



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

FACULTAD DE INGENIERÍA

**SISTEMA PARA LA AUTOMATIZACIÓN, CONTROL
Y MONITOREO RESIDENCIAL VÍA WEB; DISEÑO E
IMPLEMENTACIÓN**

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO EN COMPUTACIÓN

P R E S E N T A N:

**AÑORVE VIDAL LUIS EDUARDO
BAUTISTA GÓMEZ OSCAR ALAN**



**DIRECTOR DE TESIS:
M.I. RUBÉN ANAYA GARCÍA**

Ciudad Universitaria, 2014

A mi madre, por darme la fuerza necesaria para alcanzar mis metas y el amor necesario para disfrutarlas.

A mi padre, por trasmitirme su fascinación por el mundo del saber.

A mi hermana, por enseñarme a nunca dejar de ser único.

A mi abuela, por todo su cariño.

A Roxana, por compartir mis metas, mis logros y mis fracasos.

Y a Alan mi compañero de tesis, por soportar todos mis fallos durante la elaboración de este proyecto.

Luis.

A mis padres. Por todo su amor y apoyo a lo largo de mi vida. Porque este trabajo y todos mis logros son también fruto de ustedes. Por darme todo para ser quien soy.

A mi hermana. Por ser desde siempre mi gran ejemplo y motivación. Por todas sus enseñanzas y atenciones. Por todo su cariño.

Alan.

A la UNAM, nuestra universidad, que nos dio el conocimiento y las herramientas que nos forjaron como profesionales amantes del saber, con deseos de mejorar nuestro entorno y a nosotros mismos, siempre.

Ambos.

Índice de contenido.

Tabla de contenido.....	2
Capítulo 1: Introducción.....	8
1.1 Objetivo.....	9
1.2 Justificación.....	9
1.3 Marco teórico.....	11
1.3.1 Concepto de Domótica.....	11
1.3.2 Estructura de un sistema domótico.....	12
1.3.3 Elementos de un sistema domótico.....	14
1.4 Panorama general del sistema.....	17
Capítulo 2: Unidad Central.....	19
2.1 Objetivo y requerimientos.....	20
2.2 Cómputo embebido.....	21
2.3 Tarjeta Gizmo.....	23
2.3.1 Unidad de Procesamiento Acelerado.....	24
2.3.2 DDR3 SDRAM.....	25
2.3.3 Chipset AMD A55EFCH.....	25
2.3.4 Dispositivos adicionales.....	25
2.3.5 Interfaces de la tarjeta Gizmo.....	26
2.4 Sistema Operativo.....	28
Capítulo 3: Unidad de Control y Automatización de Iluminación.....	30
3.1 Objetivo.....	31
3.2 Estructura.....	31
3.3 Protocolo de comunicación X10.....	32
3.3.1 Definición del protocolo.....	33
3.3.2 Implementación de dispositivos X10.....	37

3.4	Software.....	38
Capítulo 4:	Unidad de Detección de Intrusión y Videovigilancia.....	41
4.1	Objetivo.....	42
4.2	Estructura.	42
4.3	Sensores.	43
4.4	Protocolo X10 RF.....	45
4.5	Cámaras IP	47
4.6	Software.....	49
Capítulo 5:	Unidad de gestión de llamadas a la puerta.	52
5.1	Objetivo y justificación.....	53
5.2	Estructura.	53
5.3	Botón de timbre y cerradura eléctrica.	54
5.3.1	Puertos de propósito general de entrada y salida	54
5.3.2	Circuitos de interfaz.....	56
5.4	Circuitos de Audio.	58
5.4.1	Amplificador para altavoz.....	58
5.4.2	Preamplificador de micrófono.	60
5.5	Software.....	61
5.6	Construcción e implementación.	62
Capítulo 6:	Aplicación Web.....	65
6.1	Objetivo y Justificación.....	66
6.2	Configuraciones previas.....	66
6.2.1	Servidor Web.	66
6.2.2	Manejador de base de datos.	67
6.2.3	Manejo dinámico de DNS.	67
6.2.4	Configuración del enrutador.	68
6.3	Tecnologías para la construcción de aplicaciones web.	70
6.3.1	HTML.	70
6.3.2	PHP.....	70

6.3.3	AJAX.....	71
6.3.4	JQuery.....	72
6.4	Interfaz de usuario.....	72
6.4.1	Control y automatización de iluminación.....	73
6.4.2	Videovigilancia.....	76
6.4.3	Detección de intrusión.....	77
6.4.4	Gestión de llamadas a la puerta.....	80
Capítulo 7:	Resultados, conclusiones y trabajo a futuro.....	81
7.1	Pruebas y resultados.....	82
7.2	Conclusiones.....	84
7.3	Trabajo a futuro.....	87
7.3.1	Mejoras.....	87
7.3.2	Cambios.....	87
7.3.3	Adiciones.....	88

Índice de ilustraciones.

Ilustración 1.1: Ejemplo de una residencia con un sistema domótico implementado.	12
Ilustración 1.2: Elementos de un sistema domótico y ejemplos.	14
Ilustración 1.3: Diagrama general del sistema.....	18
Ilustración 2.1: Tarjeta de desarrollo Gizmo	23
Ilustración 2.2: Diagrama de bloques de la tarjeta Gizmo.	24
Ilustración 2.3: Estructura de una distribución Linux.....	29
Ilustración 3.1 Estructura de la Unidad de Control y Automatización de Iluminación.	32
Ilustración 3.2 Onda de transmisión X10.....	33
Ilustración 3.3 Envío de datos binarios.	34
Ilustración 3.4 Estructura de las diferentes tramas X10.....	35
Ilustración 3.6. Proceso de control de iluminación en tiempo real.	38
Ilustración 3.7. Proceso de control de iluminación programado.	39
Ilustración 3.8. Proceso de recuperación posterior a un corte de energía.....	40
Ilustración 4.1 Estructura de la unidad de videovigilancia.....	42
Ilustración 4.2 Estructura de la Unidad de Detección de Intrusión.	43
Ilustración 4.3: Funcionamiento de un sensor de movimiento activo.	44
Ilustración 4.4: Funcionamiento de un sensor de movimiento pasivo.....	44
Ilustración 4.5: Modulación por posición de pulso.....	45
Ilustración 4.6: Ejemplo de una trama de datos X10 RF.....	46
Ilustración 4.7: Estructura de un sistema de vigilancia con cámaras IP.	48
Ilustración 4.8: Proceso de envío de mensajes de texto.	50
Ilustración 4.9: Proceso de activación y desactivación automática.	51
Ilustración 5.1: Estructura de la unidad de gestión de llamadas a la puerta.	54
Ilustración 5.2: Vista superior con algunos de los componentes de la tarjeta de exploración.	55

