacturados por X10 para Radio Shack, Phillips Consumer Electronics, (Magnavox) y el nuevo X10 Powerhouse.

2.4 Tipos de dispositivos

A continuación se muestran los tipos de dispositivos X10:

- **Transmisores:** Estos transmisores envían una señal especialmente codificada de bajo voltaje que es superpuesta sobre el voltaje del cableado. Un transmisor es capaz de enviar información hasta a 256 dispositivos sobre el cableado eléctrico. Múltiples transmisores pueden enviar señales al mismo módulo.
- **Receptores:** Estos dispositivos toman la seña enviada por los dispositivos transmisores. Una vez que la señal es recibida el dispositivo responde encendiéndose (ON) o apagándose (OFF). Los receptores generalmente tienen un código establecido por el usuario para indicar la dirección del dispositivo. Múltiples dispositivos con el mismo código pueden co-existir y responder al mismo tiempo dentro de una misma casa.
- **Bidireccionales:** Como los receptores y transmisores, pueden comunicarse con 256 direcciones distintas. Cuando se usan con algunos controladores de computadoras, estos dispositivos pueden reportar su estado.

2.5 Comunicación entre plataformas:

X-10 permite la comunicaron entre 3 distintas plataformas [04], a saber:

- **PLC (Transmisión por red eléctrica):** Transmisión de datos mediante el cableado existente de red eléctrica dentro del hogar u oficina. Se envían señales a los controladores conectados a la carga a controlar.
- **RF (Radio Frecuencia):** Mucho más flexible ya que es inalámbrico y no requiere una línea vista para la conexión con los dispositivos.
- **Infrarrojo:** Debido a que es direccional y requiere “línea vista”, el más ineficiente, pero en ambientes donde existe mucha interferencia por RF, se considera apropiado y muy útil.

Si bien existe flexibilidad a la hora de escoger los dispositivos X10, es importante detenerse a pensar cual de estos dispositivos será el mas indicado dependiendo de la función, el lugar dentro de la red eléctrica, carga de RF, tipos de cargas conectadas dentro de la rama (cargas inductivas o capacitivas).

Para el control automatizado de estos dispositivos interconectados existen soluciones como el dispositivo HAL2000 o la inclusión de controladores para computadores que harán la gestión de la automatización de procesos y procedimientos entre los distintos elementos.

Es necesario, para el correcto funcionamiento de la tecnología X10, escoger bien la plataforma que se va a utilizar ya que en el caso de la comunicación entre dispositivos emplazados en la red eléctrica se debe tomar en cuenta algunos problemas como el ruido que afecta a la red.

2.6 Tecnología X10 (Powerline Carrier (PLC))

El protocolo X10 provee comunicación entre emisores y receptores a través del cableado eléctrico del hogar o edificio, mediante ráfagas de Radio frecuencia de 120 kHz. Las transmisiones con sincronizadas al inicio de la onda de la corriente alterna (de 50Hz a 60Hz). El objetivo es transmitir lo mas cerca del punto de cruce cero, permitiendo una variación máxima de 200 microsegundos (Figura 3)

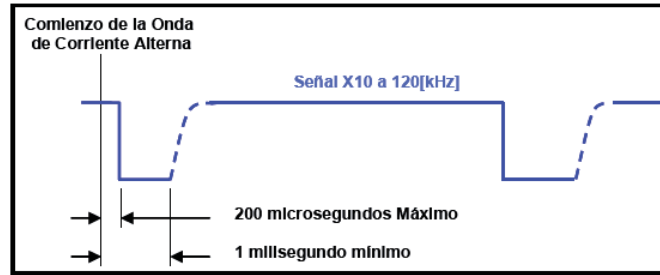


Figura 3: Sincronización de la señal X10 con la corriente alterna

Un 1 binario se representa mediante una ráfaga a 1 ms de 120 kHz. Esta ráfaga debe inyectarse en el punto cero de cruce de un medio ciclo de la corriente eléctrica. La ausencia de esta ráfaga de 120kHz representa el 0 binario. Para compatibilidad con sistemas trifásicos el 1 binario se transmite 3 veces, como 3 ráfagas de 120kHz para coincidir con el punto cero de las tres fases de la corriente. [01]

En la Figura 4 se muestra la relación sincrónica de las 3 ráfagas para representar el 1 binario.

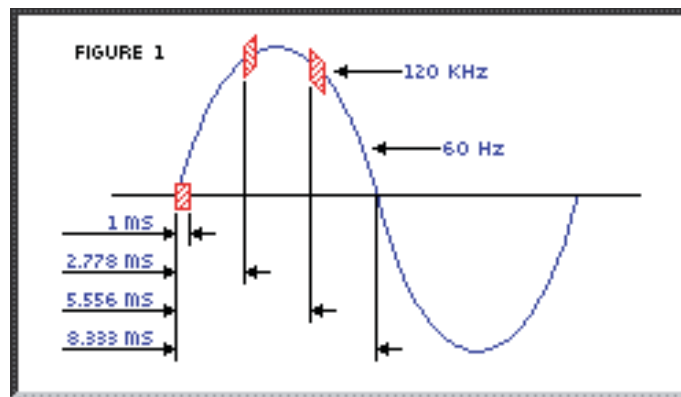


Figura 4: Forma de onda en transmisión X10.

Para construir una red de dispositivos X10, a cada dispositivo se le asigna una identificación consistente de un código de 9 bits, donde los primeros 4 bits corresponden al código de casa (House Code) y los otros 5 bits al código de dispositivo (Number Code).

Para enviar una señal hacia un dispositivo se utiliza el siguiente formato:

Start Code – House Code – Key Code

El Key Code corresponde al Number Code o Function Code, es decir un identificador de dispositivo o un código de función aplicable a uno o más dispositivos.

El código de inicio (Start Code) siempre es el binario 1110, y puesto que cada dígito binario ocupa medio ciclo de la corriente eléctrica, se requieren 2 ciclos completos para enviar el Start Code.

Los bits del House Code y el Key Code se transmiten en forma de complemento real en ciclos alternos de la corriente. Es decir, para enviar el 1 se envía un 1 en el primer medio ciclo y un 0 en el segundo medio ciclo; para enviar el 0 se envía un 0 en el primer medio ciclo y un 1 en el segundo medio ciclo.

Puesto que el código de casa consta de 4 bits, y al transmitirlo se usa el complemento, entonces en realidad se envían 8 bits, cada uno ocupando 1 medio ciclo de la corriente, en total 4 ciclos. Para el Key Code de 5 bits se envían 10 bits ocupando 5 ciclos de la corriente. Así para transmitir un código X10 completo se necesitan 11 ciclos completos de la corriente eléctrica, como se muestra en las figuras 5 y 6.

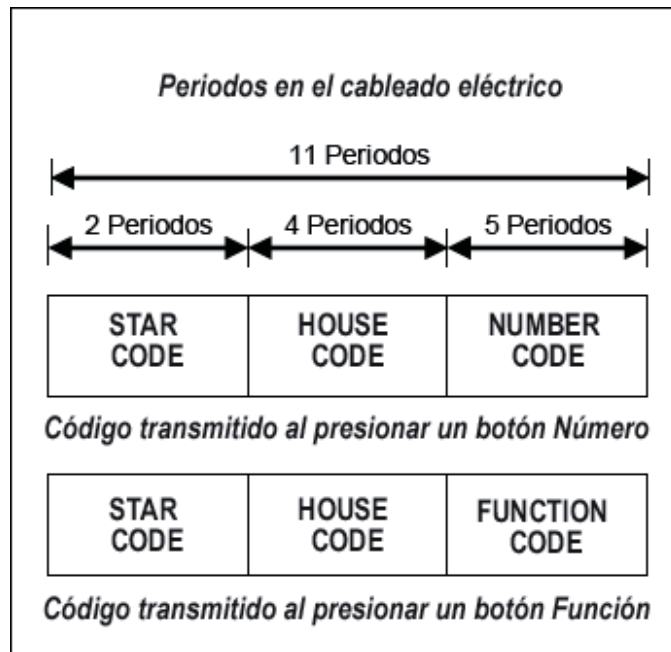


Figura 5: Composición de un comando X10

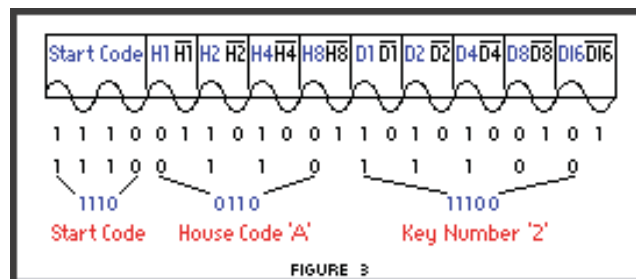


Figura 6: Composición de un comando X10

El bloque completo Start Code – House Code – Key Code es enviado en grupos de 2, con 3 ciclos completos de la corriente entre cada grupo de 2 bloques. Para el caso de comandos para aumentar o disminuir la iluminación, los bloques son enviados consecutivamente, sin los 3 ciclos entre ellos.

El comando viaja sincronizado con la corriente alterna en pares de bloques como lo muestra la Figura 7.

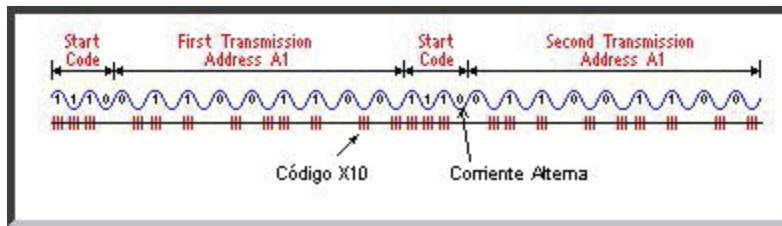


Figura 7: Sincronismo entre la señal X10 y la corriente alterna

En la Tabla 6.1, se muestran los códigos binarios que son transmitidos en la señal X10. El Start Code siempre es 1110 el cual es único y el cual no sigue la relación complementaria de los medios periodos alternados visto anteriormente.

HOUSE CODES					KEY CODES					
	H1	H2	H4	H8		D1	D2	D4	D8	D16
A	0	1	1	0	1	0	1	1	0	0
B	1	1	1	0	2	1	1	1	0	0
C	0	0	1	0	3	0	0	1	0	0
D	1	0	1	0	4	1	0	1	0	0
E	0	0	0	1	5	0	0	0	1	0
F	1	0	0	1	6	1	0	0	1	0
G	0	1	0	1	7	0	1	0	1	0
H	1	1	0	1	8	1	1	0	1	0
I	0	1	1	1	9	0	1	1	1	0
J	1	1	1	1	10	1	1	1	1	0
K	0	0	1	1	11	0	0	1	1	0
L	1	0	1	1	12	1	0	1	1	0
M	0	0	0	0	13	0	0	0	0	0
N	1	0	0	0	14	1	0	0	0	0
O	0	1	0	0	15	0	1	0	0	0
P	1	1	0	0	16	1	1	0	0	0
All Units Off					0	0	0	0	0	1
All Lights On					0	0	0	0	1	1
On					0	0	1	0	1	1
Off					0	0	1	1	1	1
Dim					0	1	0	0	0	1
Bright					0	1	0	1	1	1
All Lights Off					0	1	1	0	1	1
Extended Code					0	1	1	1	1	1
Hail Request					1	0	0	0	1	①
Hail Acknowledge					1	0	0	1	1	
Pre-Set Dim					1	0	1	X	1	②
Extended Data (analog)					1	1	0	0	1	③
Status-on					1	1	0	1	1	
Status-off					1	1	1	0	1	
Status Request					1	1	1	1	1	

Tabla 2: Códigos X10 en bits. (House Codes y Key Codes)

Hail Request se utiliza para saber si hay algún dispositivo escuchando en la red, si lo hay, el dispositivo responde con un Hail Acknowledge.

Extended Data y Extended Code se emplean para enviar otros códigos de funciones, si es que los existentes no son suficientes, o por ejemplo, para controlar dispositivos más complejos como un horno microondas o un refrigerador. Al enviar un Extended Data o Extended Code, a continuación de éste se envían 8 bits que indican la cantidad de datos a

transmitir, y luego se envían los bits de datos. Entre el extended data o code y los siguientes bits no deben haber silencios, es decir, la información debe enviarse continuada en los ciclos de la corriente eléctrica.

2.7 Elementos básicos de los sistemas X10

En la actualidad existe una muy variada gama de dispositivos X10 en el mercado, en efecto, prácticamente cualquier artefacto eléctrico tiene su correspondiente compatible X10, desde lámparas, sensores, portones eléctricos hasta lavadoras.

Por ejemplo, para armar instalar una lámpara X10 en el hogar y controlarla desde otro sector de la casa se pueden seguir los siguientes pasos:

1. Conectar una lámpara a un Módulo porta lámparas X10, y luego enchufar el módulo en la corriente eléctrica. - Asignar una dirección al módulo, para ello establecer el House Code y Number Code, por ejemplo, House Code=A y Number Code=1.
2. En otro enchufe de la casa, conectar un Módulo Interruptor X10 o similar, y asignar la dirección del dispositivo que se desea controlar, en este caso, A1.

Luego ya se puede tener control a distancia de la lámpara. Se puede hacer lo mismo con cualquier otro dispositivo, o por ejemplo, asignar la misma dirección a varias luces para que se enciendan con la pulsación de un solo interruptor. Se puede lograr el control remoto por teléfono instalando una central telefónica X10, la cual recibe llamados telefónicos y según la pulsación de alguna tecla inyecta alguna señal X10 en la red eléctrica para realizar alguna operación sobre algún artefacto.

Existen también dispositivos que permiten crear una interfaz con el computador, con lo que se puede llegar a tener control desde Internet de los aparatos eléctricos del hogar.

2.7.1 Controladores

Los controladores transmiten uno de las 256 posibles combinaciones de códigos del X10. Estos se representan como dos perillas, similares a las mostradas en la Figura 8. Una de A a P que constituye el código de casa y la otra de 1 a 16, que indica el código de unidad.

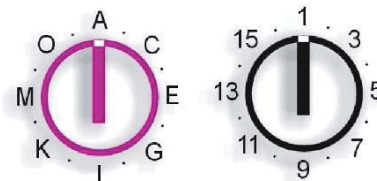


Figura 8: Controladores hasta 256 posibles códigos

Existen seis funciones básicas de los comandos de X10: encendido, apagado, brillo, atenuación, todas las luces encendidas y todas las unidades apagadas.

Los dispositivos pueden ser controlados por vanos medios como se había mencionado, ya sea controladores alambrados, de enchufe, o inalámbricos los cuales se describirán a continuación.

2.7.2 Controladores manuales

Son los controladores más básicos, simplemente se conecta y está disponible para ser manipulado manualmente por el usuario más cercano. Entre ellos tenemos al PHC01 y al PHC02 (Figura 9)



Figura 9: Controladores manuales PHC01 y PHC02

2.7.3 Controladores por teléfono

El controlador telefónico PHC05 permite operar con o sin una contestadora telefónica, partiendo de la suposición de que el controlador mismo será el que conteste la llamada. Este controlador permite encender hasta 10 luces y electrodomésticos utilizando cualquier teléfono desde cualquier parte del mundo. También puede controlar hasta 8 dispositivos en modo local.

Además tiene la opción que permite hacer centellear un módulo para lámparas cuando el teléfono suena para tener una indicación visual cada vez que hay una llamada entrante. Esta función es de gran utilidad para las personas con discapacidad auditiva.



Figura 10: Controladores PHC05

2.7.4 Controladores inalámbricos (por radiofrecuencia)

Es un sistema de control remoto inalámbrico que permite controlar hasta 8 módulos X10 desde el interior o el exterior de la residencia. También puede aumentar o disminuir la intensidad de las luces conectadas a módulos para lámparas y módulos para interruptores de muro. El receptor es un módulo para electrodomésticos que envía señales X10 hasta para 16 códigos de unidad con el mismo código de vivienda, lo que permite controlar hasta 16 luces y electrodomésticos usando un control remoto manual. Nótese que se debe utilizar un receptor por cada código de vivienda.

2.7.5 Receptores

Los módulos receptores son los encargados de ejecutar las órdenes recibidas a través de la red eléctrica de la casa. Hay dos tipos básicos: El módulo de lámpara y el módulo de aplicación. La diferencia es que el de lámpara permite regular la intensidad de las luces conectadas. El de aplicación funciona como un interruptor que conecta y desconecta el aparato que tiene enchufado. Hay módulos en formato de enchufar, que no necesitan instalación; los hay cableados directamente a la red en formato interruptor, que sustituyen a los que previamente existentes, y también existen de rosca.

- Receptores de enchufado
- Receptores de rosca
- Receptores de cableado

2.7.6 Interfaces

Muchos fabricantes electrónicos han adoptado el protocolo X-10 en sus productos como método para control. Como se explicó anteriormente, el protocolo X-10 se puede programar como una simple salida digital para los equipos, para así poder inyectar señales de control a la red eléctrica vía las interfaces uní o bidireccionales.

a) Interfaz de baja tensión

Para los dispositivos que previamente no están diseñados para operar con el X10, existe una interfaz denominada PSC01 (Figura 11), o interfaz Powerflash. Este se utiliza frecuentemente en sistemas de seguridad. La señal X-10 es activada por el cierre de un contacto seco o por una orden de bajo voltaje (6 - 18VDC), lo que le permite controlar cualquier dispositivo que se cableé a sus terminales.



Figura 11: Interfaz PSC01

b) Interfaz para conexión a un computador

Permite controlar cualquier artefacto X-10 desde la pantalla del computador. Se puede programar cronogramas de funcionamiento de luces y electrodomésticos las 24 horas del día. Un uso frecuente es controlar luces para que se enciendan y simular que la residencia se encuentra ocupada.

2.8 Filtrado de ruidos y señales

Debido a que el protocolo X-10 hace uso de la red eléctrica para transmitir los comandos necesarios para gestionar los recursos X10, es imprescindible que el nivel de ruido en la red no entorpezca el correcto funcionamiento de dichos recursos. Una lámpara podría desconocer algún comando enviado por un transmisor confundiendo con un simple ruido de la red y no apagarse en el momento indicado.

Para esto existen dispositivos capaces de filtrar, repetir y amplificar la señal en aquellos puntos controversiales. Entre ellos tenemos:

- **XPCP:** Acople pasivo. Enlace entre las fases eléctricas de una instalación de hasta 900 mts. cuadrados.
- **XPCR:** Repetidor de acople, diseñado para acoplarse hasta tres fases y repetir señales validas X10, también puede amplificar señales en instalaciones mas grandes.