Ilustración 5.3: Circuito de interfaz del botón del timbre.	57
Ilustración 5.4: Circuito interfaz de la cerradura eléctrica.	58
Ilustración 5.5: Circuito amplificador de audio.....	59
Ilustración 5.6 Circuito preamplificador de micrófono.	61
Ilustración 5.7: Proceso de gestión de llamadas a la puerta.	62
Ilustración 5.8: Estructura de conexiones físicas de la instalación del videoportero.....	63
Ilustración 5.9: Tarjeta de interfaces del lado de la unidad central.....	64
Ilustración 5.10: Tarjeta de interfaces del lado del videoportero.	64
Ilustración 6.1: Servicios requeridos para la aplicación web.	69
Ilustración 6.2: Ejemplo de funcionamiento de PHP.	71
Ilustración 6.3: Arquitectura de comunicación por medio de AJAX.	71
Ilustración 6.4: Pantalla del módulo de control y automatización de iluminación de la interfaz.	73
Ilustración 6.5: Menú de acciones por lámpara.	74
Ilustración 6.6: Estructura del software de control de iluminación desde la interfaz de usuario.....	74
Ilustraciones 6.7 y 6.8: Pantallas de Programación y Edición de programaciones.	75
Ilustración 6.9: Estructura del software de programación de los dispositivos de iluminación desde la interfaz de usuario.....	75
Ilustración 6.10: Pantalla del Módulo de Videovigilancia.	76
Ilustraciones 6.11 y 6.12: Pantalla de visualización individual de cámara IP y Pantalla de visualización simultanea de cámara.	77
Ilustración 6.13: Pantalla del módulo de detección de intrusión.	77
Ilustración 6.14: Proceso de activación y desactivación de los desde interfaz de usuario. .	78
Ilustración 6.15: Proceso de programación de los sensores desde la interfaz de usuario. ..	79
Ilustración 6.16: Menú de edición de números para notificación por mensaje de texto.	79
Ilustración 6.17: Menú de edición de cuentas con acceso y notificación al videoportero. ..	80

Capítulo 1: Introducción.

1.1 Objetivo.

El presente trabajo tiene como objetivo el diseño y la implementación de un sistema que permita el control y la automatización de la iluminación y el monitoreo de sensores y cámaras de una residencia, utilizando la computadora *Gizmo* y el grupo de herramientas de código abierto *LAMP* (*Linux, Apache, MySQL, PHP*) en conjunto con la biblioteca *jQuery* para la implementación de una aplicación web como interfaz de usuario.

En términos específicos se buscará que el sistema sea capaz de llevar a cabo las siguientes tareas:

1. Automatización y control de los sistemas de iluminación, entendiéndose básicamente como el encendido y apagado de lámparas de manera remota y/o automática.
2. Monitoreo local y remoto con cámaras de vigilancia.
3. Detección de intrusión y notificación mediante mensajes de texto.
4. Gestión de llamadas a la puerta, con interacción audiovisual y capacidad de apertura, vía local y remota.

Adicional a esto, se pretende que posea las cualidades de ser multiplataforma, remoto, automático, adaptable y altamente escalable.

1.2 Justificación.

Al tratarse de un sistema del campo de la domótica abarcará sus principales áreas de aplicación: brindar comodidad, seguridad y ahorro energético, cuyo objetivo es finalmente el uso de la tecnología para incrementar la calidad de vida.

Con este proyecto se busca la innovación, al diseñar e implementar un sistema que cumpla con características y cualidades que no suelen encontrarse en conjunto en el mercado actual. A continuación se profundizará en cada una de ellas.

Multiplataforma. Un sistema se considera multiplataforma, si puede ser ejecutado en diferentes entornos (dispositivos o plataformas de software), sin que estos requieran una configuración previa o adicional a las características que por sí mismos poseen.

Se hace referencia a sistemas multiplataforma en distintos niveles de abstracción, partiendo del hardware hasta las aplicaciones con que interactúa el usuario final. Una aplicación web se encuentra idealmente en la **capa de abstracción** más alta, lo que la convierte en una de las mejores soluciones para la implementación de una interfaz gráfica de usuario. Así se garantiza tener el control y permitir la interacción con todo el sistema domótico desde prácticamente cualquier dispositivo, móvil o fijo, que posea un navegador web relativamente reciente, independientemente del sistema operativo que ejecute y sin tener que recurrir a tecnologías privativas y costosas.

Remoto. Adicional a las ventajas ya mencionadas, el uso de las tecnologías web nos ofrece la posibilidad de acceder a las características de configuración y control del sistema de manera local y remota, siempre de forma transparente para el usuario. Las implicaciones de esto se reflejan en usos como la simulación de presencia al salir de viaje y por supuesto el monitoreo de módulos de seguridad y comunicación desde prácticamente cualquier lugar.

Automático. Implica el control autónomo sobre los diferentes dispositivos o servicios involucrados, en todo momento. Al automatizar los sistemas de iluminación de una vivienda por ejemplo, se puede lograr la disminución del consumo de energía eléctrica, lo que resulta en reducción de costos del servicio, y por supuesto una gran ventaja en comodidad para el usuario.

Adaptable. Existen soluciones que demandan instalaciones invasivas o con un alto costo, en muchas ocasiones requiriendo residencias aún en construcción. Un sistema domótico adaptable se debe caracterizar por evitar al máximo estos inconvenientes, ofreciendo facilidad de instalación y adaptándose a diferentes escenarios de espacio y presupuesto.

Escalable. La escalabilidad es la cualidad que posee un sistema o servicio de manejar convenientemente su crecimiento a futuro, principalmente al añadirsele

mejoras o características adicionales a aquellas con las que contaba en un inicio. En domótica, es inevitable pensar en que las necesidades o requerimientos de los usuarios pueden cambiar a lo largo del tiempo, ya sea por el surgimiento de una necesidad no contemplada inicialmente, o por el simple deseo de nuevas características y funcionalidades. Por todo ello es deseable desarrollar un sistema altamente escalable.

Finalmente, se espera que este trabajo sirva como una base sólida para futuros desarrollos en el campo de la domótica, así como en el área de sistemas embebidos, mostrando sólo algunas de las posibilidades que nos ofrecen las nuevas tecnologías con un enfoque que puede sin duda brindar un gran apoyo a personas mayores o con alguna discapacidad.

1.3 Marco teórico.

1.3.1 Concepto de Domótica

El concepto de domótica (del latín *domus*, casa, e *informática*), se define como el conjunto de sistemas que automatizan las diferentes instalaciones de una vivienda. Se entiende como aquellos sistemas que en su conjunto le proporcionan a una vivienda los servicios de comodidad, seguridad y gestión energética, de forma automática mediante el uso de la tecnología. [1]

Considerando lo anterior, se puede establecer una diferencia entre aquellos sistemas de gestión de edificios industriales, cuyo enfoque principal ha sido la seguridad y el ahorro energético, despreciando el nivel de comodidad que pueda obtenerse con ellos. En contraste, los servicios enfocados a una residencia están motivados principalmente para incrementar la comodidad de quienes la habitan. Este trabajo se centrará en el desarrollo de un sistema de uso residencial, por lo que sólo se pretende abarcar esta área de aplicación, apegándose al concepto inicial de domótica.

El desarrollo de la domótica ha ido a la par de la evolución tecnológica agregando a ella dispositivos de mayor complejidad cada vez, prácticamente desde el inicio de la electricidad domestica a principios del siglo XX. Se pueden encontrar ejemplos de domótica desde antes de la expansión masiva del uso de la computadora como son el uso de interruptores de pared, el drenaje interno y el agua corriente, los calentadores automáticos de agua, los termostatos, los sistemas de bombeo automático de agua, etc. [2] [3]

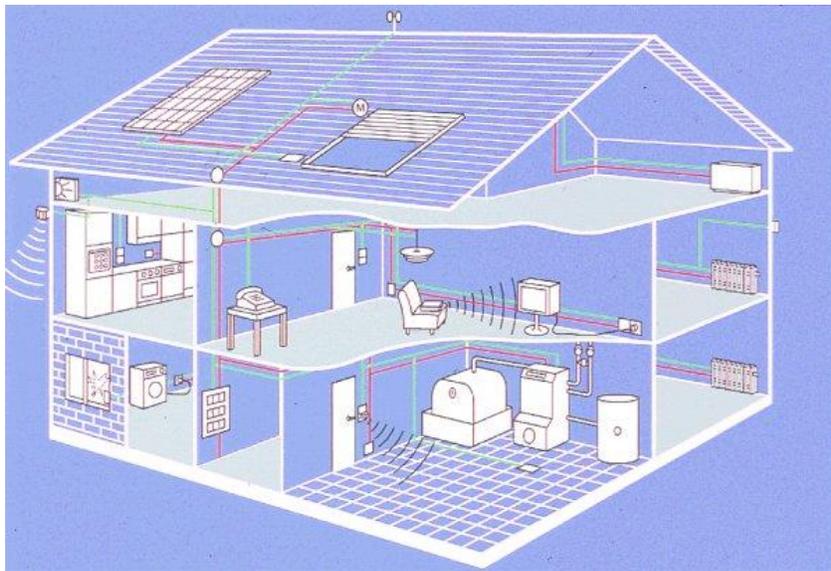


Ilustración 1.1: Ejemplo de una residencia con un sistema domótico implementado.¹

1.3.2 Estructura de un sistema domótico.

Un criterio para la clasificación de los sistemas domóticos es la manera en cómo se organizan e interactúan entre sí los diversos dispositivos que los componen. De acuerdo con este criterio existen dos arquitecturas principales, además de una tercera que combina características de las otras:

- **Arquitectura centralizada.** Aquella que concentra todas las tareas de control y coordinación de los demás componentes en un solo dispositivo

¹ Tomado del sitio "Web de Domótica". Alfonso Moratalla Moreno y Eduardo Ruiz Peña. <http://www.info-ab.uclm.es/labelec/Solar/domotica/index.htm>

principal. Entre sus fortalezas están el hecho de que en caso de ocurrir fallos en los dispositivos periféricos el resto del sistema no se vería afectado. El poseer un equipo complejo de cómputo le permite añadir funcionalidades que exijan mayor capacidad de procesamiento. Sus debilidades radican en la completa dependencia que existe hacia la unidad central, ya que si esta llegara a presentar alguna falla, se vería comprometido el funcionamiento de todo el sistema.

- **Arquitectura distribuida.** Es aquella en la que varios de los dispositivos cuentan con sistemas de cómputo independientes capaces de intercambiar información entre sí sin que alguno de ellos en particular se dedique al procesamiento de todos los datos, estos sistemas pueden incluir sensores y actuadores dentro del mismo dispositivo. Sus fortalezas contrastan con las de la arquitectura centralizada. Al tener los dispositivos que lo conforman un mayor grado de autonomía el sistema se vuelve más tolerante ante posibles fallas de estos; de igual forma, al distribuir la carga de procesamiento no es necesario contar con dispositivos demasiado complejos. La debilidad fundamental es que los equipos están limitados a realizar tareas más “simples” que las que puede realizar una arquitectura centralizada.
- **Arquitectura mixta.** Combina características de las dos arquitecturas anteriores buscando poseer sus fortalezas evitando sus debilidades. En términos prácticos sus debilidades y fuerzas dependen de la forma en cómo se combinen las arquitecturas y los requerimientos específicos de cada proyecto.

En lo relativo al costo se prefiere una u otra arquitectura dependiendo de la cantidad de dispositivos necesarios para el proyecto. Así pues, para pocos dispositivos resulta más costoso invertir en un sistema centralizado pues gran parte del gasto está innecesariamente destinado a la unidad central y en menor medida a los dispositivos periféricos.

Cabe señalar que en la realidad no existen prácticamente sistemas domóticos que se clasifiquen totalmente en alguna de las arquitecturas antes mencionadas. En la

mayoría de los casos suele ser cuestión de enfoque definir si un sistema se encuentra en una u otra clasificación.