Para obtener el nivel de ruido que afecta a los dispositivos tenemos el equipo para testear ruido:

- **XPTT:** transmisor de señal de prueba
- **XPTR:** indicador de la calidad de la señal

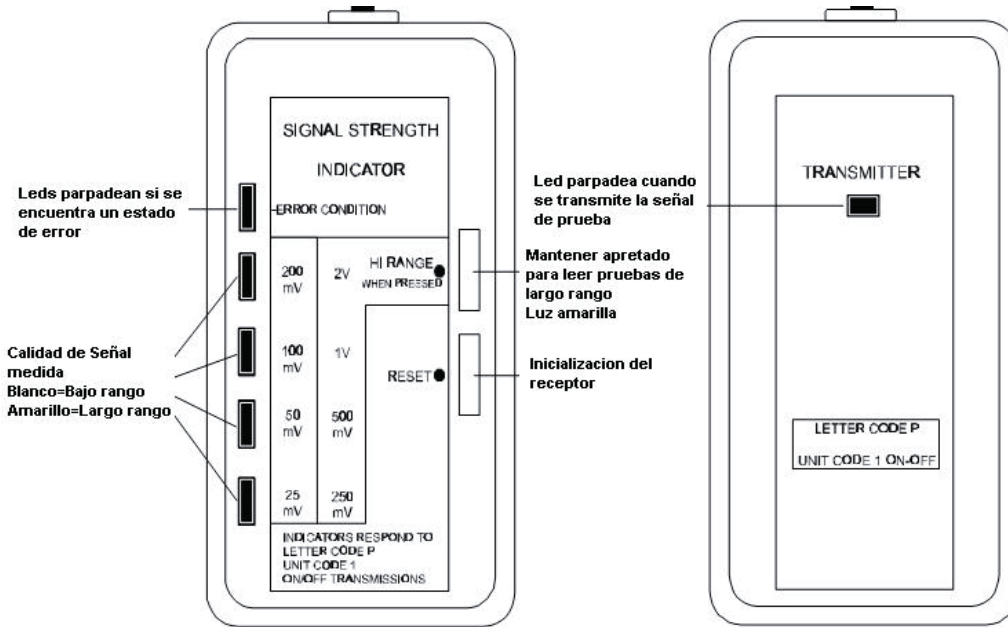


Figura 12: Dispositivos para detección de ruido. Izquierda: XPTR, derecha: XPPT

Antes de realizar la prueba, es recomendable cambiar cualquier dispositivo que tenga como identificador "PI" (Código de letra P y Número de unidad 1), ya que el transmisor utiliza esta dirección por defecto. Si existe otro dispositivo con esta misma dirección puede generarse resultados erróneos, e incluso el XPTT se puede encielar.

Además se recomienda inicializar el dispositivo receptor apretando el botón de reinicio. Si el LED de error está parpadeando, es indicativo de que existe ruido excesivo en la red, o señales electrónicas falsas o incorrectas en la línea de transmisión.

El XPTT transmite una señal de 2 voltios y el XPTR puede detectar señales de 2 voltios hasta un mínimo de 25mV. La señal mínima para que un módulo responda correctamente es de 100mV.

El XPTR se deja conectado, y se comienzan a desconectar todas las posibles fuentes de ruido, uno por uno, hasta detectar variaciones en la amplitud de la señal enviado por el XPTT. La variación se presenta como un aumento en los leds rojos que posee el dispositivo.

También se debe considerar que algunos dispositivos que sean de cableado en vez de enchufado pueden ser los causantes de ruido en la red eléctrica. Por lo que se debe proceder por desconectar los disyuntores, hasta probar todos, en búsqueda de posibles fuentes de ruido.

Con respecto a los filtros, hay que recalcar que el XPPF, o filtro de enchufe, descrito, no se debe conectar a equipos que demanden mas de 5 amperios, ya que esta es la corriente máxima que puede permitir el filtro. Por ello este dispositivo sólo se recomienda para uso

residencial. Similarmente, el XPF puede soportar corrientes hasta de 20A. En la Figura siguiente se describen las características físicas del filtro XPPF.

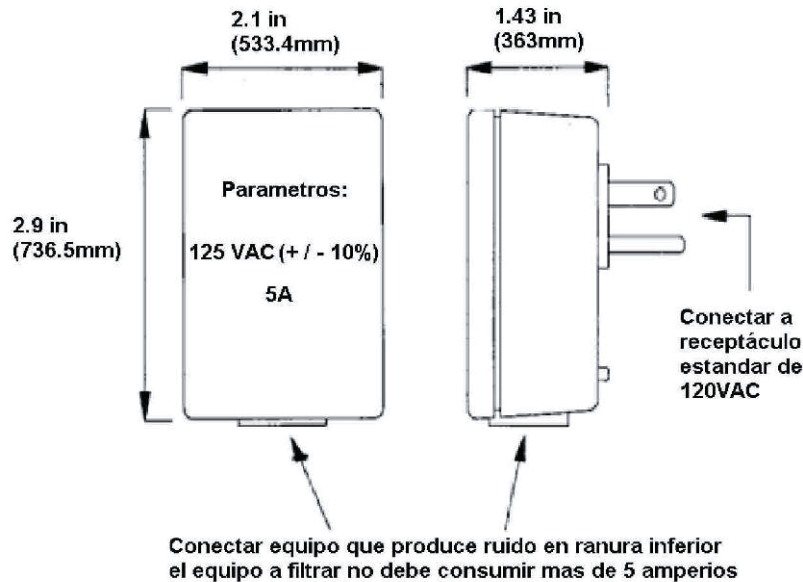


Figura 13: Dimensiones del filtro XPPF para uso residencial1"1

Con respecto al reductor de ruido XPNR, el diagrama de conexión se muestra a continuación:

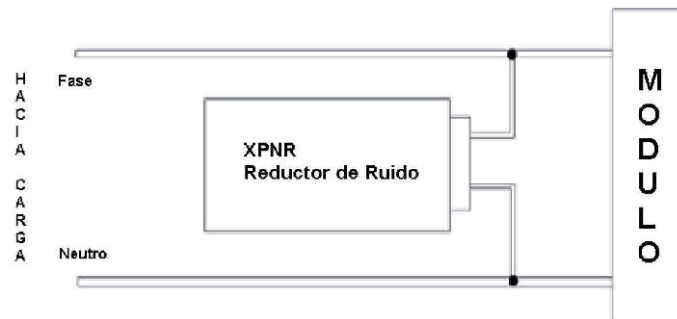


Figura 14: Filtro XPNR para reducir ruido

Como se puede observar, el XPNR se conecta directamente a los conductores en el circuito ramal.

En el caso que se detecte que el ruido afecta a un dispositivo específico o que este mismo lo provoca, existen:

- Filtro de alambrear: Si el dispositivo que se quiere filtrar no es de enchufe, sino que esta directamente en el cableado.
- Filtro de enchufe, igual que el anterior pero para dispositivos que se enchufan.

Cuando no se sabe con exactitud que provoca el ruido se utiliza el filtro reductor de ruido XPNR. Consideraciones de diseño para sistemas domóticos

Si bien es cierto, la tecnología X10 se destaca por su facilidad de uso e implementación, existen consideraciones que no se deben pasar por alto si se desea un alto grado de fiabilidad.

2.9 Consideraciones de implementación

2.9.1 Tamaño de a residencia

La atención que se debe prestar al tamaño de la residencia radica en que la distancia entre un receptor y un transmisor puede provocar una pérdida en la señal durante la transmisión, debido al estado del medio o las largas distancias a recorrer.

2.9.2 Medio de comunicación

Se deben tomar en cuenta siempre el lugar donde se implementaran los dispositivos para definir finalmente por que medio se comunicaran. Los medios pueden ser una de las tres plataformas que soporta X10: Red eléctrica, radio frecuencia (RF) e Infrarrojo.

Normalmente la implementación por el cableado por la red eléctrica, principal medio de comunicación en estos dispositivos. Sin embargo, deberá considerarse el estado de los materiales dentro de la residencia, un cable demasiado desgastado generara ruidos que harán errática la comunicación entre dispositivos. En viviendas muy antiguas pasara esto, por lo que se recomienda renovar el cableado o usar medios de transmisión por RF principalmente. En esos caso es mas eficiente aunque mas costoso.

Radiofrecuencia es muy útil en cuando ofrece movilidad, ya que no es necesario dirigirse a un lugar específico para realizar un mando y por que los dispositivos pueden alojarse en lugares donde no existe cableado. No es recomendable usar RF cuando el lugar existe la posibilidad de que otros emisores RF puedan provocar interferencia.

Finalmente en el caso del problema por interferencias por RF y si es que se desea controlan un dispositivo a distancia utilizamos infrarrojo, que por ser unidireccional y con línea vista no es muy utilizado.

2.9.3 Estado del medio de transmisión

Un punto importante es el estado del medio de transmisión, ya que el caso de un medio cableado, el ruido producido por el deterioro de los cables afectara la comunicaron mediante el protocolo X-10.

Para determinar la viabilidad de la transmisión por este medio, es donde se necesita usar el equipo de testeo de ruido compuesto por XPTR y XPTT. Con estos se mide la pérdida en la transmisión para probar que la señal llegara correctamente de un punto a otro.

2.9.4 Tipo de transmisor

Una vez que se tenga identificado el tipo de transmisión que se quiera utilizar, se procede a escoger un transmisor adecuado para el medio seleccionado. También se debe considerar si se requiere controlar los dispositivos localmente o se busca controlarlos remotamente. Para ello se puede recurrir ya sea a transmisores de escritorio, o telefónicos controlados por tonos, e incluso transmisores conectados a una computadora mediante una interfaz de traducción a RS-232.

Lo recomendable es utilizar transmisores capaces de comunicarse con la computadora pues existen muchas aplicaciones que pueden flexibilizar los sistemas domóticos hechos con base en un código computacional, que es la base de un sistema de control automático.

2.9.5 Tipo de receptor

Una vez que se sabe el tipo de transmisor que se utilizará, hay que seleccionar los receptores adecuados, considerando tanto el tipo de carga que se va a controlar, como la compatibilidad con el transmisor. En el aspecto de compatibilidad, lo importante es saber si los receptores que se van a utilizar son unidireccionales o bidireccionales, esto porque los últimos son compatibles con la interfaz para computadores.

Generalmente existe un modelo similar para controlar por computadora y otro por interfaces de escritorio o pared.

Con respecto a los tipos de cargas que se van a controlar, se debe tomar muy en cuenta su naturaleza, ya sea inductiva (no lineal) o resistiva (lineal). Esto porque los dispositivos del módulo cambian según sea el caso. También es importante tomar en cuenta la potencia que va a consumir la carga controlada así como la corriente máxima de la misma.

También se debe tomar en cuenta el tipo de conexión que se va a realizar, ya que existen los dispositivos de enchufe, que son para residencias con instalaciones ya desarrolladas, a las cuales no se pueden realizar trabajos de cableado extra. También están los dispositivos de cableado que pueden ser implementados en nuevas instalaciones eléctricas, los más comunes de este tipo son los interruptores de pared. Por último están los dispositivos de roseta, exclusivos para control de luminarias.

Así por ejemplo, existen módulos de enchufe para lámparas incandescentes bajo el nombre de PLM01, y si se requiere controlar por computadora está el PLM21, ambos soportan potencias de 40-300W, y corrientes hasta de 2A. Luego están los de cableado, bajo el código XPDF, que soporta también hasta 300W. Y por último el XPFM para corrientes hasta de 15 A.

Luego su contraparte para cargas inductiva (no lineales) bajo el código de PAM01 y PAM22, son los dispositivos unidireccional y bidireccional respectivamente. Y también el PAM04 que es para electrodomésticos de hasta 20A y un voltaje de 220V.

2.9.6 Reducción de ruido en el sistema

En caso de presentarse la necesidad de introducir en el sistema un dispositivo que produzca mucho ruido (el caso de los motores por ejemplo de un ventilador), se recomienda enchufarlo anteriormente a uno de los dispositivos descrito que reduce la cantidad de ruido en el sistema (filtros).

Si el ruido no es producido por un dispositivo conocido existen los filtros que son conectados a las fases de la red eléctrica.

2.9.7 Control de brillo

En el caso de las luminarias, existen dispositivos a los cuales les es posible regular la intensidad, aumentando y disminuyendo su brillo. En estos casos hay que destacar que precisan unos módulos controladores específicos para cargas lineales.

2.10 Mercado actual de software X10

Existen en el mercado algunas soluciones que utilizan tecnología X10, pero no proveen la posibilidad de incrementar su potencial agregando nuevos módulos de comunicación fuera de los proporcionados. Se presentarán a continuación dos sistemas comerciales basados en X10.

a) Software ActiveHome

El software de domótica ActiveHome viene incluido con el interfaz. S110210 y su utilización es sencilla e intuitiva, no requiere de conocimientos de programación, ni tener un equipo hardware de última generación, con un 386 y 4 Mb de RAM puede utilizarlo. Viene provisto de una pantalla gráfica donde rápidamente comprenderá el manejo, ya que dispone de ejemplos pre-programados. Para la programación de módulos al amanecer o al crepúsculo, puede elegir de la lista, la zona geográfica. Programando de esta forma el apagado de la luz del porche al amanecer, el cierre de persianas eléctricas al anochecer, para simulación de presencia, etc. También puede programarse activar un determinado módulo (luz/aparato) en una fecha concreta.

La programación de Macros se realiza de la siguiente forma; Por ejemplo la macro B1 (que podemos denominarla "Ambiente salón") hará que el módulo A1 se apague (luz del techo del salón), el módulo A2 se encienda al 50 % (Halógenos del mueble de mampostería), y que medio minuto más tarde comiencen a encenderse el módulo A3 hasta alcanzar el 40 % (luces indirectas de los cuadros de la entrada). Y así podemos programar en una sola macro las órdenes que nuestra imaginación llegue a concebir para crear los ambientes deseados.

El software está en español, y todas las etiquetas y nombres de los módulos y sistemas domóticos se pueden personalizar en cualquier idioma sin ningún problema. Además su interfaz es totalmente gráfico por lo que su utilización resulta muy intuitiva.

b) Power Home

El Power Home es un paquete de software para la automatización de la casa que permite que controles la iluminación y aplicaciones caseras así como los dispositivos infrarrojos. La iluminación y las aplicaciones son controladas vía el Insteon y los reguladores siguientes X-10: PowerLinc V2 (USB y cuento por entregas), CM11A, CM17A, MR26A, PowerLinc (cuento por entregas y USB), W800RF32, y CPUXA/Ocelot. El control infrarrojo se alcanza a través de los reguladores IR siguientes: El círculo (telecontrol infrarrojo automatizado), Multi-CÍRCULO, RedRat2, RedRat3, CPU-XA/Ocelot, USB-UIRT, y slink-e. Con el CPU-XA/Ocelot y los módulos adicionales o el regulador de Velleman K8000 también tienes acceso a las entradas digitales/a las salidas, las entradas análogas y las salidas análogas (K8000 solamente).

Con este interfaz programable, el control se alcanza vía teclado, ratón, email, X-10, el reconocimiento IR, de voz, la mensajería de Windows.

Características:

- Interfaz completamente programable con opción de idiomas. Puedes utilizar scripting interno o cualquier lengua apoyada por el anfitrión de Windows Script (por ejemplo: VBScript, JScript, etc.)
- Web server interno para el mando a distancia y supervisar. Apoya el contenido dinámico definido por el usuario vía PSP (las páginas caseras del servidor de la energía)
- Reconocimiento de voz.
- El servidor incorporado del WAP para el mando a distancia vía Internet.

Como conclusión, se puede afirmar que no se dispone de una amplia gama de software de control de dispositivos X10, y los que existen, son en su mayoría para control de los dispositivos desde el PC. Sin embargo, el presente proyecto, proporciona un control dispositivos tanto de forma remota a través de un navegador Web y para cualquier sistema operativo.

3 Web Services

Un Web Service es un sistema software diseñado para apoyar la interoperabilidad máquina-máquina interactuando a través de una red. Es una clase cuya interfaz se puede hacer pública en un servidor Web mediante un lenguaje de descripción de interfaces (WSDL). Dicha interfaz ofrece un conjunto de actividades que un cliente puede invocar. El cliente accede al Web Service usando los estándares de Internet.

Para acceder a un Web Service se pueden utilizar varios protocolos Web estándar como HTTP GET o HTTP POST, aunque se ha diseñado un protocolo específicamente diseñado para ello: SOAP (Simple Object Access Protocol). Este protocolo se basa en la utilización de HTTP para el transporte de mensajes y el lenguaje XML para la escritura del cuerpo de estos mensajes. Todo esto permite a cualquier aplicación ser capaz de generar e interpretar mensajes en SOAP independientemente de la plataforma. [05]

Características:

Las características de esta nueva tecnología son las que se citan a continuación:

- **Interoperabilidad:** Los Servicios Web se pueden consumir por clientes de otras plataformas.
- **Acceso externo desde Internet:** Los Servicios Web realizan una buena gestión para los accesos que provienen de clientes de Internet.
- **Tipos de datos de las Interfaces:** Los tipos de datos definidos para los Servicios Web se corresponde con los tipos de datos definidos por la mayoría de lenguajes de programación.
- **Uso de los estándares de Internet:** Los servicios Web utilizan los estándares de Internet y evitan, en la medida de lo posible, reinventar soluciones a problemas que ya están resueltas.
- **Soporte de cualquier lenguaje:** La implementación de un Servicio Web no está ligada a un particular lenguaje de programación. Esta es una gran ventaja frente a otras tecnologías como Java RMI, que está completamente ligada al uso de lenguaje Java, haciendo realmente difícil hacer una llamada a un objeto Java desde un objeto Visual Basic o Perl. De este modo, un cliente puede implementar o usar un Servicio Web independientemente del lenguaje de programación en el que fue implementado.

- Soporte para cualquier infraestructura de componentes distribuidas: Los Servicios Web no están ligados a una arquitectura de componentes en particular. Los protocolos facilitan a nivel base la comunicación entre las distintas infraestructuras de objetos distribuidos. Tecnologías de Web Service

Arquitectura de servicios Web implica a muchos niveles y tecnologías relacionadas entre sí. Hay muchas formas de visualizar estas tecnologías, así como hay muchas maneras de crear y utilizar servicios Web. La Figura a continuación proporciona un ejemplo de la tecnología de algunas de estas familias.

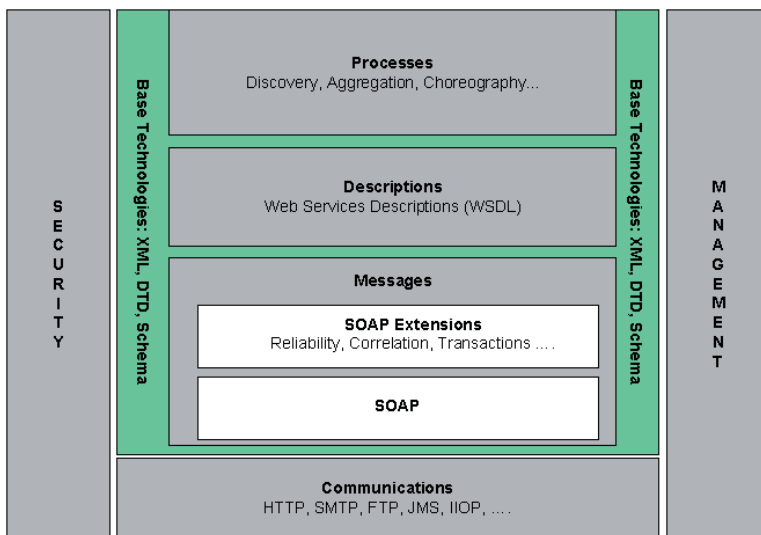


Figura 15: Principales tecnologías de Web Service

3.1 XML

XML o en inglés Extensible Markup Language, es un estándar desarrollado por W3C [06], el cual se caracteriza por ser independiente de la plataforma y que sirve (fundamentalmente) para describir datos, sin un formato estructurado. Esto encasilla a XML como un “metalenguaje”, dicho en otras palabras, un lenguaje que sirve para conformar otros lenguajes de marcado; lo cual brinda la facultad de generar descripciones entendibles tanto para humanos como para las máquinas. Un sencillo ejemplo de un envío de datos con XML es el siguiente:

```
<Pais>
<name> Chile </name>
<capitol> Santiago</capitol>
<animal> Pudu </animal>
<bird> Condor </bird>
</Pais>
```

Esto nos dice que la capital de Chile es Santiago, que el animal representativo de Chile es el Pudú, que el ave es el Cóndor y que el árbol es la Araucaria. Como vemos la composición es heredada de HTML en su estructura de tags, pero lo que logra es separar la presentación de los datos, cosa que es muy importante para desarrollar aplicaciones Web robustas, y con la incorporación de abstracciones como MVC descritas anteriormente. También como se indicó anteriormente, este lenguaje es un pilar fundamental de la tecnología WS, al proveer de lo necesario para la comunicación y descripción entre ellos o entre las aplicaciones que hagan uso de ellos, lo cual provee interoperabilidad y eficiencia.

XML sale a la luz en el año 1998, proveniente de un lenguaje llamado SGML, con las siguientes metas por alcanzar:

- Será fiable y ampliamente usado en Internet.
- Dará bases y soporte a una amplia gama de aplicaciones.
- Será compatible con su predecesor, SGML.
- Será fácil realizar programas que interactúen con documentos XML.
- Las opciones extras generadas por XML serán mínimas.
- Los documentos XML serán legibles por humanos y de lógica fácil.
- El diseño XML será rápido de realizar.
- Los documentos XML serán fáciles de crear.

Esto confirma todo lo anteriormente descrito, especificando cual es el fin de XML como metalenguaje. Aunque nunca se consideró algún estándar en la sintaxis de marcado se han generado iniciativas como GML que consta de ciertas declaraciones y modos de hacerlo para ser fácilmente entendible por máquinas y humanos, también está el caso Geotech-XML (Geography Markup Language), los cuales entregan resultados excelentes en las aplicaciones que los ocupan. Existe el Mathematical Markup Language (MathML), generado por W3C que maneja operaciones matemáticas, XBRL (Extensible Business Reporting Language) se usa para describir datos financieros y de negocios, también Wireless Markup Language (WML) u otros aplicados en distintas áreas.

3.2 WSDL: Web Service Description Language

WSDL es el lenguaje de descripción de Web Services basado en XML. WSDL permite describir la interfaz ofrecida por un Web Service, las operaciones que el servicio puede soportar y los parámetros de entrada y salida [07]. El elemento principal de un documento WSDL es el bloque de definiciones, aunque admite otras extensiones opcionales. El bloque de definiciones está compuesto a su vez por cinco bloques: Types, Message, PortType, Binding y Service.

- **Types:** Contiene las definiciones de los mensajes que pueden ser enviados o recibidos por un servicio. Se representan en un esquema utilizando XML Schema.
- **PortType:** Define el conjunto de interfaces que ofrece el Web Service. Cada interfaz es asociada con uno o más mensajes. Se especifica una interfaz por cada

uno de los métodos de acceso que se deseen utilizar SOAP, HTTPGET o HTTPPOST. En el siguiente ejemplo solamente se muestra para SOAP:

- **Service:** Define una colección de puertos ofrecidos por el Web Service. Dichos puertos son cada elemento de Binding con su correspondiente URL.

Además de estos bloques, el bloque de definiciones establece los espacios de nombres que serán utilizados en el documento WSDL.

```
<?xml version="1.0" encoding="utf-8"?>
<definitions
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:s0="http://tempuri.org/"
...
targetNamespace="http://tempuri.org/"
xmlns="http://schemas.xmlsoap.org/wsdl/">
</definitions>
```

3.3 UDDI: Universal Description, Discovery and Integration

UDDI proporciona un servicio de directorio centralizado para publicar información acerca de Web Services. En el directorio UDDI se publican los ficheros WSDL que describen a los Web Services que se ofertan [08].

Presenta una estructura (Figura 16) formada por un conjunto de elementos tipo Registry y otro de elementos tipo Registrar. Los Registry tienen una copia del directorio UDDI al completo, mientras que los Registrar proporcionan servicios de registro al cliente. Un Registrar puede estar proporcionado por una compañía, que tenga su Registry, y mediante una interfaz HTML, por ejemplo, nos permita modificar los registros del directorio. Los cambios hechos en un Registry, son actualizados automáticamente en el resto. Un cliente accede a un Registry mediante uno de los Registrar del mismo, y los cambios se propagan a los demás.

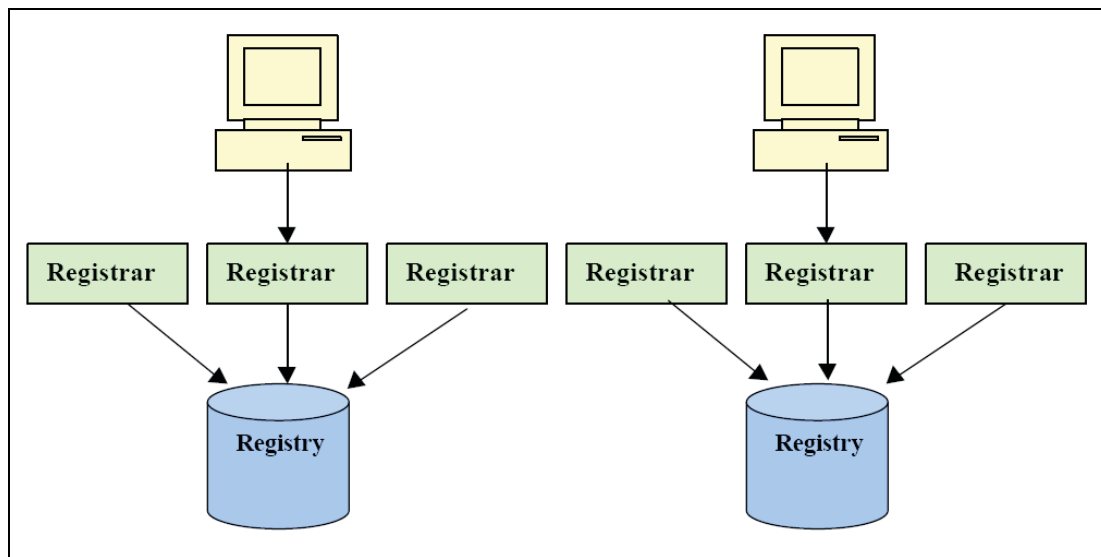


Figura 16: Estructura de UDDI

3.4 SOAP (Simple Object Access Protocol)

Definido por la W3C, la descripción de esta dice: “SOAP es un protocolo liviano para intercambiar información en un ambiente descentralizado” [09]. El ciclo de vida SOAP lo constituye en mensaje SOAP request, el cual invoca a un WS, y posteriormente, un mensaje SOAP response para dicha solicitud. Sus principales características son las siguientes [11]:

- No está restringido a un lenguaje ni a una plataforma. SOAP no especifica una API, lo que permite al programador abstraer el lenguaje sobre el que pueda estar implementado el objeto invocado. Esto facilita la reusabilidad de código, y las actualizaciones de la aplicación.
- No está restringido a un protocolo de transporte. La especificación SOAP describe su transporte en HTTP pero puede ser transmitido en cualquier protocolo de transporte de texto.
- No está restringido a un Middleware.
- Está basado en estándares ya existentes. De hecho, incluso el sistema de tipos que utiliza es el de XML.
- Hace posible la comunicación entre entornos muy heterogéneos, ya que cualquier aplicación capaz de escribir y leer XML sobre HTTP puede utilizar SOAP.

Un mensaje SOAP esta estructurado en una cabecera y un cuerpo. En primer lugar aparece la cabecera, es opcional y sirve para poner información complementaria al cuerpo. Por ejemplo, si la información del cuerpo del mensaje estuviera comprimida o cifrada, aquí

se indicaría la información a cerca de la compresión o del cifrado. Como este campo es opcional, y muchas aplicaciones lo ignoran, para evitar que sea descartada se debe indicar con la etiqueta ‘mustUnderstand=’1’’. A continuación aparece el cuerpo del mensaje, que es obligatorio. Aquí puede aparecer cualquier texto, pero en la especificación de SOAP se propone un método de codificación que se describe a continuación:

Dentro del bloque body, hay que definir otro con el nombre del método, y dentro de éste, tantos como parámetros necesite la invocación. Los parámetros deberán tener el mismo nombre, y posición que en la definición del método. A continuación se muestra la diferencia entre el uso de tipos simples y tipos estructuradas en un mensaje SOAP.

- **Tipos simples:** Los tipos simples de XML son un conjunto de tipos estándar para muchos lenguajes de programación (int, boolean, float, string, etc). Una instancia de estos tipos es codificada como un elemento XML.
- **Tipos Estructurados:** Se codifica la estructura como un elemento XML, y dentro del mismo cada uno de sus campos.

El paso de parámetros por referencia también está contemplado. Una invocación a un procedimiento con parámetros por referencia es exactamente igual que si fueran por valor. La diferencia radica en que el mensaje de respuesta nos retornará también los parámetros, con sus nuevos valores. SOAP también proporciona un mecanismo para comunicar que se ha producido un error en el servidor, mientras procesaba la petición. Para ello incorpora el elemento Fault en el bloque body. Existen una serie de errores predeterminados, cuyo significado se detalla a continuación:

3.5 Arquitectura de Web Services

En una arquitectura de Web Services hay dos partes claramente diferenciables, el modo de utilizar un Web Services y cómo desarrollarlo. A continuación se detallan las partes implicadas y los pasos necesarios para publicar un Web Service ya desarrollado y cómo puede ser utilizado (Figura 17).

1. El programador desarrolla el Web Service
2. El programador describe el Web Service en un fichero WSDL
3. El programador publica el Web Service en un directorio como UDDI
4. La persona suscrita al directorio busca el Web Service
5. La persona suscrita al directorio invoca el servicio con SOAP
6. La persona suscrita al directorio recibe la respuesta mediante SOAP.

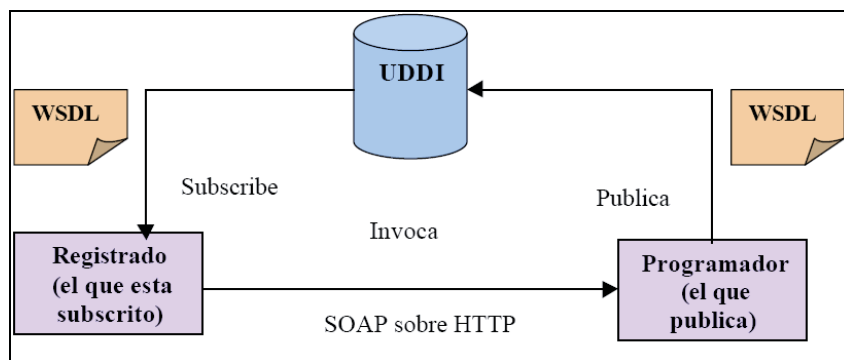


Figura 17: Arquitectura Web Service

La arquitectura necesaria para el desarrollo de Web Services es la de un servidor que contenga las herramientas adecuadas para el soporte al desarrollo de este tipo de tecnología. Estas herramientas proporcionan el entorno de desarrollo de Web Services y la gestión de invocaciones de los servicios Web.

3.6 SOA

SOA es una arquitectura en la cual componentes de software débilmente acoplados pueden interactuar, independientes de los protocolos y tecnologías utilizadas, permitiendo flexibilidad para la integración de estos

Hay que destacar que SOA no es una tecnología, sino un enfoque que -en términos simples- provee una metodología y un marco de trabajo para diseñar una arquitectura de software orientada a los servicios.

Los servicios, se caracterizan por tener las siguientes propiedades:

- **Vista Lógica:** El servicio es una abstracción de las aplicaciones, bases de datos y/o procesos de negocios, definida en términos de lo que hace, es decir, que operación del negocio realiza.
- **Orientación a Mensajes:** El servicio se define formalmente, en términos de los mensajes intercambiados entre proveedores y consumidores de servicios y no está basado en las propiedades del lenguaje de programación, bases de datos, procesos, u otros. La estructura interna de éstos se abstrae en SOA, lo que permite incorporar cualquier componente o aplicación a ésta arquitectura.
- **Granularidad:** Los servicios son modelados débilmente acoplados, permitiendo un mayor nivel de abstracción de estos.

SOA y Servicios Web, están íntimamente relacionados, ya que debido a la naturaleza de “servicios” propia de SOA, los Servicios Web, proporcionan una plataforma tecnológica para apoyar la integración de procesos de negocios de la organización. Los Servicios Web,

se basan en un conjunto de protocolos mediante los cuales es posible publicar, descubrir y usar servicios de una manera estándar y tecnológicamente neutra

3.7 JAX-WS

JAX-WS [10] es una nueva API (basada en servicios Web XML) para el desarrollo de WS en un ambiente JAVA. Es llamada como uno de los “servicios Web base JAVA” y como los demás tildados con ese nombre, se usa en conjunto con otras tecnologías. Es la nueva generación en del modelo de programación de Servicios Web que complementa la infraestructura que proporciona el modelo anterior JAX-RPC.

Como tecnología, JAX-WS tiene como fin el desarrollar servicios Web JAVA (SOAP y RESTful – servicios Web con herramientas REST o representational state transfer) y dar la posibilidad a clientes y WS de comunicarse usando XML (documentos XML) como lenguaje común para la estructura, descripción y mensajes entre ellos. Lo anterior es por medio de llamados remotos SOAP pero ocultándolos, por medio de una API basada en JAVA llamada JAXB (Java Architecture for XML Binding). Los desarrolladores utilizan la API que brinda JAX-WS para definir métodos, y la API se encarga de la comunicación. Por otro lado, los clientes crean un Proxy local para representar el servicio, pudiendo luego invocarlos directamente. El runtime de JAX-WS convierte las llamadas descritas por medio de la API y hace un matching para crear mensajes SOAP.

La implementación de éste estándar de programación proporciona las mejoras:

- Con las API de JAX-WS se simplifica el desarrollo de clientes y servicios Web, aportando una mayor independencia de la plataforma para las aplicaciones JAVA. Se aprovecha el mecanismo de Proxy dinámico para proporcionar un modelo de delegación formal con un proveedor.
- La fácil exportación a servicio Web y su independencia de transporte. Utiliza un modelo de programación compuesto por código de aplicación, anotaciones de clases, y la capa de mensajes XML (transformados automáticamente).

Luego de anotar con Servicio Web la clase, tenemos que TODOS los métodos públicos pasan a ser operaciones del WS, y se genera automáticamente WSDL/SCHEMA.

- Con JAX-WS, los Servicios Web se invocan en de forma sincrónica y asíncrona, lo anterior por medio de un soporte para sondeo y devolución de llamada cuando se invocan los servicios de forma asíncrona. Con esto, se puede emitir una petición y obtener un objeto de respuesta que se sondea para ver si el servidor ha respondido, recuperando la respuesta cuando esto ocurre. Esto permite al cliente continuar su trabajo sin necesidad de esperar una respuesta, y ofrecen un modelo más dinámico y eficaz de invocar servicios Web.

- Hace uso de la API y herramientas JAXB (Java Architecture for XML Binding) como tecnología de enlace para las correlaciones entre los Objetos Java y documentos XML. JAX-WS se basa en esta API para el enlace de datos y generación de correlaciones en dos direcciones (JAVA-XML y XML-JAVA).
- Se especifican mecanismos para el establecimiento de clientes dinámicos o de asignación (relativos a invocaciones asíncronas), y se establece el modelo de programación para clientes estáticos (Proxy), el cual invoca un servicio Web basado en una interfaz de punto final de servicio (SEI) que se proporciona.
- Mediante JAX-WS se da la posibilidad de envío de documentos binarios como imágenes o archivos mediante la petición de servicio Web. Se provee de un mecanismo especializado como MTOM (Message Transmisión Optimización Mechanism).
- JAX-WS proporciona más de una tecnología de enlace, estas pueden ser XML Source, SAAJ (SOAP Attachments API for JAVA) y JAXB. Esto permite un amplio margen de adecuación a cada problemática a solucionar con este nuevo modelo de programación.
- En JAX-WS 2.0 se añade el soporte para SOAP 1.2 y 1.1, lo que posibilita el envío de documentos adjuntos. Junto con esto se proporciona un mecanismo para la optimización de estas transferencias.

3.8 JAXB

JAXB (Java Architecture for XML Binding) proporciona una forma fácil de correlacionar clases JAVA y esquemas XML, proporcionando una forma transparente para enlazar esquemas XML con aplicaciones JAVA, sin necesidad de ser expertos en XML.

JAXB da soporte a la transformación entre objetos de esquema y objetos JAVA, así como entre documentos XML y objetos JAVA. Se compone de una API de tiempo de ejecución y herramientas que simplifican el acceso a los documentos XML. Se puede utilizar una definición de esquema XML (XSD) y la herramienta respectiva (xjc) o con un conjunto de Objetos y una herramienta (schemagen) para generar esquemas XML. Cuando exista relación entre XML y JAVA se utiliza JAXB en tiempo de ejecución.

Las clases anotadas JAXB contienen toda la información que necesita la API para procesar los documentos de instancia XML y los objetos JAVA respectivos y proveer el enlace en tiempo de ejecución, lo que permite operaciones para acceder, manipular y validar el contenido XML. Con esto se provee a un cliente la habilidad de convertir datos XML en objetos derivados de JAXB y también de objetos a documentos XML.

La Figura 18 muestra la arquitectura JAXB, donde se puede ver el proceso descrito anteriormente.

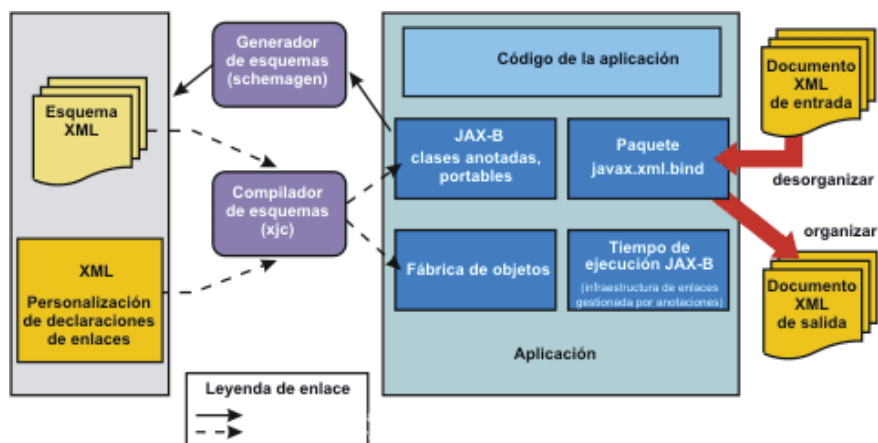


Figura 18: Mecanismo de funcionamiento JAXB

3.9 Implementación de Web Service

En la actualidad existen múltiples herramientas para construir un Web Service de manera fácil, a modo de ejemplo se puede utilizar en conjunto:

<i>Software o recurso</i>	<i>Versión requerida</i>
NetBeans IDE	Web & Java EE versión 6.0 o 6.1
Java Developer Kit (JDK)	versión 6 o versión 5
Java EE-compliant Web o application server	Tomcat Web server 6.0 y/o GlassFish application server v2 Ambos incluidos en la distribución JavaEE de NetBeans IDE

Crear un Web Server a partir del IDE de NetBeans es muy fácil, instrucciones de esto podemos encontrar en: <http://www.netbeans.org/kb/60/websvc/jax-ws.html>. En el caso de la Figura se ha creado un proyecto llamado *CalculatorWSApplication* y dentro de él se creó el Web Service *CalculatorWS*.