1.3.3 Elementos de un sistema domótico.

Cualquier sistema domótico está conformado por tres partes esenciales, coincidentes con las de cualquier sistema computacional: las entradas de datos al sistema, que pueden provenir de sensores o controles (hardware) e interfaces de usuario (software); las salidas del sistema o actuadores, representados por dispositivos físicos, o bien en un sentido más laxo aplicaciones informáticas que de igual forma “actúan” realizando acciones concretas de software propias de sistemas domóticos modernos. Por último los controladores, encargados del procesamiento de los datos provenientes de las entradas al sistema, el control y coordinación de todas las salidas del mismo, así como el almacenamiento de la información requerida para la realización de estas tareas.



Ilustración 1.2: Elementos de un sistema domótico y ejemplos.

1.3.3.1 Dispositivos de entrada.

En el contexto de la domótica el diseño de los sensores y demás dispositivos de entrada ha estado orientado a las necesidades planteadas por los servicios principales que por lo general ofrece un sistema de este tipo (seguridad, comodidad y gestión energética). Así pues, una gran cantidad de los sensores actualmente existentes prometen brindar seguridad mediante detección de peligros potenciales tales como incendios, inundaciones, fugas de gas, intrusos, etc. Los restantes se pueden clasificar en los otros dos rubros: un sensor de temperatura puede utilizarse para mantener la temperatura de un espacio a niveles que el usuario

considere confortables y a la vez ahorrar la energía al hacer más eficiente este proceso.

Por otra parte en lo que refiere a los controles incorporados por el sistema (mandos de hardware e interfaces de software) su uso está enfocado mayoritariamente a la comodidad pues facilitan el control, aunque de forma indirecta, de todos los actuadores, siendo la tendencia actual la implementación de aplicaciones de esta índole en dispositivos móviles “inteligentes”.

1.3.3.2 Controladores.

Estos brindan la capacidad de procesamiento a los sistemas domóticos. Se trata de dispositivos que van desde sistemas analógicos de control hasta computadoras dedicadas a esta tarea, que de acuerdo a los elementos que conforman el sistema y su estructura pueden tener menor o mayor complejidad, capacidades de comunicación e interacción, dimensiones etc.

Como se mencionó anteriormente, la arquitectura del sistema define en gran medida las características de los controladores. Es decir, una arquitectura distribuida suele tener varios equipos simples a modo de controladores, mientras que las centralizadas cuentan con equipos con mayores capacidades y por lo tanto son de mayor tamaño, consumo energético y costo. Generalmente existe una mayor variedad de controladores en este tipo de arquitectura, que cumplen con el papel de la unidad central tales como PC's, sistemas embebidos y **microcontroladores**.

1.3.3.3 Dispositivos de salida.

Estos conforman el eslabón final de la cadena del sistema, ejecutando de forma directa las acciones deseadas por el usuario. En la actualidad los sistemas domóticos son capaces de controlar un sinfín de servicios y dispositivos. Ejemplos de estos son los interruptores electromecánicos o electrónicos que se encarguen de apagar o encender un electrodoméstico, motores para la apertura y cierre de persianas, electroválvulas o inclusive procesos de software que automaticen el envío de alertas mediante correos electrónicos.

1.3.3.4 Medios de transmisión.

Los desarrollos en domótica han utilizado a lo largo del tiempo diferentes medios físicos para transmitir información entre dispositivos. La tendencia actual ha sido y es, buscar métodos que alteren en la menor medida posible las viviendas, por lo que la mayoría de los sistemas domóticos prefieren tecnologías que ocupen la instalación eléctrica existente o bien sean inalámbricas.

El primer intento formal por crear una tecnología de red de propósito general para la automatización de viviendas ocurrió a finales de la década de los 70's con el desarrollo del protocolo X10 por la compañía escocesa *Pico Electronics of Glenrothes*. X10 utiliza el cableado de la línea eléctrica para transmitir datos mediante la técnica conocida como corrientes portadoras, lo que supuso el primer gran avance en las tecnologías domóticas, convirtiéndose en un estándar *de facto*. Hasta el día de hoy este protocolo sigue utilizándose ampliamente, principalmente en Norteamérica y Europa.

Posteriormente han surgido una serie de estándares para comunicaciones a través de la línea eléctrica, orientados principalmente a la implementación de redes de datos como el IEEE-P1901, IEC-61334, sin embargo y a pesar de las notables mejoras que presentan frente a X10 su difusión en el campo de la domótica no ha sido tan amplia.

Por otro lado, las comunicaciones inalámbricas han sido una buena alternativa al ofrecer facilidad en la instalación de los sistemas domóticos. Sus características coinciden con las de las redes inalámbricas de área personal (WPAN) debiendo además ser capaces de ofrecer un bajo costo y consumo mínimo de energía.

Como respuesta a esto se creó la especificación IEEE 802.15.4, la cual define la capa física y de acceso al medio para las WPAN con bajas tasas de transmisión de datos con un máximo de 250 kbps. Para las capas superiores se han desarrollado diversas soluciones, libres y propietarias como **ZigBee**, **Z-Wave**, **ONE-NET**, entre otras.

Actualmente, los desarrollos relacionados con domótica, suelen hacer uso de múltiples tecnologías y protocolos, esto de acuerdo a las necesidades específicas de cada caso. [4]

1.4 Panorama general del sistema.

En general el sistema está formado por cinco unidades de acuerdo a sus objetivos específicos.

1) Unidad central: Es la encargada de controlar, coordinar, y comunicar a todas la demás unidades; a la vez de permitir la interacción con el usuario a través de la aplicación web. Para su implementación se utilizará la computadora *Gizmo* que cuenta con interfaces USB, **Gigabit Ethernet**, entrada y salida de audio, puertos de propósito general de entrada y salida (GPIO), además de las características de cómputo necesarias para ejecutar todo el software que comprenderá al sistema.

2) Unidad de control de iluminación: Permite que la unidad central controle los dispositivos de iluminación; utilizando el protocolo de comunicación X10. Hace uso de un **transceptor** que se conecta a la unidad central mediante el puerto USB y enlaza con los interruptores X10 los cuales de acuerdo con las señales de control realizan el apagado o encendido de las lámparas, en tiempo real o actuando de acuerdo a la programación introducida por el usuario.