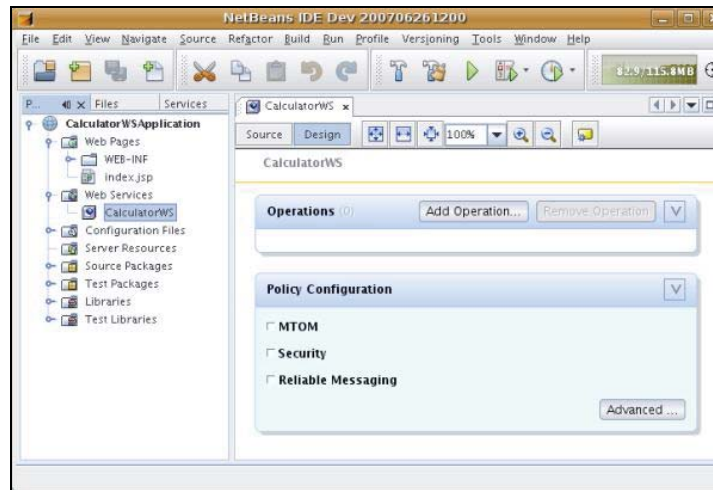


Figura 19: Creación Web Service mediante NetBeans

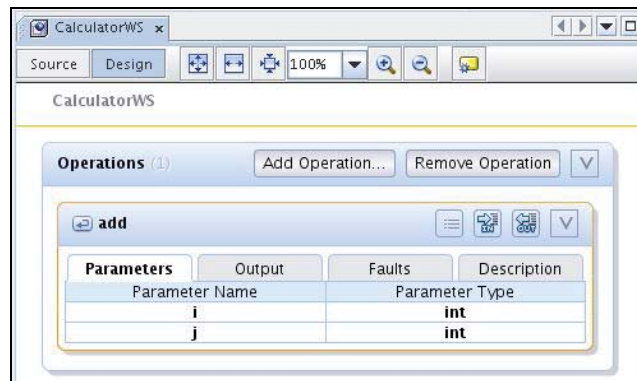


Figura 20: Edición de parámetros Web Service

Una vez creado el servicio, es posible abrirlo y editarlo (Figura 19). A primera vista pareciera ser una clase simple de Java, pero con ciertas anotaciones adicionales. Entonces es posible crear por ejemplo un servicio que sume dos números enteros, agregando un método que reciba dos parámetros *j* e *i*, y que retorne la suma de ambos, de la siguiente manera:

```
@WebService()
public class CalculatorWS {
    @WebMethod
    public int add(@WebParam(name = "i") int i, @WebParam(name = "j") int j) {
        int k = i+j;
        return k;
    }
}
```

Una vez listo el IDE se encarga de generar los archivos necesarios y hacer el despliegue hacia el servidor Web, de manera que sólo basta ejecutar la aplicación para que esté correctamente activa.

3.10 FrameWorks para Web Services

Encontramos una gran variedad de FrameWorks para su implementación en muchos y variados lenguajes de programación. A continuación se listaran algunos de ellos:

Nombre	Plataforma	Destinación	Protocolos
ActionWebService	Ruby (on Rails)	Cliente/Servidor	SOAP, XML-RPC, WSDL
Apache Axis	Java/C++	Cliente/Servidor	SOAP, WSDL
Apache Axis2	Java/C	Cliente/Servidor Asyn Sypport	SOAP, MTOM, WSDL 2.0, WSDL
AlchemySOAP	C++	Cliente/Servidor	SOAP
JSON-RPC-Java	Java	Servidor	JSON-RPC
Java Web Services Development Pack	Java	Cliente/Servidor	SOAP, WSDL
JAX-WS	Java	Cliente/Servidor	SOAP, WSDL
NuSOAP	PHP	Cliente/Servidor	SOAP, WSDL
Web Services Invocation Framework	Java	Cliente	SOAP, WSDL
Windows Communication Foundation	.Net	Cliente/Servidor	SOAP, WSDL
gSOAP	C/C++	Cliente/Servidor	SOAP, XML-RPC, WSDL
Zolera SOAP Infrastructure (ZSI)	Python	Cliente/Servidor	SOAP, WSDL

4 Arquitectura domótica

Con el fin de lograr seguridad en el hogar, oficinas o cualquier lugar que requiera ser resguardado tanto de una emergencia externa (en caso de intento de robo) e interna (incendios, fugas de gas, etc.), como también de proporcionar algún grado de comodidad y ahorro energético se desarrollo una arquitectura domótica que facilite la creación de escenarios según sea la necesidad en casa caso.

El gran potencial de esta arquitectura radica en la posibilidad de integrar tanto dispositivos X10 como genéricos a la vez que se pueden generar innumerables configuraciones con dispositivos tanto de entrada como de salida y las interacciones que se puedan generar entre un grupo de estaciones. Si bien los dispositivos X10 funcionan sin necesidad de un PC, al dotarlos de “inteligencia” podemos superar ampliamente su utilidad.

Los pilares fundamentales sobre los que se sostiene esta arquitectura domótica son los siguientes:

- Utilización de dispositivos X10 sensores y actuadores.
- Utilización de dispositivos genéricos (sensor de humedad, cámaras Web, etc.)
- Comunicación entre distintas estaciones dentro de un grupo definido
- Comunicación mediante la publicación de un WSDL generados por los Servicios Web de cada estación.
- Envío de mensajes electrónicos (email) y mensajes a teléfonos celulares (SMS)

Es importante, para comprender cabalmente el uso que se le puede dar, entender algunos conceptos básicos:

- Una estación es en estricto rigor un PC que estará en una vivienda.
- Cada estación pertenece a un grupo.
- Un grupo puede contener una o mas estaciones.
- La información de la conformación de los grupos se encuentra en la base de datos del Sistema de Gestión Central (SGC).
- Los usuarios pertenecientes a la estación son almacenados en el SGC.
- Los usuarios administradores de grupo son almacenados en el SGC.

La Figura siguiente muestra los conceptos anteriormente mencionados:

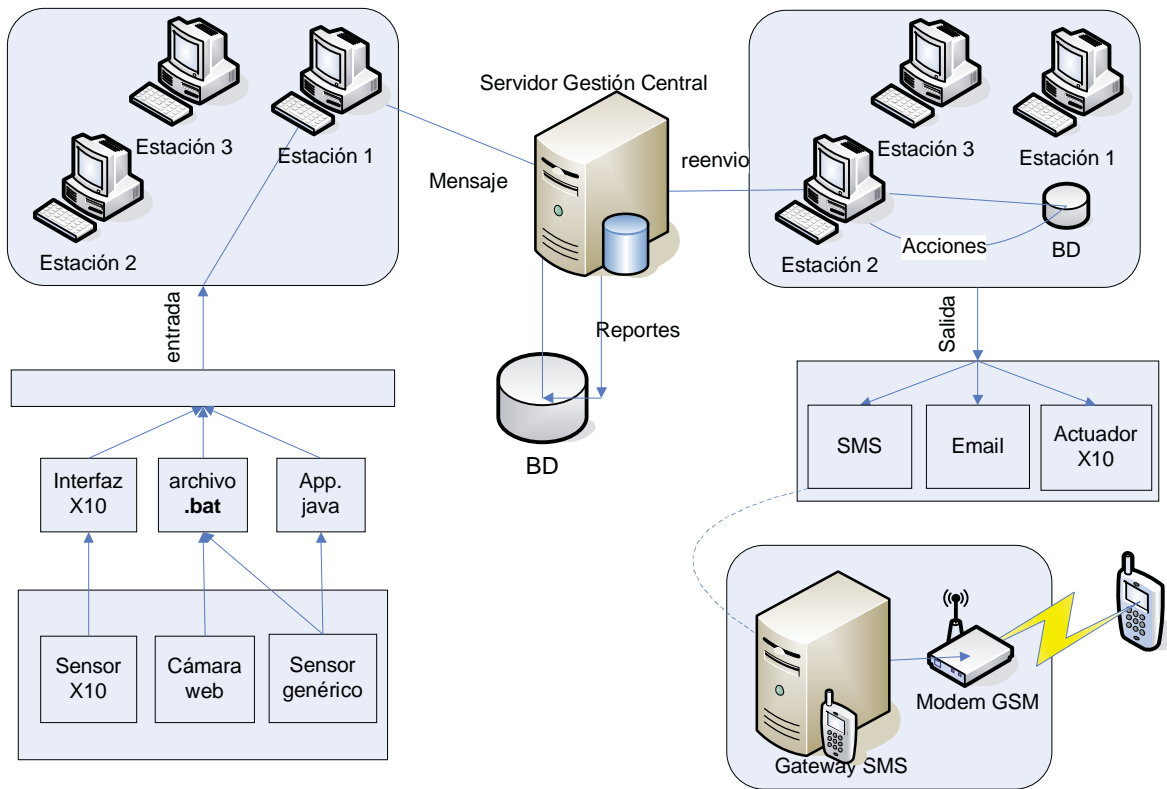


Figura 21: Modelo de despliegue de arquitectura

El sistema esta pensado para poder utilizarse en muchos ámbitos distintos, estos pueden ser:

- Condominios
- Villas
- Viviendas de familiares
- Sucursales de una empresa
- etc...

4.1 Aplicaciones de la arquitectura

La arquitectura cuenta con un set de aplicaciones que pueden ser configuradas para generar los escenarios que sean necesarios según sea el caso.

4.1.1 Servidor de Gestión Central (SGC)

El SGC es el encargado de lograr la comunicaciones entre las distintas estaciones para que pueden llevarse a cabo las acciones necesarias y configuradas ante cada evento obtenido, además de mantener la información sobre los sucesos ocurridos para poder tomar medidas en cuanto a nuevas configuraciones o dar mayor seguridad en ciertos puntos donde sea mas repetitivo alguna clase de evento. Se puede por lo tanto decir que tiene dos misiones principales:

a) Broker entre estaciones

Cuando una estación recibe una señal de alarma de alguno de sus sensores, envía un mensaje de alarma al SGC quien es el encargado de reenviar el mensaje (Código de Sensor, Código de estación) a todas las estaciones pertenecientes a su grupo.

b) Repositorio de Reportes

Cada vez que reenvía un mensaje, guarda un reporte del evento, estación, sensor, lugar y hora del hecho.

4.1.2 Aplicación Web para el Ipod Touch/Iphone

La aplicación Web para el Ipod Touch/Iphone se utiliza para las configuraciones de las distintas partes dentro del sistema domótico:

- Estaciones
- Grupos de estaciones
- Usuarios
- Acciones
- Reportes
- SMS

Esta es una aplicación Web cuyo lenguaje de programación es PHP. Cuenta con una base de datos en Mysql, para almacenar los reportes, grupo de estaciones y usuarios.

Para que un usuario de una estación pueda acceder a la aplicación, previamente el administrador debe haber ingresado la estación junto con el usuario.

A la estación se le debe especificar la dirección de su WSDL y un nombre. Al usuario asignado a la estación se le debe ingresar su nombre de usuario y contraseña. Además posee una sección para accionar los actuadores X10 en la estación.

4.1.3 Aplicación para cada estación:

Cada estación posee un Web Services quien recibe la información sobre un evento y puede ejecutar una o más de las siguientes acciones:

- Control sobre actuadores
- Envío de email
- Ejecución de acciones
- Conexión al Gateway SMS para envío de SMS

Las acciones que se crean a través de las aplicaciones Web son almacenadas en cada estación en una Base de datos MySQL, es por esto que cada vez que se genera una alarma (reenviada desde la estación X a la estación Y por el SGC), la aplicación busca en la BD la acción que concuerde con el sensor de origen y la estación de origen para su ejecución.

Las acciones permitidas y configurables por cada estación dentro del sistema pueden ser cualquiera de las siguientes:

- Acciones sobre actuadores X10
- Envío de Email
- Envío de SMS

Para que el SGC se reenvíe el mensaje de la estación X a la estación Y, cada estación posee un Web Service con los métodos necesarios para la interacción. Los métodos entre otros son:

- Ingresar Sensor
- Ingresar Actuador
- Crear acción
- Ejecutar una operación sobre un actuador
- Ejecutar acción

4.1.4 Aplicación SMS

Se cuenta con un Gateway SMS que se conecta mediante cable USB a un celular que hace la vez de MODEM GSM. Este Gateway recibe una copia de los mensajes enviados al celular que se le ha conectado y también puede enviar mensajes de texto a otro celular. Este Gateway es capaz de procesar los mensajes (operaciones de envío y recepción) mediante archivos PHP.

Existen dos aplicaciones encargadas de manejar los SMS dentro de la arquitectura, una capaz de enviar mensajes de textos a un celular previamente definido en una acción y otra que recibe un mensaje de texto y ejecuta una operación de encendido o apagado sobre un actuador.

a) Envío de SMS

Esta aplicación posee un Servicio Web que recibe la señal de la estación para enviar el mensaje. Como parámetro de entrada tiene el número de celular y el mensaje de alarma.

b) Recepción de SMS

Posee a su vez un Servicio Web que se comunica con la aplicación de la estación. Como parámetros de salida tiene el número de celular que envió el mensaje (previamente definido en la aplicación Web), código del actuador (ej:A5) y la operación (ON/OFF).

4.2 Diseño de la arquitectura

Para el diseño de la arquitectura se utilizó UML, principalmente los Casos de Usos, diagramas de interacción, diagramas de estado y diagramas de actividad.

4.2.1 Diagramas de casos de uso

Los actores en el sistema son cuatro y se describirán sus funciones a continuación:

- **Administrador del sistema:** Gestiona usuarios, estaciones, grupo de estaciones, reportes
- **Usuario de estación:** Puede ingresar y/o eliminar dispositivos en su estación. Crea acciones para algún evento generado por una estación específica con un sensor específico, utilizando sus actuadores y puede enviar emails. Además puede obtener información de todas las estaciones del grupo. Puede ver tanto los reportes por eventos de alarmas generadas, como los dispositivos actuadores que poseen las estaciones, pero sin accionarlos.
- **Estación:** Envía señal de alarma percibida por sus sensores al SGC. Ejecuta además las acciones que tiene registrada cuando una señal de alarma de otra estación.
- **SGC:** Broker entre estaciones, reenviando los mensajes al grupo. Guarda reporte de los hechos ocurridos.

Se detallan a continuación en la Figura 22 las acciones principales dentro del sistema:



Figura 22: Caso de uso general

Los principales escenarios sobre los que se desarrollan gran parte de la funcionalidad de la arquitectura son los siguientes:

4.2.2 Ejecución de acciones:

Cada estación, al recibir un mensaje sobre un evento de una estación con un sensor específico, debe buscar en su lista de acciones, que acción debe tomar ante tal mensaje. Cada acción puede activar uno o más actuadores y enviar uno o más emails.

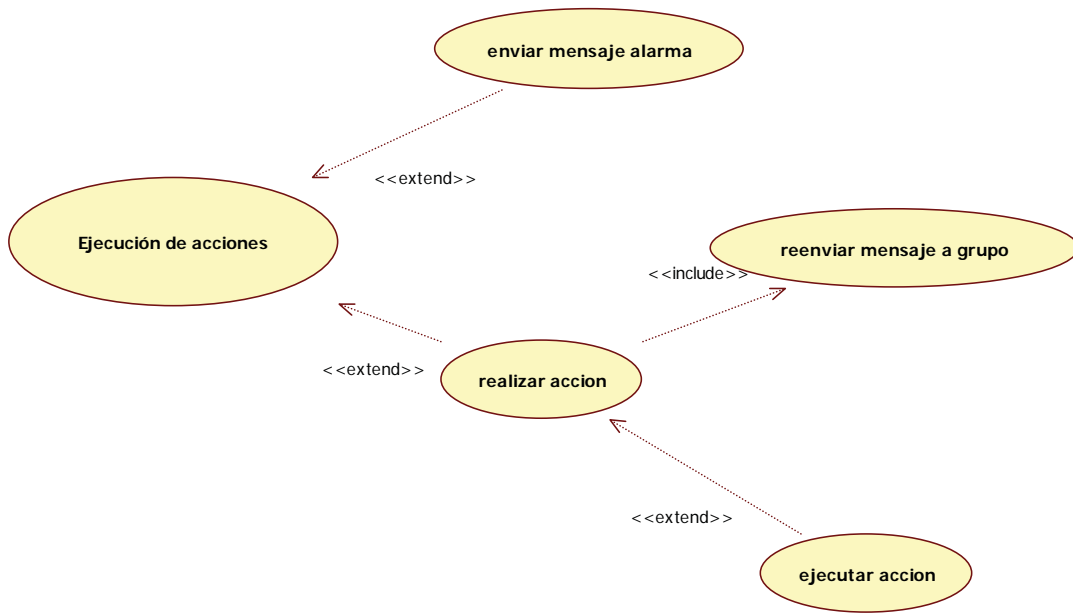


Figura 23: Caso de uso “Ejecución de acciones”

4.2.3 Gestión de acciones

Cada acción va asociada a uno o más dispositivos que se activarán cuando se reciba un mensaje (evento de alarma emitido por un sensor que ha sido activado). Una acción es configurada por un usuario de la estación. Las acciones pueden ser ingresadas, modificadas y/o eliminadas en cualquier momento.

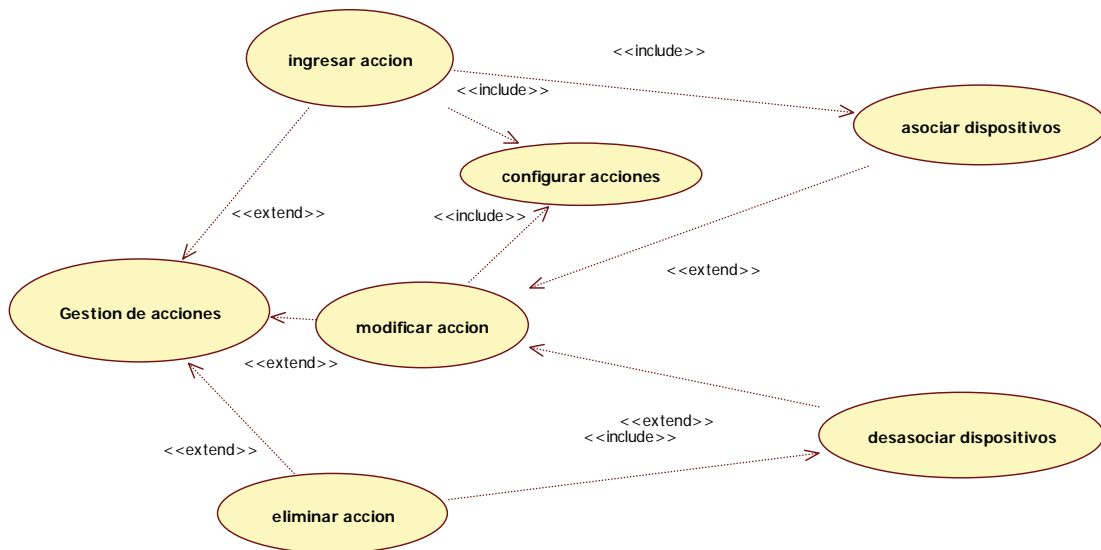


Figura 24: Caso de uso “Gestión de acciones”

4.2.4 Gestión de grupos

Cada estación debe estar asociada a un grupo. Todos los eventos de alarma son retransmitidos a los miembros del grupo y son ellos que toman las acciones. Por esto es importante ya que si una estación no pertenece a ningún grupo no podrán ejecutarse acciones remotas.

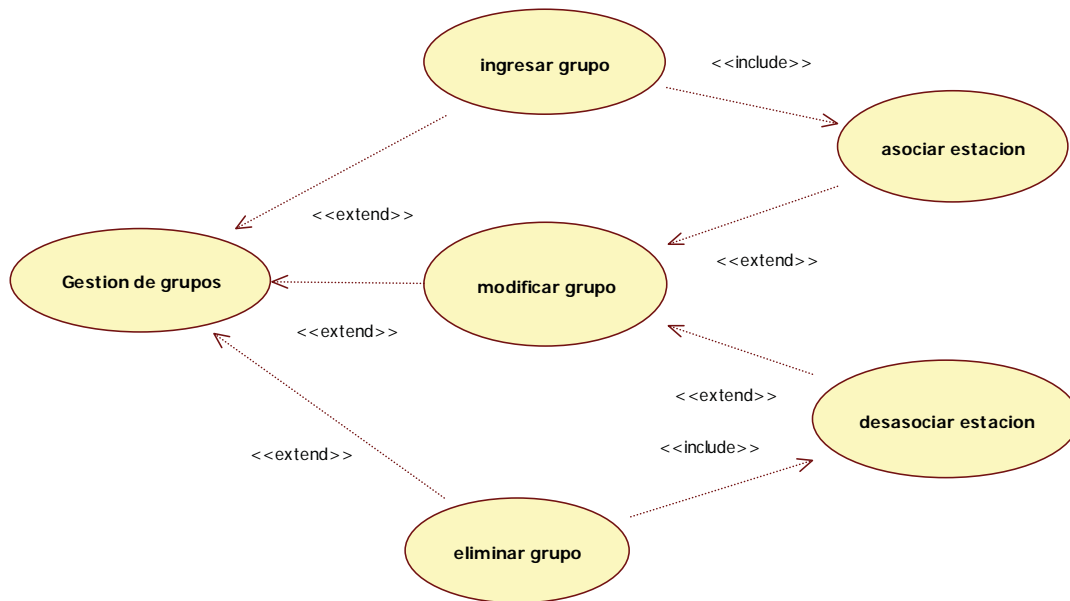


Figura 25: Caso de uso “Gestión de grupos”

4.2.5 Gestión de informes

Existen 2 tipos de informes: estado de los dispositivos de una estación y eventos ocurridos en el sistema (alarmas) los cuales pueden ser filtrados por estaciones, ubicación, fecha, hora, etc.

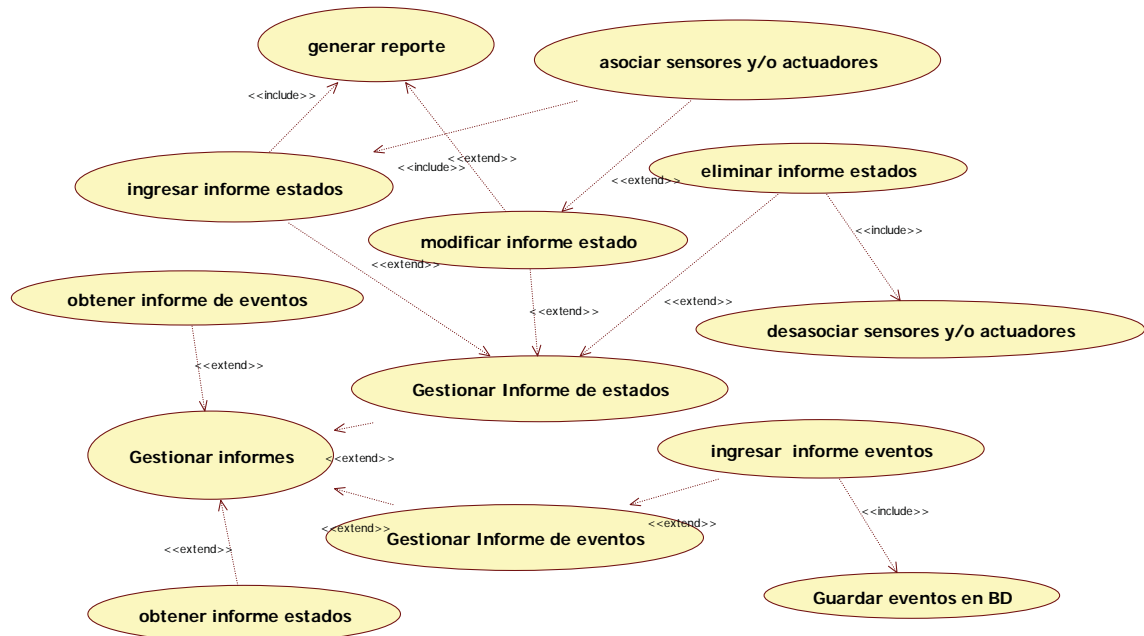


Figura 26: Caso de uso "Gestionar informes"

El sistema permite configurar reportes para que una vez que se cumplan las restricciones se envíe un mensaje mediante un email.

Ej.:

1. Enviar email cuando la estación 3 active por 3 vez de noche el sensor del patio. La información que se debería ingresar al sistema para una configuración correcta sería:

- Grupo: null (vacío)
- Estación:3
- Sensor:1 (patio)
- Cantidad:3
- Horario: N (noche)

2. Enviar email cuando en el Grupo 1 se activen 5 veces alarmas cualesquiera. La información que se debería ingresar al sistema para una configuración correcta sería:

- Grupo: 1
- Estación: null (vacío)
- Sensor: null (vacío)
- Cantidad: 5
- Horario: null (vacío)

4.2.6 Diagrama de clases

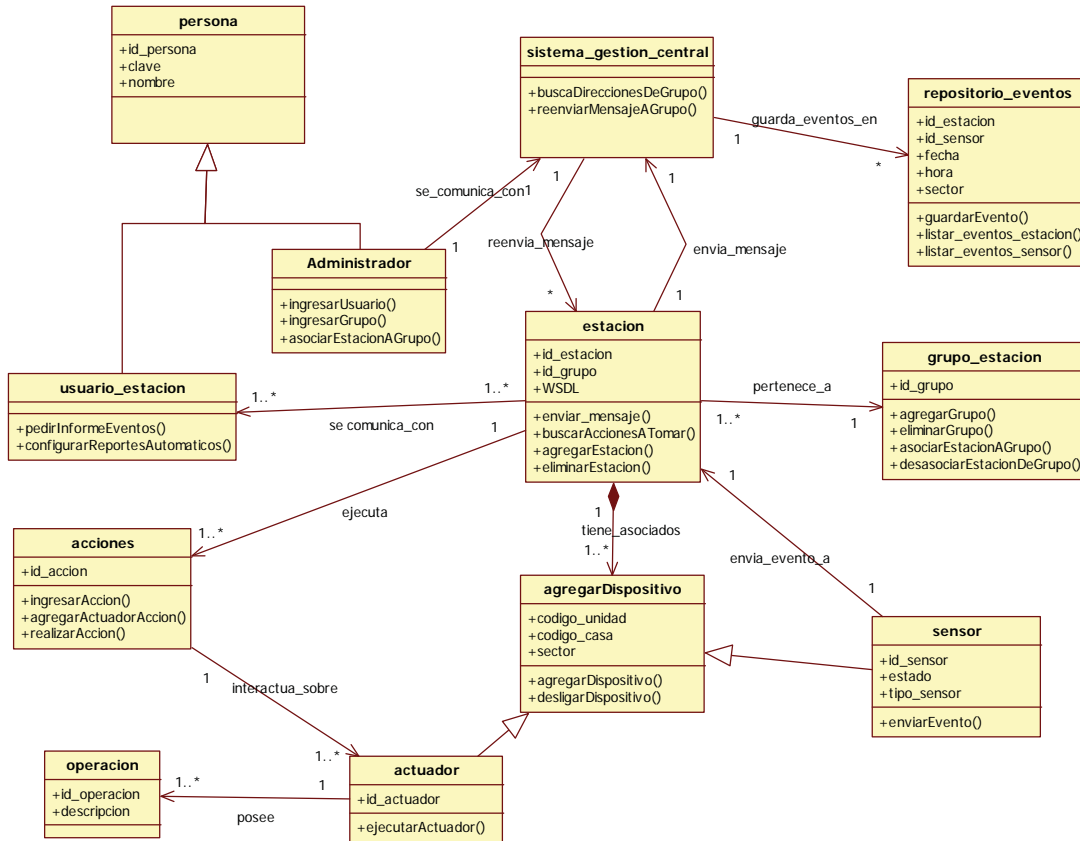


Figura 27: Modelo de datos

4.2.7 Modelo relacional de base de datos

Ambas aplicaciones estación y SGC, poseen base de datos MySql la cual se describe a continuación:

a) Estación

Cada estación lleva el control de sus dispositivos X10, sean estos sensores u actuadores y la configuración de sus acciones. Una acción puede accionar varios actuadores, puede enviar varios emails y/o enviar mensajes SMS.

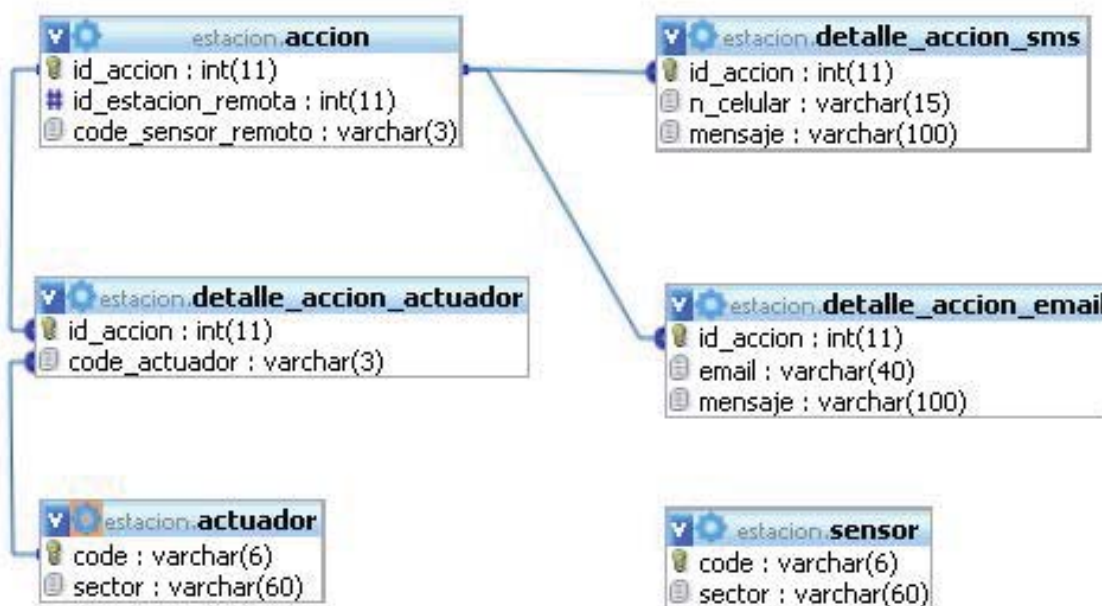


Figura 28: Modelo de base de datos de una estación

b) Servidor de Gestión Central (SGC)

El SGC lleva el control de los usuarios, estaciones y grupo dentro del sistema. Cada vez que una señal de alarma se activa, al pasar por el SGC, se almacena en reporte_evento. También es en el SGC donde los usuarios se configuran sus reportes.

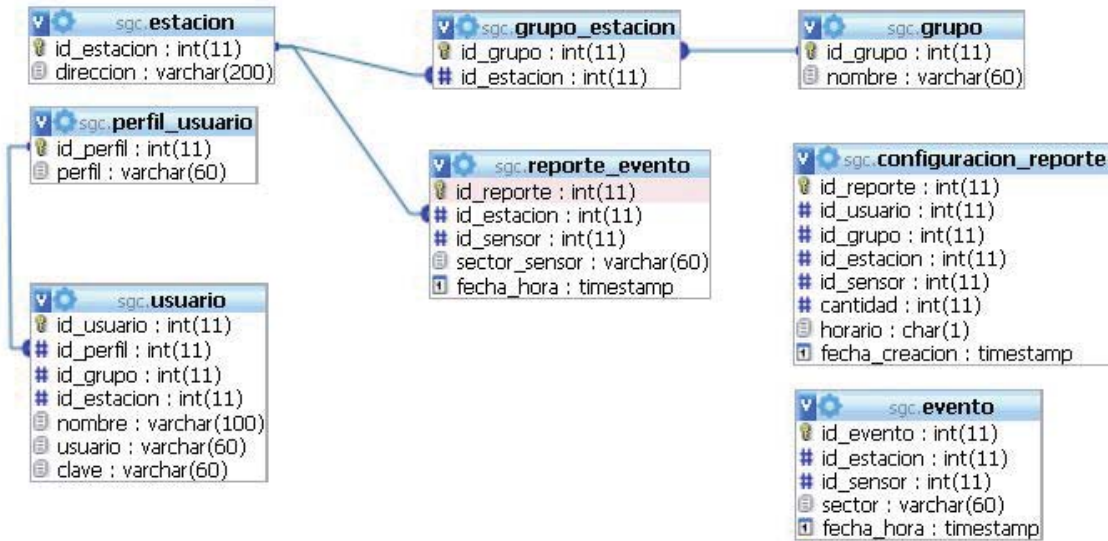


Figura 29: Modelo de base de datos del sgc

4.2.8 Diagramas de interacción

a) Envío de alarma

Todo el sistema se basa en el envío de mensajes desde una estación a las otras (pertenecientes a su grupo), por esto es importante detallar con cuidado cuales son los pasos a seguir.

A continuación se describe en un diagrama de secuencia. Se inicia con el impulso generado por el sensor al ser activado y termina con la acción que se toma y posterior confirmación.

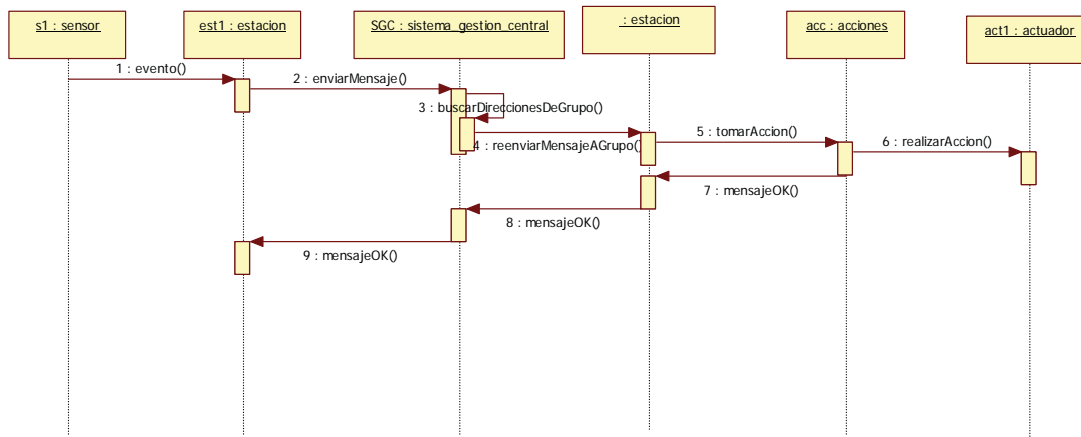


Figura 30: D. Secuencia - Envío de alarma

b) Ingreso de estación

Cada estación debe pertenecer a un grupo, es por esto que una vez ingresado al sistema, inmediatamente se le asocia un grupo.

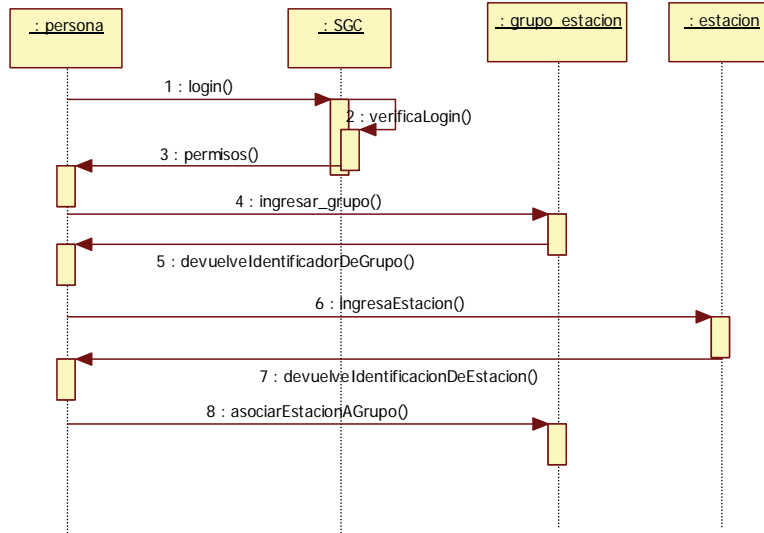


Figura 31: D. Secuencia - Ingreso de estación

4.2.9 Diagrama de colaboración evento de sensor:

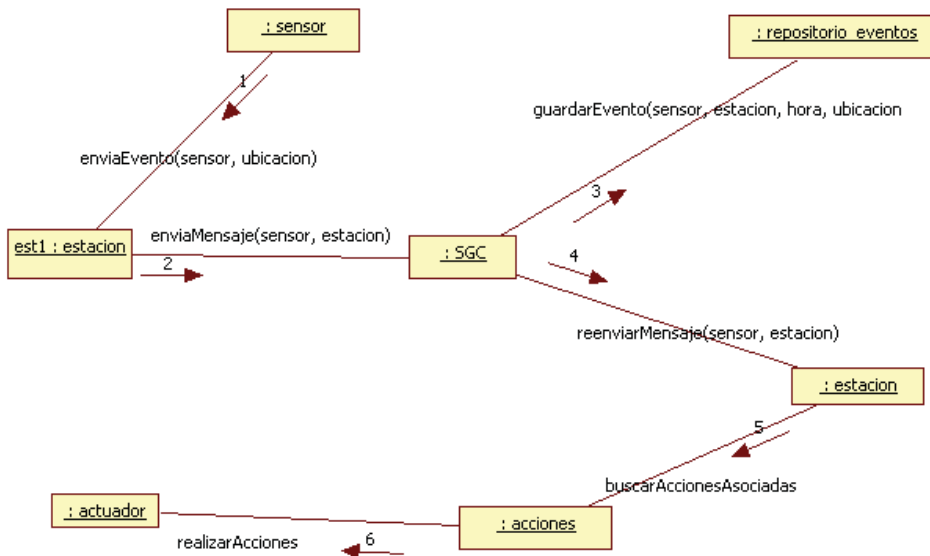


Figura 32: D. de colaboración - Evento sensor

4.2.10 Diagramas de estado

Se muestran dos diagramas que definen los estados por los que pasan las estaciones dependiendo del rol por el que están pasando, se distinguen por:

a) Estación recibe evento desde un sensor que se ha activado: Estación se encuentra en estado de espera hasta que recibe un impulso desde uno de sus sensores.