3) Unidad de detección de intrusión y videovigilancia: Alerta a la unidad central sobre la detección de movimiento enviando su estado a través de una señal X10 inalámbrica al transceptor para su procesamiento en la unidad central misma que enviará un mensaje de texto corto al usuario con la alerta de intrusión . Proporciona además la capacidad de supervisar ciertas áreas de la residencia utilizando cámaras IP.

4) Unidad de gestión de llamadas a la puerta: Gestiona el acceso a la residencia utilizando un servicio de videollamadas, que se desencadenará presionando el botón de timbre, comunicando al visitante frente a la puerta con el residente y brindado además la opción de abrir la puerta a distancia usando una cerradura eléctrica controlada a través de la unidad central. Todo esto sin importar la ubicación del residente pudiendo este abrir desde cualquier lugar, siempre que cuente con una conexión a internet

La arquitectura del sistema en su totalidad se puede apreciar en la ilustración 1.3.

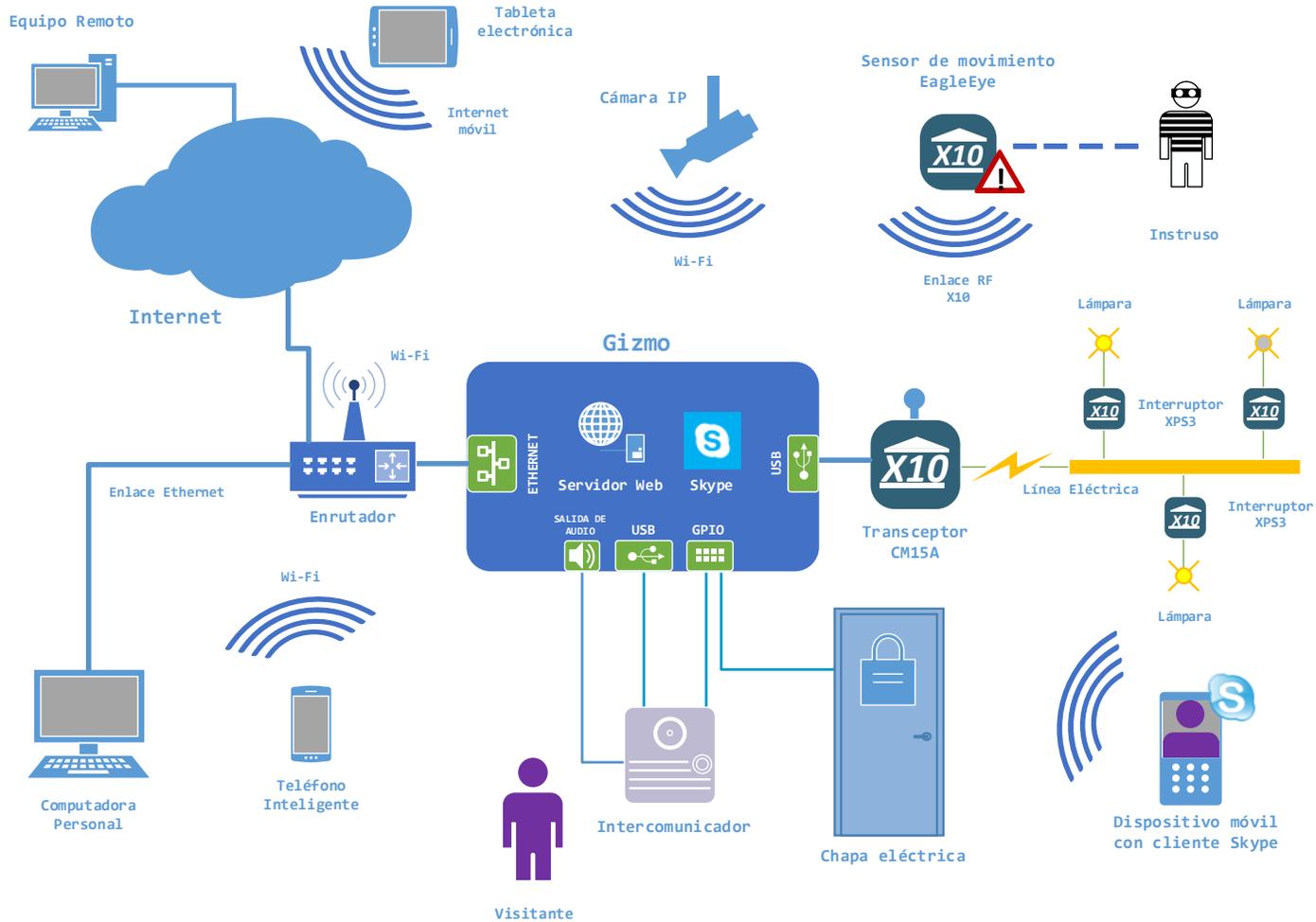


Ilustración 1.3: Diagrama general del sistema

Capítulo 2: Unidad Central

2.1 Objetivo y requerimientos

La unidad central es el núcleo del sistema, encargada de controlar todos los subsistemas, procesar las señales provenientes entre ellos y comunicar al usuario con todo el sistema en general a través de la aplicación web o desde alguno de sus puertos de entrada o salida de propósito general. Por esto es que resulta obvia su importancia, así como la necesidad de que cuente con ciertas características específicas de acuerdo a los requerimientos de este proyecto.

- Espacio reducido. Como parte de un sistema de uso residencial, es deseable que su presencia sea la más discreta posible, por tanto, el espacio que ocupe debe ser mínimo.
- Comunicación por red de datos. Una de las características principales de este sistema debe ser su capacidad de comunicación a través de una red de datos, accesible para cualquier dispositivo capaz de conectarse a ella. La unidad central debe por tanto poseer conectividad con este tipo de redes.
- Bajo consumo de energía eléctrica. Debiendo permanecer en todo momento encendido y en constante funcionamiento, debe tratarse de un dispositivo con un consumo reducido de energía eléctrica.
- Conjunto de instrucciones x86. Algunas de las características que el sistema requiere para su pleno funcionamiento son la capacidad de realizar videollamadas y el envío de mensajes de texto de forma automática. La aplicación que es capaz de llevar a cabo todas estas tareas sólo puede ejecutarse en computadoras que posean un **conjunto de instrucciones x86**, por lo que se vuelve un requerimiento indispensable.
- Capacidad de cómputo. Llevar a cabo las tareas propias de la unidad central exige capacidad de cómputo suficiente para tener montado un servidor web, procesar audio y video, y mantener activo el control sobre cada periférico.
- Interfaces de comunicación específicas. Cada uno de los dispositivos que integran el sistema demanda un tipo de interfaz específica, por lo que la unidad central debe estar provista de las interfaces necesarias para comunicarse con todos ellos: interfaces USB Host, entrada y salida de audio, así como puertos de entrada y salida de propósito general.