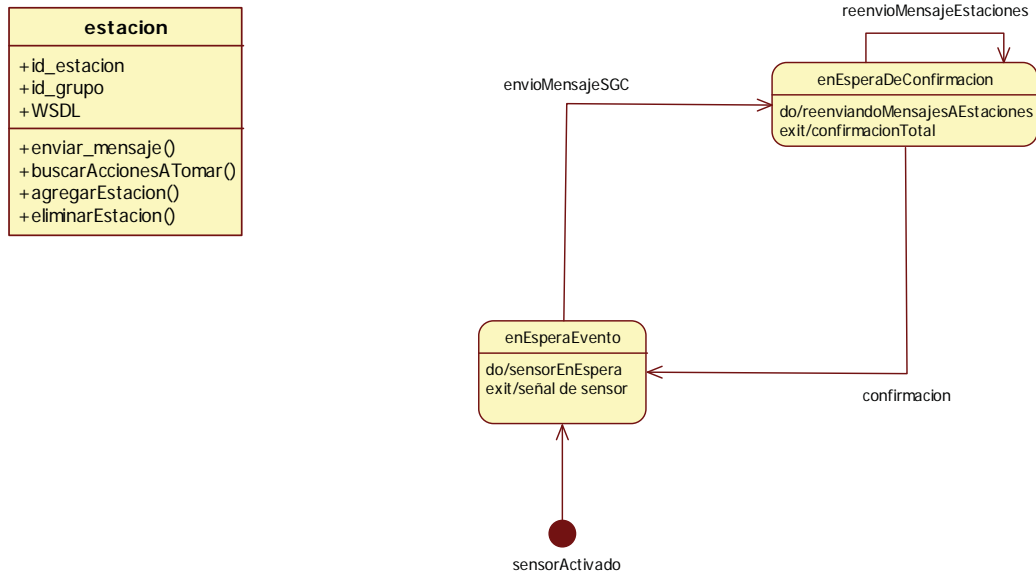


Figura 33: Diagrama de estado “Estación en espera de evento (sensor)”

b) La estación recibe un mensaje relacionado a un evento ocurrido (alarma): Debe tomar una acción relacionada al evento. Si para ejecutar una acción se necesitan mas de un sensor, se debe esperar por un tiempo limitado [ej: 30 seg.] los mensajes de la correspondiente estación que ejecuto la alarma (sensor) relacionado a otros sensores. Una vez que han llegado todos los mensajes necesarios para la ejecución de la acción (mensaje es igual a una alarma generado por un sensor).

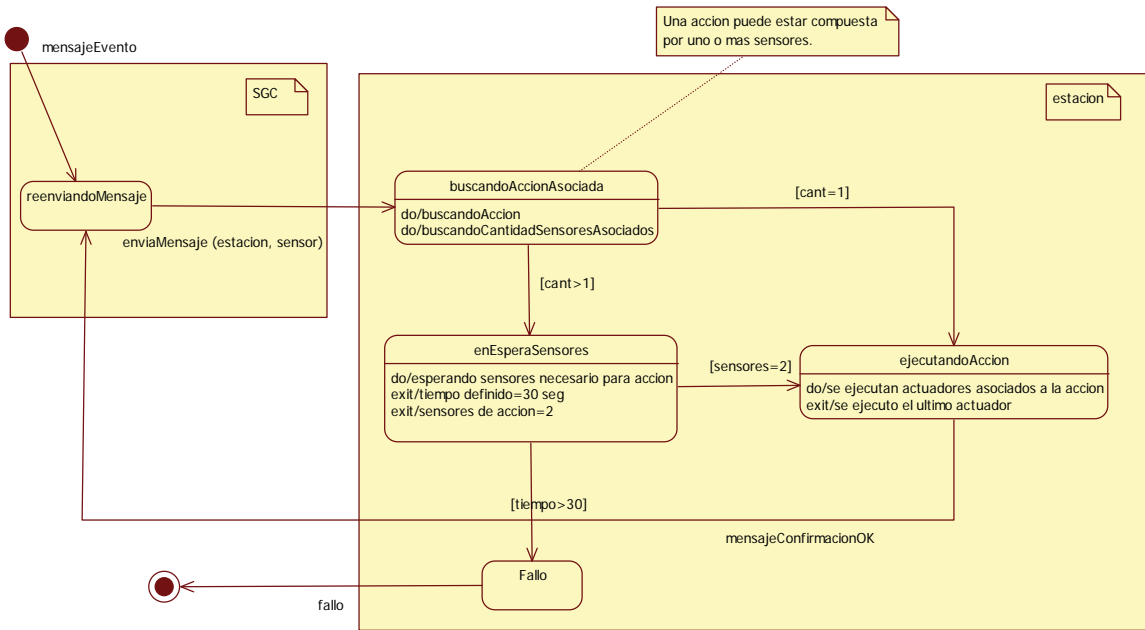


Figura 34: Diagrama de estado “Estación ejecutando acción”

4.2.11 Diagramas de actividad

a) Agregar estación:

Un administrador debe agregar una estación en el sistema, para ellos deberá loguearse, agregar la estación y luego agregarla a un grupo.

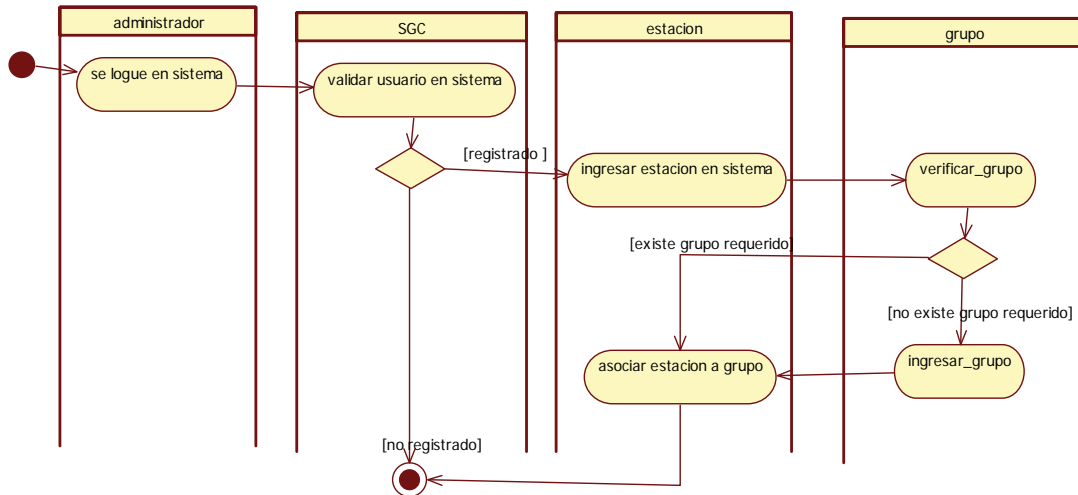


Figura 35: Diagrama de actividad “Agregar Estación”

b) Eventos Sistema de Gestión Central (SGC):

Cada vez que el SGC recibe un mensaje, debe reenviarlo a las estaciones del grupo además de guardar el evento en el repositorio de eventos.

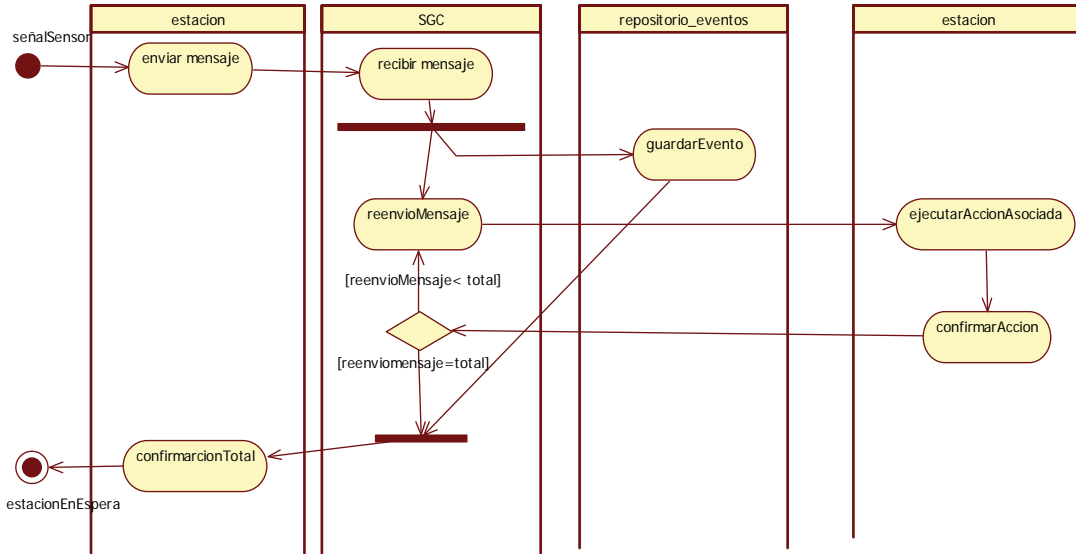


Figura 36: Diagrama de actividad “Eventos SGC”

4.3 Diseño de interfaz aplicación para Iphone

Para la gestión de las estaciones, grupos, usuarios, reportes, acciones, etc. se realizó una aplicación desarrollada en php con plugins de jquery y estilos (css) con tal de lograr una navegación fácil y fluida en dispositivos móviles como el Iphone y el Ipod Touch, que poseen un navegador muy sofisticado y vanguardista.

Para esta aplicación se utilizó principalmente el framework creado por Diego Martín L. [13] y los plugins jtouch [14] y los de groupware [15] para darle un aspecto y el diseño actual.

4.3.1 Sistema de logueo

Existen 2 perfiles de usuario: **Administrador** y **Usuario de estación**



Figura 37: Sistema de Logueo

4.3.2 Menú de usuario

Las opciones del menú son diferenciados según el usuario logueado. Mientras el Administrador se encarga de crear los nuevos grupos, estaciones y usuarios, el Usuario de estación se encarga de asignar los dispositivos al sistema que previamente fueron conectados a la corriente eléctrica y de seteados de manera física en el dispositivo el código (House code+Unit code)



Figura 38: Menú Administrador



Figura 39: Menú Usuario

4.3.3 Formulario de ingreso de dispositivo

En la misma interfaz de ingreso de dispositivos a la estación, se da la opción de revisar los actuadores y sensores que actualmente tiene ingresados la estación.



Figura 40: Formulario de ingreso de nuevo dispositivo

4.3.4 Mensaje de ingreso exitoso

Una vez que se ha ingresado el dispositivo se despliega el mensaje de “Ingresado exitosamente”



Figura 41: Nuevo actuador ingresado (código A3)

4.3.5 Listado de dispositivos y prueba de actuadores

Una vez que se ingresa el dispositivo se listan para comprobar que todo este bien. Desde ese mismo listado se pueden probar los dispositivos haciendo clic a la fila donde aparece el dispositivo.



Figura 42: Opciones de prueba de actuador

Las figuras siguientes ilustran el uso de la aplicación luego de haber ingresado el actuador al sistema y ejecutar la opción “Encender”.



Figura 43: Lámpara antes de encenderla remotamente



Figura 44: Lámpara encendida remotamente

4.4 Herramientas y entorno de desarrollo

Las aplicaciones fueron desarrolladas en lenguaje Java y PHP, para ello se utilizaron las siguientes herramientas:

- NetBeans IDE: Este IDE facilita la generación del código JAVA, el código JAX-WS para consumir y publicar los servicios Web, así como la posibilidad de generar los enlaces XML mediante JAXB.
- JAX-WS: El desarrollo del código basado en Java y XML para la obtención del servicio Web está basado en este modelo de programación.
- Apache Tomcat: Se utilizó tomcat como servidor de aplicaciones para la aplicación escrita en Java.
- MySQL: Como motor de Base de datos para las aplicaciones tanto en Java como en PHP.
- NuSoap: Librería php para la generación y consumo de Servicios Web. El paso de mensajes entre las aplicaciones se hace enteramente a través de los Servicios Web generados por JAX-WS y NuSoap, pudiendo utilizarse otro sin ningún problema.

4.5 Implementación

A continuación describirán las partes del código que se han considerado más importante para entender el funcionamiento de la aplicación:

4.5.1 Aplicación Web del SGC, desarrollada en lenguaje PHP.

- Método encargado de ejecutar una instrucción de encendido o apagado:

```
<?
require_once('../php/nusoap/nusoap.php');

function instruccionX10($WSDL, $operacion, $code){
    $client = new nusoap_client();
    $parametros = array('operacion' => $operacion, 'code' => $code);
    $result = $client->call('instruccion', $parametros , "http://servicios.sis/");
}
instruccionX10("http://proyectox10.dyndns.org:8080/estacion/opcionesDispositivosService?WSDL",
    $_POST['operacion'], $_POST['code']);
?>
```

Se realiza una llamada mediante un cliente Web Service a la estación cuya WSDL se conoce.

- Método para obtener el listado de los dispositivos Actuadores/Sensores

```
<?php
require_once('../..../php/nusoap/nusoap.php');
$direccionEstacion="http://proyectorx10.dyndns.org:8080/estacion/opcionesDispositivosService?WSDL";
$tipo=$_POST['tipo'];

function listarDispositivosEstacion ($direccionEstacion, $tipo){
    $client = new nusoap_client($direccionEstacion);
    $parametros = array('tipo' => $tipo);
    $result = $client->call('obtener', $parametros , "http://servicios.sis/");
    if(isset($result))
        foreach($result as $x=>$y)
            if(isset($y["item"]))$arr[]=$y["item"];
    return $arr;
}

$arr=listarDispositivosEstacion ($direccionEstacion, $tipo);
```

Se debe indicar si se quieren listar los actuadores o los sensores.

4.5.2 Aplicación de la estación, desarrollada en lenguaje java.

- Métodos mas relevantes para las acciones sobre los dispositivos X10

```
public void encender(String Modulo) {
    comando = new Command(Modulo, Command.ON);
    controller.addCommand(comando);
}
public void apagar(String Modulo) {
    comando = new Command(Modulo, Command.OFF);
    controller.addCommand(comando);
}
public void aumentarBrillo(String Modulo,int nivel) {
    comando = new Command(Modulo, Command.BRIGHT, nivel);
    controller.addCommand(comando);
}
public void disminuirBrillo(String Modulo,int nivel) {
    comando = new Command(Modulo, Command.DIM, nivel);
    controller.addCommand(comando);
}
```

- Método Utilizado para ingresar un dispositivo a la BD de la estación. Método invocado por el cliente de la aplicación del SGC.

```
public int ingresarDispositivo(int tipo, String code, String sector) throws
SQLException{
    Conexion c = new Conexion();
    System.out.println(c.Conectar("jdbc:mysql://localhost/estacion","root", ""));
    Statement st = (Statement) c.getConnection().createStatement();
    ResultSet rs;
    String consulta = null;
    if(tipo==1){ // sensores
        consulta="INSERT INTO sensor (code, sector) values('"+code+"',
'"+sector+"')";
    }else{ //actuadores
        consulta="INSERT INTO actuador (code, sector) values('"+code+"',
'"+sector+"')";
    }
}
```



```

    st.execute(consulta);
    return 1;
}

```

- Método encargado de ejecutar el encendido o apagado de los dispositivos

```

public void controlX10(int opcion, String cod){
    if(opcion==1) this.apagar(cod);
    else         this.encender(cod);
}

```

- Método que llama a controlx10 para encender o apagar dispositivos.

```

public int comandoX10(int operacion, String code){
    Servidor xx=new Servidor();
    xx.controlX10(operacion, code);
    return 1;
}

```

Algunos Web Services importantes de la estación:

```

package sis.servicios;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.jws.WebMethod;
import javax.jws.WebParam;
import javax.jws.WebService;
import modulos.dispositivos;

```

- Recibe parámetros para apagar o encender un dispositivo

```

@WebMethod(operationName = "instruccion")
public int instruccion(@WebParam(name = "code")
String code, @WebParam(name = "operacion")
int operacion) {
    dispositivos disp =new dispositivos();
    return disp.comandoX10(operacion, code);
}

```

- Encargado de recibir los datos para ingresar un nuevo dispositivo

```

public int ingresar(
    @WebParam(name = "tipo") int tipo,
    @WebParam(name = "code") String code,
    @WebParam(name = "sector")String sector) {
    dispositivos disp =new dispositivos();
    int valor=0;
    try {
        valor=disp.ingresarDispositivo(tipo, code, sector);
    } catch (SQLException ex) {
        Logger.getLogger(opcionesDispositivos.class.getName()).log(Level.SEVERE, null,
ex);
        valor=3;
    }
    return valor;
}

```

- Devuelve el listado de los dispositivos del tipo “**tipo**” (1:Sensor, 2:Actuador)

```
@WebMethod(operationName = "")
public String[][] obtener(@WebParam(name = "tipo") int tipo){
    dispositivos disp =new dispositivos();
    String[][] est= new String[10][2];
    try {
        est=disp.obtenerDispositivos(tipo);
    } catch (SQLException ex) {

        Logger.getLogger(opcionesDispositivos.class.getName()).log(Level.SEVERE, null,
        ex);
    }
    return est;
}
```

4.5.3 Gateway SMS

Para la recepción y envío de mensajes SMS se utilizo un Gateway SMS de pago escrito en PHP, que posee los métodos necesarios para la comunicación.

Existen de todas maneras librerías en JAVA que permiten en el envío del SMS.

Un teléfono celular provee al gateway de señal para en envío y recepción de mensajes. Este gateway almacena una copia de los mensajes recibidos por el celular y envía mensajes a través de este mismo.

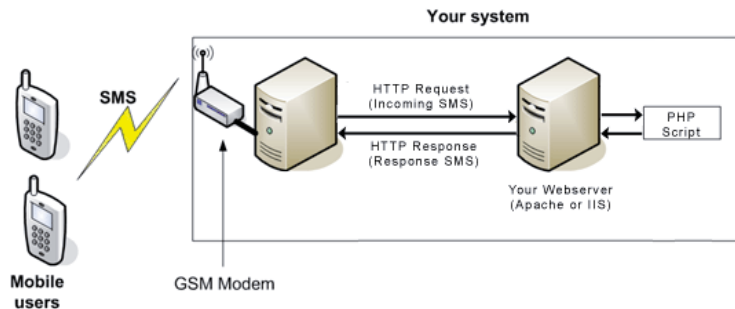


Figura 45: Como recibir un SMS desde un sitio Web

Como MODEM GSM se utilizo un celular Sony Ericsson W580. La conexión al PC fue a través de cable USB, pero también podría haber sido por puerto serie o por BlueTooth.

- **Recepción de SMS**

En una página, definida por la aplicación del Gateway, se reciben los parámetros de: Nº de teléfono y Mensaje SMS. Estos parámetros se envían a la aplicaron de la estación quien los procesa y ejecuta acciones según corresponda.

La comunicación es a través de un servicio Web como se muestra a continuación.

```
<?php
```

```
$sender = $_GET['sender'];  
$message = $_GET['msgdata'];  
require_once('nusoap/nusoap.php');  
list($id_estacion, $code_sensor) = split(" ", $message);  
$client = new nusoap_client($direccionEstacion);  
$parametros = array('id_estacion' => $id_estacion, 'code_sensor' => $code_sensor);  
$result = $client->call('ejecutar_alarma', $parametros , "http://servicios/");
```

```
?>
```

4.6 Implantación

En esta arquitectura se utilizaron interfaces que utilizan el puerto serie para la comunicación con el PC, pero existen en el mercado actualmente también interfaces que utilizan el puerto usb.

A continuación se describen 2 librerías que permiten la comunicación desde una aplicación java con el puerto serie.

4.6.1 COMM

La comunicación por puerto serie, permite al ordenador comunicarse con todo tipo de dispositivos periféricos: módems, impresoras, escáneres, lectores de código de barras, etc. El API de Comunicaciones Java, constituido por el paquete javax.comm, que proporciona JavaSoft, no forma parte del JDK, pero añade soporte a Java para dispositivos serie y paralelo.

El paquete proporciona soporte para dispositivos serie y paralelo al estilo Java, es decir, utilizando una semántica semejante a la que se usa con streams y eventos. Para comunicarse con un dispositivo serie a través de uno de los puertos serie de un ordenador, desde una aplicación Java o un applet, es necesario un interfaz. El API de Comunicaciones Java, permite transmitir y recibir datos a través de dispositivos conectados al puerto serie; proporcionando además un conjunto de opciones que permiten la configuración de todos los parámetros asociados a los puertos serie y paralelo. Este API es una proposición para establecer un método estándar de acceso a los puertos de comunicaciones, que permita escribir programas Java independientes de la plataforma. Así se pueden escribir programas para emulación de terminales, programas de fax, lectores de tarjetas, etc.

El desarrollo de buenos programas normalmente pasa por construir unos cuantos interfaces claramente definidos. El diagrama de alto nivel de las capas que componen el API de comunicaciones Java es el que se reproduce en la siguiente Figura:

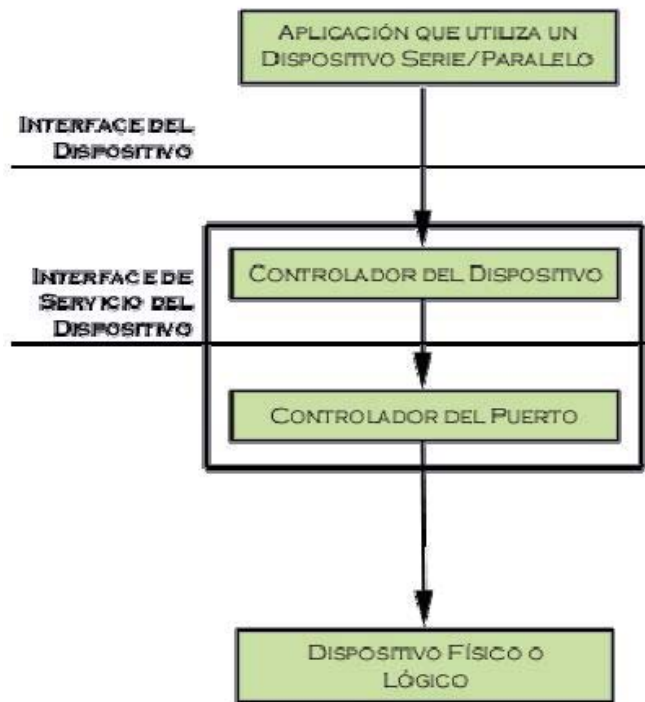


Figura 46: Capas del API

El diagrama que muestra la Figura siguiente describe los objetos involucrados en la lectura o escritura de datos en un puerto serie desde Java.

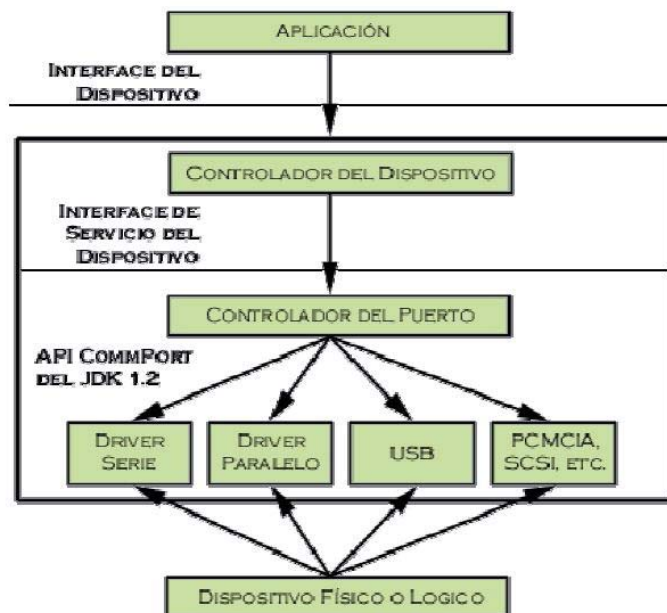


Figura 47: Lectura/Escritura

4.6.2 RXTX

Tienen exactamente la misma funcionalidad que `javax.comm`. Es una biblioteca que provee los métodos necesarios y la interfaces para la comunicación tanto por el puerto paralelo como por el puerto serie.

Sun ya no da soporte para windows por lo que es una buena opción utilizar la librería `rxtx`.

Para utilizar esta librería en aplicaciones que estén utilizando **comm**, solo se debe reemplazar en el proyecto donde dice `javax.comm` por `gnu.io`.

Instalación de `rxtxSerial` (www.rxtx.org) en Windows:

- copiar **rxtxSerial.dll** a `[JDK-directory]\jre\bin\rxtxSerial.dll`
- copiar **RXTXcomm.jar** a `[JDK-directory]\jre\lib\ext\RXTXcomm.jar`

Este driver es el que finalmente se usa en el sistema para la comunicación a través del puerto serie.

5 Casos de estudio

Con el fin de utilizar de forma práctica la arquitectura, se detallarán dos escenarios de prueba. Estos escenarios intentarán utilizar al máximo las opciones denotadas con anterioridad.

Cabe destacar que pudieran fácilmente crearse muchos escenarios más según el caso en que fuese necesario.

Los escenarios son:

- Ejecución de acciones remotas
- Alarma por sensor de movimiento

5.1 Ejecución de acciones remotas

Encender una lámpara a través de la aplicación Web optimizada para el Iphone y a través de un mensaje de texto.

Implementos necesarios para el escenario:

- **Firecracker (cm17a):** Interfaz de comunicación entre el PC y el Trasmisor RF X10.
- **Transmisor/Receptor RF X10:** Encargado de recibir las señales inalámbricas X10 e inyectarlas en la corriente eléctrica.
- **Ipod Touch:** Se utilizara su navegador para acceder a la aplicación Web mediante WIFI.
- **Gateway SMS:** Utilizado para enviar y recibir mensajes SMS
- **Celular GSM:** Se utilizará para dotar al Gateway SMS de conexión celular. Normalmente en vez de un celular se utiliza un MODEM GSM.
- **Modulo de lámpara:** Modulo que activara la lámpara una vez que reciba una señal.
- **Lámpara:** Se conectara al modulo de lámpara X10.

5.1.1 Ejecutar instrucción a través de aplicación Web

Desde el Ipod se accede al sistema Web, que previo logueo lista los dispositivos que puede controlar de la estación. Al usuario solo se le listarán los actuadores que posea su estación particular.

En el siguiente diagrama de identifican las partes que interactuaran para la consecución final de una lámpara encendida.

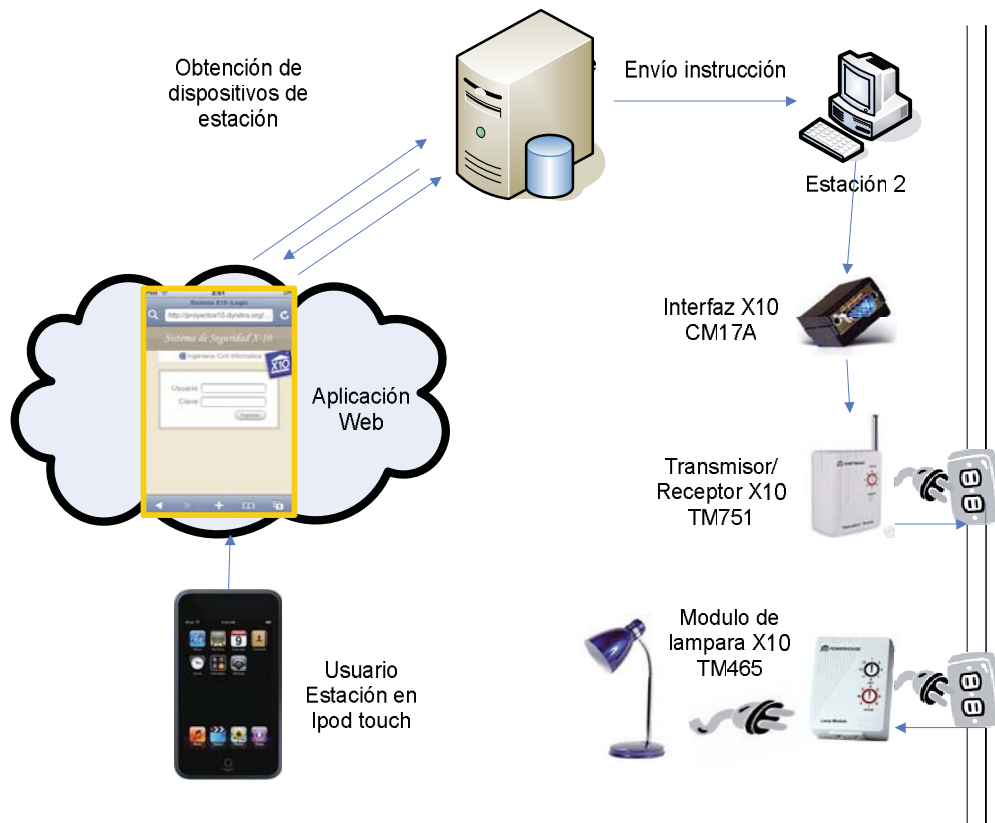
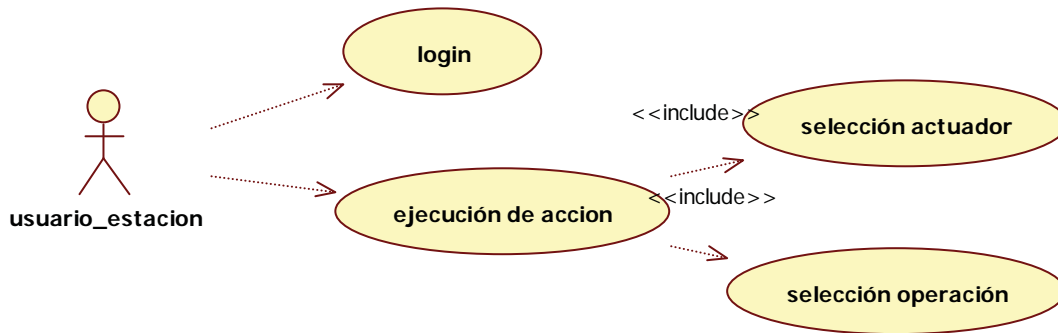


Figura 48: Diagrama despliegue “Escenario de prueba 1a”

- Caso de uso



Caso de Uso: Operación sobre actuador mediante WEB	
Actor principal: Usuario de estación	
Personal Involucrado e intereses	
<ul style="list-style-type: none"> - Usuario: Envía ingresa a la aplicación - SGC: Lista dispositivos estación y envía mensaje a estación receptora. - Estación: Ejecuta operación ON/OFF sobre actuador 	
Precondiciones: Es usuario debe estar registrado para esa estación en el SGC	
Garantías de éxito (Post condiciones): Se ejecuta la operación sobre el actuador.	
Escenario Principal de éxito	
Paso	Acción
1	Usuario ingresa al sistema mediante logueo
2	Ingresa a opción de operaciones sobre actuadores
3	El SGC lista todos los dispositivos actuadores de su estación
4	Usuario selecciona el actuador y la operación (ON/OFF)
5	Estación ejecuta la operación sobre el actuador
Extensiones	
Paso	Acción
1	Usuario no registrado en ninguna estación. Es botado del sistema.
3a	No existen actuadores asociados. Usuario sale del sistema
3b	No existen actuadores asociados. Usuario asocia actuadores.
5	El actuador no existe. No se ejecuta operación

Tabla 3: Caso de uso "Operacion sobre actuador mediante Web"

• Diagrama secuencia

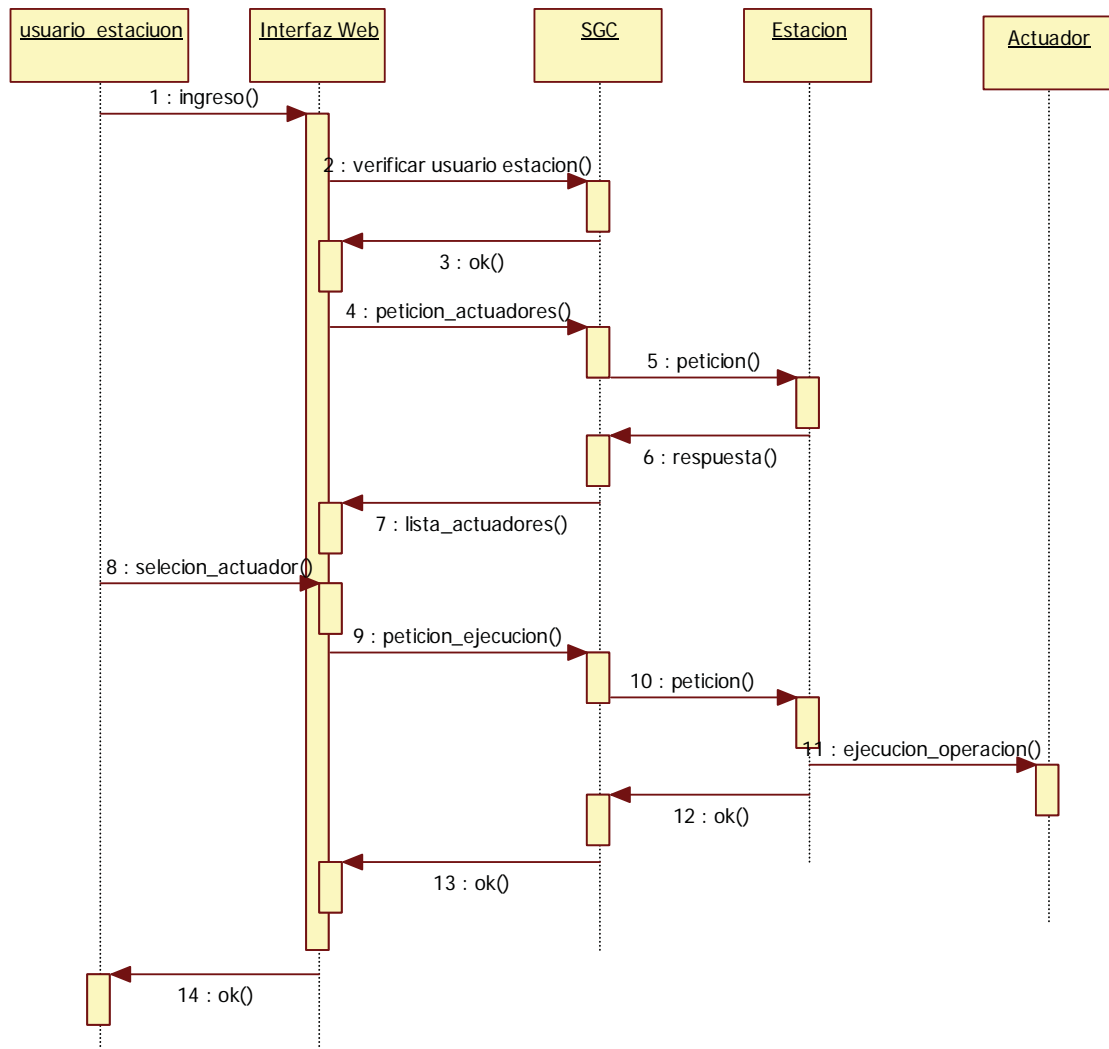


Figura 49: Diagrama de secuencia "Escenario de prueba 1a"

5.1.2 Ejecutar aplicación a través de mensaje de texto (SMS)

El usuario de una estación, previamente ingresado a la estación mediante la interfaz para el Iphone crea un mensaje de texto. El mensaje debe contener:

- ID de estación
- Código de actuador

La estación receptora valida el número de celular antes de realizar la operación sobre el actuador.