- Sistema operativo. Para facilitar el desarrollo del sistema es deseable delegar las tareas de gestión de hardware a bajo nivel a un sistema operativo, permitiendo así enfocar el proyecto mayormente en el desarrollo de software.

Por último, resulta obvio que al realizar un proyecto de ingeniería se busca minimizar costos en la medida de lo posible, por lo que el costo es también un factor importante a considerar.

2.2 Cómputo embebido

En congruencia con los requerimientos antes planteados para la implementación de la unidad central, la solución más adecuada resulta ser el cómputo embebido, descartando el cómputo personal debido a sus dimensiones, su alto consumo de energía y su costo excesivo; en contraste, el cómputo embebido es capaz de ajustarse a las necesidades de gran variedad de proyectos, particularmente en este.

Un sistema de cómputo embebido es un dispositivo con un procesador diseñado para desempeñar una función o un rango de funciones específicas, y a diferencia de los sistemas de propósito general, el usuario final no puede cambiar las características de software o hardware con las que ya cuenta. [5] [6]

Típicamente un sistema embebido se diseña con ciertas restricciones. Al ser parte de un sistema mayor se busca que tenga un tamaño pequeño y poco peso; usualmente operan utilizando baterías por largo tiempo lo que se traduce en bajo consumo de energía. Finalmente al ser usados en ocasiones para llevar a cabo aplicaciones críticas estos deben ofrecer altos niveles de confiabilidad.

De acuerdo con la complejidad de un sistema embebido este puede contar o no con un sistema operativo que provea una capa de abstracción de hardware para simplificar la interacción de las aplicaciones con el hardware y en consecuencia permitir la integración de software de múltiples fabricantes.

En el mercado existen plataformas de hardware tanto para el desarrollo como la implementación de sistemas embebidos. Cada una cuenta con características de

hardware y software particulares, y en su conjunto ofrecen una amplia gama de posibilidades para dar solución a distintos problemas o necesidades.

La tabla 2.1 muestra una comparación de algunas plataformas de desarrollo que a primera vista cumplen con los requerimientos anteriormente planteados, de acuerdo a sus características técnicas y de costo. [7] [8] [9]

Nombre	Gizmo	MinnowBoard	Norco BIS-6630
Procesador	AMD G-T40E APU (Doble Núcleo) @ 1 [GHz]	Intel Atom E640 @1[GHz]	Intel Atom D2550 @1.7[GHz]
RAM	1 [GB] DDR3	1 [GB] DDR2	2 [GB] Incluidos y Hasta 4[GB] DDR3
Almacenamiento	Puerto SATA 3	Ranura μ SD y Puerto SATA 2	2 puertos SATA 3
USB Host	2	2	6
Ethernet	1	1	2
GPIO	8	8	0
Video	VGA	DVI vía HDMI	VGA, DVI
Audio	Entrada y Salida	Entrada y Salida	Entrada y Salida
S.O.	Windows y Linux	Ångström Linux*	Windows y Linux
Dimensiones [cm]	15 X 15	10 x 10	19 x 19
Costo (USD)	199	199	279

Tabla 2.1: Comparación de plataformas de desarrollo embebidas.

La tarjeta de desarrollo *Gizmo* tiene una mejor proporción entre sus características de cómputo y costo, adicionalmente ofrece una mayor flexibilidad en cuanto a elección del sistema operativo, en contraste con la *MinnowBoard* que hasta la fecha (Julio de 2013) no ha sido probada en sistemas adicionales a la distribución *Ångström Linux*, lo cual será una característica relevante a la hora de implementar el sistema de software en su totalidad.

2.3 Tarjeta Gizmo

La Tarjeta *Gizmo* es una plataforma de desarrollo embebido de dimensiones reducidas y bajo costo basada en el conjunto de instrucciones x86. Utiliza la unidad de procesamiento acelerado AMD Embedded G-Series G-T40E, es manufacturada por la organización *GizmoSphere* y fue lanzada al mercado en el 2013. A continuación se describen sus características técnicas.

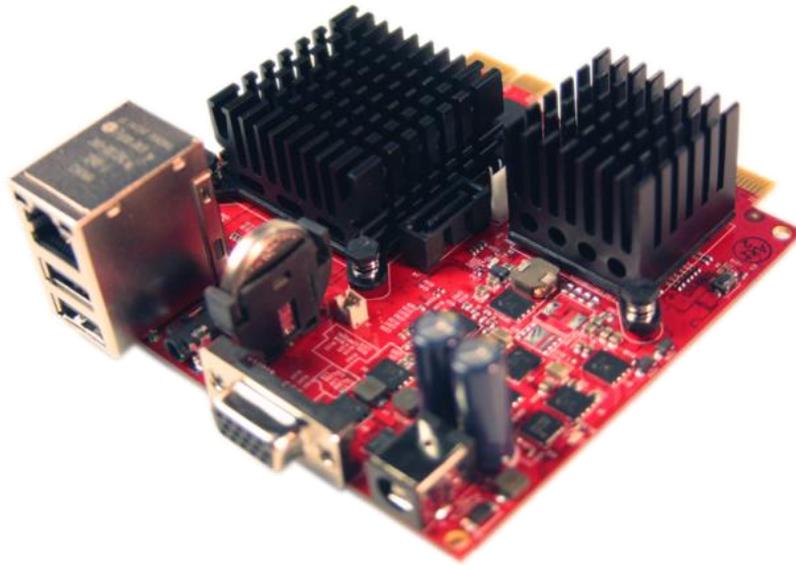


Ilustración 2.1: Tarjeta de desarrollo Gizmo

La figura 2.2 muestra un diagrama básico con la estructura general de la tarjeta Gizmo, ilustrando los circuitos integrados e interfaces que posee.