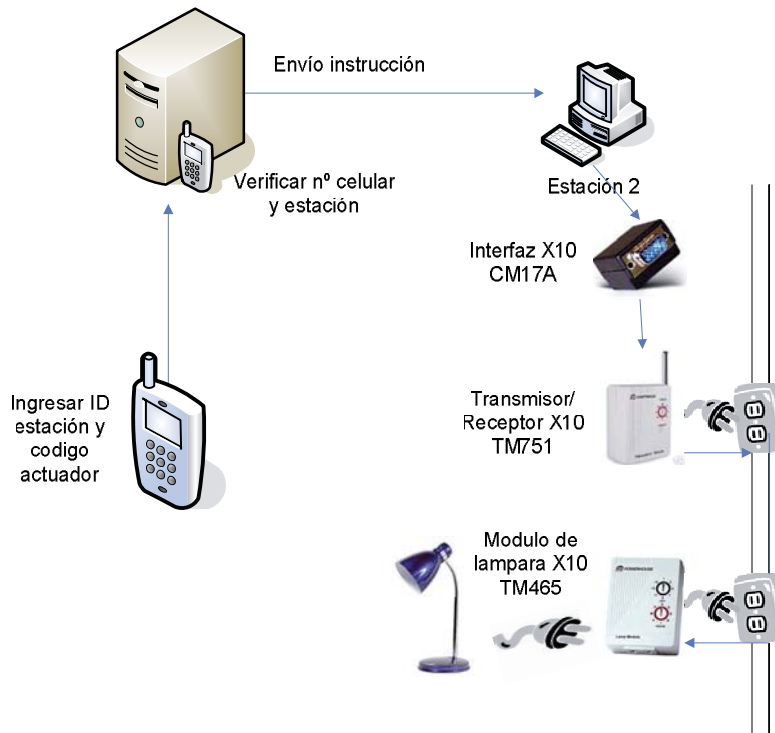
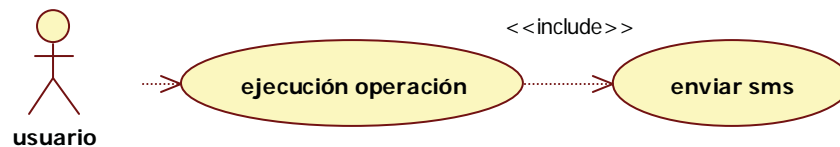


Figura 50: Diagrama de despliegue "Escenario de prueba 1b"

- Caso de uso



Caso de Uso: Operación sobre actuador mediante SMS	
Actor principal: Usuario de estación	
Personal Involucrado e intereses	
<ul style="list-style-type: none"> - Usuario: Envía mensaje se texto a Gateway SMS. - Gateway: Recibe SMS y lo reenvía mediante WS a estación receptora - SGC: Indica a Gateway Estación receptora. - Estación Receptora: Ejecuta operación ON/OFF sobre actuador 	
Precondiciones: El N° celular debe estar registrado en la estación	
Garantías de éxito (Post condiciones): Se ejecuta la operación sobre el actuador.	
Escenario Principal de éxito	
Paso	Acción

1	El usuario envía un mensaje de texto, explicitando la estación y el código del actuador
2	El Gateway recibe el mensaje y pide a SGC que le envíe dirección de estación remota
3	El SGC recibe el identificador de la estación y le devuelve al Gateway el WSDL de la estación.
4	Gateway se comunica mediante WS con la estación receptora
5	Estación receptora verifica número celular y ejecuta la operación sobre el actuador
Extensiones	
Paso	Acción
3	Identificador de estación no válido. No se ejecuta la operación.
5a	Nº de celular no registrado en estación: No se ejecuta la operación.
5b	No existe actuador asociado a ese código. No se ejecuta la operación.

Tabla 4: Caso de uso "Operación sobre actuador mediante SMS"

- Diagrama de secuencia

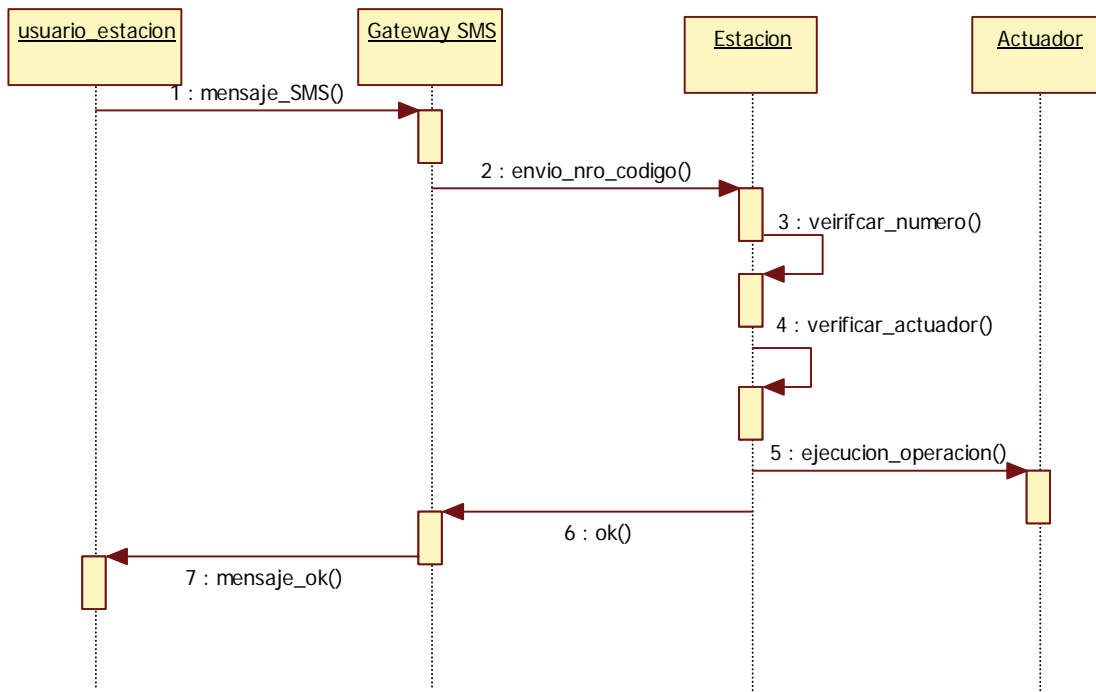


Figura 51: Diagrama de secuencia "Escenario de prueba 1b"

5.2 Alarma por sensor de movimiento

Para que una alarma genere una operación remota, es necesario configurarla. Para esto es necesario que el usuario de la estación ingrese a la aplicación Web y la configure. Debe configurar el sensor y luego crear las acciones que se realizarán en la estación remota.

En este escenario la acción será:

- Encender una lámpara
- Enviar un correo electrónico (email)
- Enviar un mensaje de texto (SMS)

Implementos necesarios para el escenario:

- **Firecracker (cm17a):** Interfaz de comunicación unidireccional entre el PC y el Transmisor RF X10.
- **CM19A:** Interfaz de comunicación bidireccional. Se utilizará para recibir las señales RF del sensor de movimiento X10.
- **Transmisor/Receptor RF X10:** Encargado de recibir las señales inalámbricas X10 e inyectarlas en la corriente eléctrica.
- **Gateway SMS:** Utilizado para enviar y recibir mensajes SMS
- **Celular GSM con conexión usb:** Se utilizará para dotar al Gateway SMS de conexión celular. Normalmente en vez de un celular se utiliza un MODEM GSM.
- **Sensor de movimiento X10:** Recibirá el estímulo que desencadenará la acción remota.
- **Modulo de lámpara:** Modulo que activara la lámpara una vez que reciba una señal.
- **Lámpara:** Se conectara al modulo de lámpara X10.

A continuación se identifica el problema con un diagrama general:

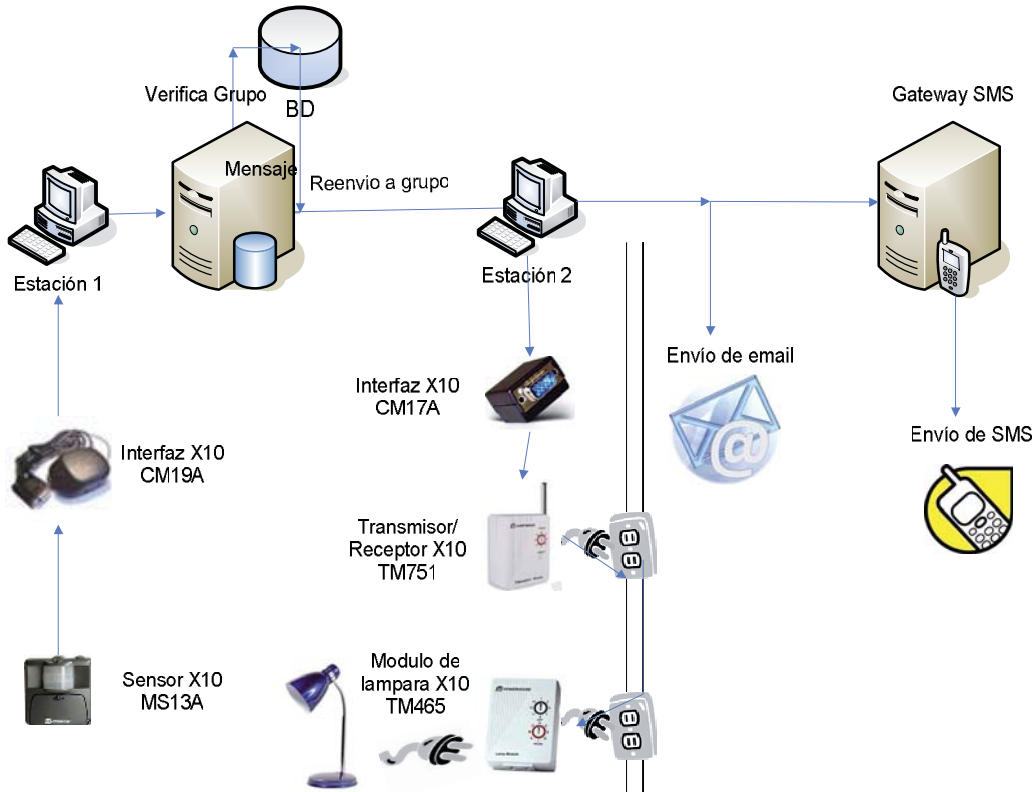
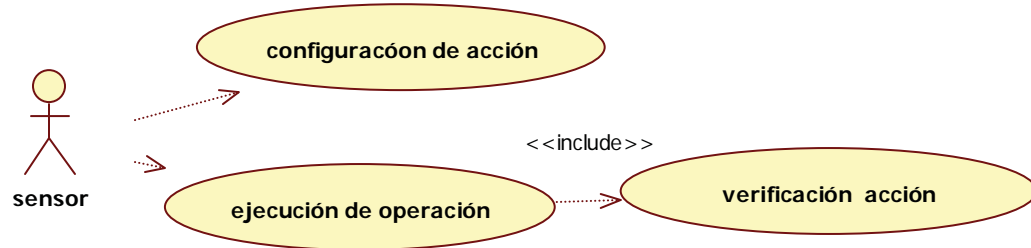


Figura 52: Diagrama de despliegue “Escenario 2”

- Caso de uso



Caso de Uso: Ejecutar Acción mediante sensor de movimiento
Actor principal: Sensor de movimiento
Personal Involucrado e intereses
<ul style="list-style-type: none"> - Sensor: Envía señal a su estación en caso de detectar movimiento. - Estación emisora: Envía mensaje a SGC - SGC: Reenvía mensaje a las estaciones pertenecientes al grupo de la estación emisora. - Estación Receptora: Ejecuta acciones asociadas al evento de alarma
Precondiciones: La estación debe tener registrado el sensor.

Garantías de éxito (Post condiciones): Se ejecutan las acciones preconfiguradas.	
Escenario Principal de éxito	
Paso	Acción
1	El sensor detecta movimiento y envía señal a estación.
2	La estación envía un mensaje al SGC que contienen (código sensor, id_estacion)
3	El SGC reenvía el mensaje a todas las estaciones pertenecientes al grupo de la estación inicial.
4	Web services de estación recibe mensaje y ejecuta acciones asociadas.
Extensiones	
Paso	Acción
1	Estación no tiene registrado el sensor. La señal se pierde.
3	La estación no esta asociada a ningún grupo. No se reenvía el mensaje. No se ejecuta acción.
4	No existe una acción asociada al sensor. No se ejecutan acciones.

Tabla 5: Caso de uso "Ejecutar acción"

- Diagrama de secuencia

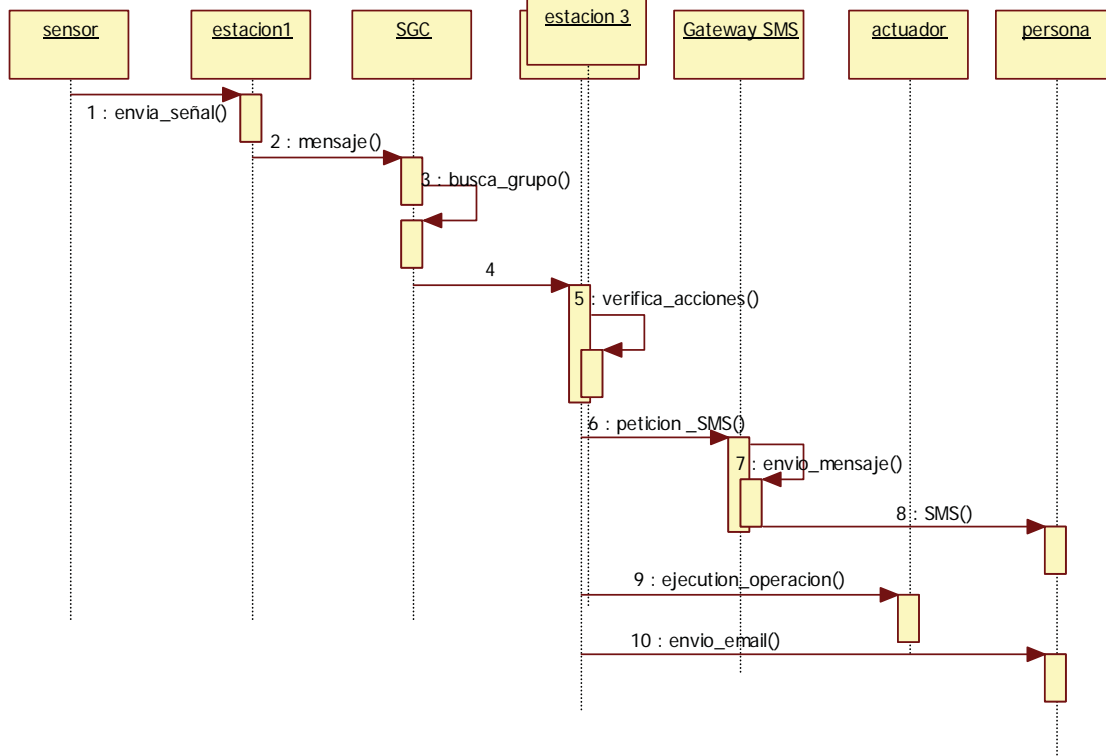


Figura 53: Diagrama Secuencia "Ejecutar acción"

6 Conclusiones

El uso de las tecnologías se ha hecho tan familiar para el ser humano que se las utiliza en casi todos los aspectos de la vida. Que en la mayoría de los hogares exista un computador o un celular no es una novedad sin embargo, la automatización de ciertos procesos básicos dentro de nuestro hogar, como apagar las luces si no hay nadie presente en una habitación, o encender la calefacción cuando estemos por llegar a nuestro hogar luego de un viaje, no es algo tan común, por lo menos en Chile.

En este trabajo se presenta una arquitectura domótica que implementa dispositivos que pueden dotar de cierta inteligencia a nuestro hogar o lugar de trabajo, configurando ciertas acciones ante eventos esperados, con el fin de lograr seguridad y/o confort. En el mercado existen variadas alternativas para lograr esta inteligencia, pero se optó por utilizar dispositivos basados en el protocolo X10 bajos costos, modularidad, gran variedad de proveedores y lo más importante, que utiliza el sistema de red eléctrica ya existente, por lo que hace que sea una solución muy asequible para la mayoría de las personas.

La arquitectura comprende la comunicación entre distintas viviendas, sucursales y/o estaciones de trabajo, mediante el envío de mensajes a través de Web Services que utilizan protocolos de comunicación estándares de Internet. La configuración de las acciones sobre actuadores, la configuración de los sensores y el mensaje transmitido a través de las distintas partes del sistema se realiza sobre un entorno Web accesible en todo momento, previa autenticación, en una interfaz optimizada para dispositivos tan actuales, como lo es en este caso particular el Iphone.

A través de este proyecto se probó que es posible crear una arquitectura utilizando tecnologías basadas en Web Services y productos hardware y software de fácil adquisición, rápida implementación, sencilla instalación y libres, como lo son los lenguaje php y java, y al uso del protocolo y dispositivos X10

La arquitectura presentada en este proyecto, utiliza un segmento acotado de las posibilidades que brindan los dispositivos X10 y la utilización de los Web Services para acceder a estos dispositivos remotamente, sin tener la intención de dejar este tema por cerrado, existen otros dispositivos X10 y otras formas de comunicación entre ellos. En este proyecto se trataron dispositivos cuya comunicación al computador se realiza mediante el puerto serial existiendo actualmente otros dispositivos que se conectan a través del puerto USB o que logran comunicación entre ellos por medio de la tecnología inalámbrica bluetooth.

No me cabe la menor duda de que X10 seguirá adoptando las nuevas tecnologías que se descubran debido a que por no ser X10 un protocolo privativo existen muchos proveedores de estos dispositivos.

En muchos países la utilización de protocolos e infraestructuras domóticas es ampliamente utilizada tanto en el sector comercial (edificios inteligentes), como en el hogar, pero en Chile están aún en un grado muy incipiente. Propósito de este estudio es el angostar un poco mas la brecha digital que aún existe, dando una mirada más de cerca de tecnología actuales en este ámbito e instando a su utilización ya que en un futuro cercano estas tecnologías dejarán de ser un lujo y formarán parte de nuestra cotidianidad.

Referencias

- [01] X10 Powerline Carrier (PLC) Technology, <http://www.x10.com/support/technology1.htm>, visitada el 10-02-2008
- [02] Smart Home Systems, <http://www.smarthomeusa.com/info/x10theory/#theory>, visitada el 25-07-2008
- [03] Claudio Devia, Diego Rey, Automatización del hogar mediante tecnología x10, 2005. Memoria para obtener el título de Ingeniero en ejecución informática, Universidad Católica de Valparaíso.
- [04] Wei Jung Shih hung, Domótica aplicada en residencias con énfasis en ínter conectividad, diseño y análisis de sus ventajas y desventajas, Julio 2005
- [05] Tarrini L., Miori V., An XML Web Services mobile interaction solution for X-10 Powerline Networking, Technical Report 2005,
- [06] Web Service Architecture, W3C Working Group Note 11 February 2004, <http://www.w3.org/TR/ws-arch/>, visitada el 12-08-2008
- [07] W3C (World Wide Web Consortium), WSDL standard <http://www.w3.org/TR/wsdl20/>, World Wide Web Consortium, visitada el 14-08-2008
- [08] W3C (World Wide Web Consortium), UDDI (Universal description, discovery, and Integration), <http://www.uddi.org>, visitada el 14-08-2008.
- [09] W3C (World Wide Web Consortium), SOAP standard <http://www.w3.org/TR/SOAP/>, visitada el 14-08-2008
- [10] Sang Shin , JAX-WS Basics, 2008, <http://www.javapassion.com/webservices/jaxwsbasics.pdf>.
- [11] Jennifer Pérez Benedí, Web Services, Desarrollo de Web Services con Software Libre, Departamento de Sistemas Informáticos y Computación Universidad Politécnica de Valencia, Agosto 2007
- [12] Ricardo Seguel P, Seguridad en Web Services, Neosecure, Mayo 2005.

[13] Diego Martin L., Framework Web Iphone,
<http://www.minid.net/iphone/> visitada el 20-03-2009

[14] David Caneda, A jQuery plugin for mobile web development on the iPhone, iPod Touch, and other forward-thinking devices,
<http://www.jqtouch.com/>, visitada el 05-04-2009

[15] Groupaware, plugin para animaciones nativas sobre el iphone,
http://groupaware.mobi/iphone/#_Samples, visitada el 15-04-2009