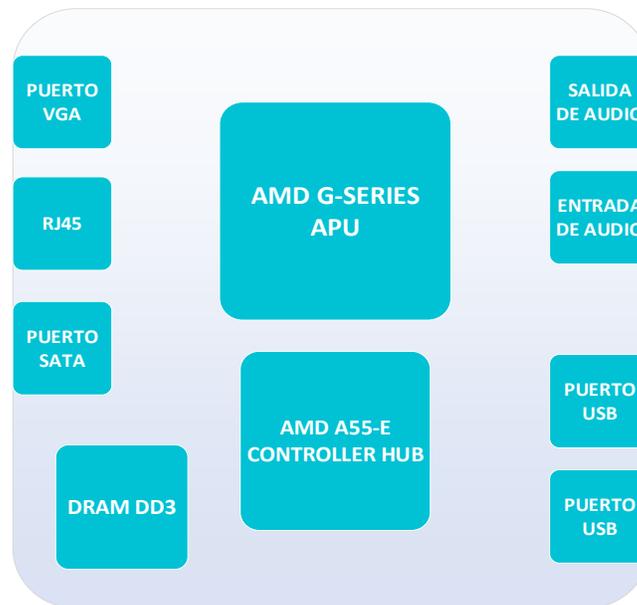


Ilustración 2.2: Diagrama de bloques de la tarjeta Gizmo.

2.3.1 Unidad de Procesamiento Acelerado.

Se conoce como APU (*Accelerated Processing Unit*) a la tecnología desarrollada por la compañía AMD que combina una **CPU** con unidades adicionales de procesamiento en un único circuito integrado. En el caso particular la serie G es la combinación de una CPU de bajo consumo de energía con una **GPU**.

La AMD Embedded G-Series G-T40E es una APU diseñada para usarse principalmente en aplicaciones embebidas como quioscos de información, puntos de venta, máquinas de juegos, etc. Constituye el núcleo de la tarjeta Gizmo encargándose de todas las tareas de cómputo de la tarjeta. Sus características técnicas son las siguientes:

- CPU de doble núcleo.
- 512 [KB] de **caché** de nivel 2 por núcleo.
- Interfaz de memoria DDR3-1066.
- Frecuencia de reloj 1.0 [GHz].
- GPU AMD Radeon HD 6250.
- Frecuencia de reloj de la GPU 280 [MHz]
- Aceleración de video por hardware UVD 3 para H.264, UVC-1 y MPEG2.

- La GPU soporta: OpenGL 4.0, DirectX 11, Shader Model 5 y OpenCL 1.1.
- Salidas de video soportadas (2 salidas independientes): VGA, DVI, LVDS, Display Port 1.1a, HDMI, DVO.

2.3.2 DDR3 SDRAM.

Double Data Rate Synchronous Dynamic Random Access Memory. La tarjeta Gizmo cuenta con 1[GB] de este tipo de memoria principal. Se trata de una memoria volátil de acceso aleatorio dinámica, síncrona (comparte y se sincroniza con el reloj del sistema) y que utiliza la tecnología DDR3. Tiene una frecuencia de oscilación de reloj interno de 120 [MHz], una de reloj de entrada/salida de 480 [MHz] y una velocidad de transferencia de 960 [MT/s] (Megatransferencias por segundo), lo que significa que es capaz de realizar, desde o hacia su bus, un total de 960 millones de transferencias por segundo. Como se observa, la característica a destacar de la tecnología DDR3 es su capacidad de realizar dos transferencias de datos por cada ciclo de reloj de entrada/salida; así mismo el reloj de entrada/salida cuadruplica la frecuencia de reloj interno.

2.3.3 Chipset AMD A55EFCH

Es un circuito integrado auxiliar que permite la gestión de los periféricos y las interfaces. Los componentes más notables del AMD A55EFCH son:

- Controlador SATA 3 6Gb/s.
- Controlador PCI express 2.0.
- Controlador USB Host 2.0.
- Controlador de Audio de Alta definición con soporte para hasta cuatro canales.
- Reloj en tiempo real.
- Soporte de hasta 102 GPIO's.

2.3.4 Dispositivos adicionales

ROM SPI. Se trata de una memoria tipo **Flash** con capacidad de 64 [MB] que almacena el BIOS (*Basic Input Output System*). Particularmente *Sage BIOS* que es una distribución del BIOS libre *Coreboot* en su versión 5.1. El BIOS es el

programa que se encarga de inicializar los dispositivos indispensables (teclado, salida de video y dispositivos de almacenamiento secundario) para enseguida cargar y arrancar un sistema operativo.

Transceptor físico Ethernet. Conocido también como *Ethernet PHY* es el circuito integrado que implementa la porción **Ethernet** de la capa física de los estándares 10/100/1000 Base-T que corresponden a velocidades de 10, 100 y 1000 [Mb/s] respectivamente sobre un medio físico de cable par trenzado. Implementa las funciones de recepción y transmisión a nivel hardware de las **tramas Ethernet**.

Unidad de manejo de poder. *Power Management Unit*. Es un microcontrolador encargado de controlar las funciones de potencia en dispositivos digitales. Es responsable del monitoreo de las conexiones de poder y de la carga de la batería (de existir esta). Realiza la carga de la batería cuando es necesario, controla el consumo de potencia de los otros circuitos integrados, apaga los componentes innecesarios del sistema cuando se encuentran inactivos, y controla las opciones de suspensión y encendido.

Códec de Audio. Es una combinación de hardware y software que permite la codificación y decodificación de datos de audio. En el caso particular, la tarjeta *Gizmo* cuenta con un códec de audio de alta definición. Este posee 4 canales de audio mediante dos convertidores analógico/digital estéreo y dos convertidores digital/analógico estéreo. Adicionalmente cuenta con un convertidor de micrófono estéreo digital, además de dos convertidores independientes de salida S/PDIF (*Sony/Philips Digital Interface Format*).

2.3.5 Interfaces de la tarjeta Gizmo

SATA. *Serial Advanced Technology Attachment*. Es una interfaz entre la tarjeta y dispositivos almacenamiento como discos duros, lectores y grabadores CD/DVD/BR, unidades de estado sólido. Puede operar a velocidades de hasta 6 [Gb/s].

USB Host. *Universal Serial Bus Host*. Anfitrión de bus serie universal. La tarjeta cuenta con dos conectores USB Host 2.0. Se trata de una interfaz estandarizada para la conexión de todo tipo de periféricos, tales como teclado y mouse,

dispositivos de almacenamiento, adaptadores de red inalámbrico, etcétera. Posee tres modos de velocidad de comunicación, de acuerdo a la aplicación: baja velocidad, para dispositivos interactivos como teclados y mouse (de 10 a 100 [Kb/s]), velocidad completa, para audio y video comprimido, telefonía, entre otros (de 500 [Kb/s] a 10 [Mb/s]), y alta velocidad para video sin compresión y dispositivos de almacenamiento (de 25 [Mb/s] a 400 [Mb/s]). Permite la conexión de idealmente hasta 127 dispositivos haciendo uso de concentradores adicionales.

Conector RJ45. *Registered Jack 45.* Es una interfaz de ocho terminales para la conexión de redes de datos. Se encuentra regulada por el estándar TIA/EIA 568-B y permite la conexión de cables par trenzado al transceptor físico Ethernet.

Entrada y salida de audio. La tarjeta cuenta con interfaces de entrada y salida de audio mediante un jack de cuatro canales de 3.5 [mm].

VGA. *Video Graphics Array.* Es un estándar de video analógico de alta resolución usado en la mayoría de los monitores de computadora. Utiliza cables separados para la transmisión de las tres señales de los componentes de color (rojo, verde, azul) y las señales de sincronización vertical y horizontal. La resolución máxima soportada es 1920 X 1200 con una profundidad de color de 30 bpp (bits por pixel).

JTAG. *Join Test Acton Group.* Es una interfaz estándar basada en la norma IEEE1149.1 (*Standard Test Access Port and Boundary-Scan Architecture*), que sirve como mecanismo para la depuración y detección de errores en aplicaciones embebidas.

Puertos de expansión.

La tarjeta Gizmo cuenta con dos interfaces especiales que funcionan como puertos de expansión permitiendo la comunicación con dispositivos adicionales de alta velocidad y dotándolo de características comúnmente encontradas en microcontroladores.

- Puerto de baja velocidad. Provee de la conexión con: El convertidor analógico/digital, el digital/analógico, **PWM**, puerto SPI, un contador, ocho puertos de entrada/salida de propósito general y el reinicio maestro.

- Puerto de alta velocidad. Se compone de las interfaces: PCIe, SATA, USB y Display Port.

2.4 Sistema Operativo

El sistema operativo es una parte fundamental de la unidad central ya que se encarga de controlar todo el hardware que conforma al sistema y ejecutar cada una de las implementaciones en software de todas las otras unidades. [10]

Para elegir el sistema operativo más adecuado para este desarrollo se busca aquel que cumpla en su totalidad los siguientes requisitos:

- Soporte total del hardware de la tarjeta *Gizmo*.
- Soporte de los dispositivos del sistema completo (Transceptor X10 y cámara web).
- Control de los puertos de propósito general.
- Capacidad de ejecutar el cliente Skype™.
- Capacidad de ejecutar el grupo de aplicaciones LAMP.

Actualmente existen una gran variedad de sistemas operativos tanto propietarios como libres. Si se les compara, para este desarrollo, los primeros no ofrecen ninguna ventaja significativa y en cambio suponen un costo adicional. Por esto, la mejor opción para esta aplicación se encuentra en los sistemas operativos libres. Dentro de estos sistemas aquellos basados en un núcleo Linux han tenido mayor desarrollo, soporte y distribución.

Linux está basado en el sistema UNIX y en el año de 1991 surge como una alternativa libre. Una de sus características más notables es su objetivo: que el usuario tenga el control total del hardware, rasgo ideal en entornos de desarrollo embebidos. A su vez cuenta con una comunidad de desarrolladores y usuarios que le proporcionan soporte. Cada vez es más frecuente encontrar sistemas Linux en un gran número de dispositivos.

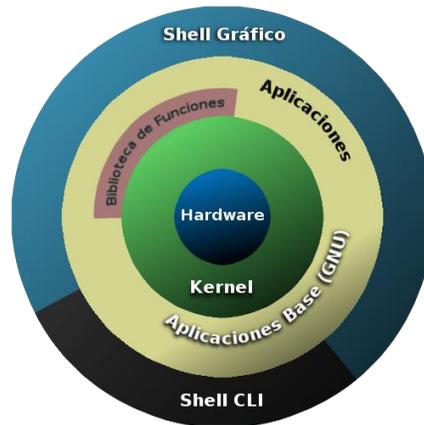


Ilustración 2.3: Estructura de una distribución Linux.²

Los sistemas basados en un núcleo de Linux se distribuyen en un grupo de paquetes de software, orientados a satisfacer las necesidades de un grupo específico de usuarios, conocidos como distribución Linux. En la ilustración 2.3 se observa que la diferencia entre una distribución y otra radica únicamente en las capas externas al **kernel** (bibliotecas, aplicaciones e interfaces de usuario).

De acuerdo con los requerimientos oficiales del software Skype™, las distribuciones de Linux que lo soportan al día de hoy son: Debian 7.0 (32 o 64 bits), Ubuntu 10.04 (32 bits), Ubuntu 12.04 (32 o 64 bits), Fedora 16 (32 bits) y OpenSUSE 12.1 (32 bits). En consecuencia, las distribuciones anteriores se convierten en las opciones más viables, por no decir las únicas, para la elección del sistema operativo. De todas ellas se elige Debian al contar una de las comunidades de desarrollo más grandes y el mayor repositorio de paquetes de software. Las demás distribuciones están orientadas a ser usadas en cómputo personal y contienen paquetes innecesarios que en lugar de ofrecer características útiles pueden afectar el rendimiento del sistema.

² Tomado del sitio "Wikimedia Commons". http://commons.wikimedia.org/wiki/File:Arquitectura_linux.png

Capítulo 3: Unidad de Control y Automatización de Iluminación.

3.1 Objetivo.

El principal objetivo de la Unidad de Control y Automatización de Iluminación es que a través de un proceso computacional se logre el encendido y apagado de una o varias de las lámparas que conforman el sistema de iluminación de una vivienda. La unidad debe ser capaz de realizar estas acciones de acuerdo a un horario establecido previamente por el usuario (temporización) o bien directamente a través de órdenes inmediatas. Este proceso debe además tener la capacidad de comunicarse con la interfaz de usuario (aplicación web), recibiendo de ella las órdenes de control o los parámetros de configuración necesarios para la automatización.

3.2 Estructura.

La Unidad se compone concretamente de dos tipos de dispositivos. El primero de ellos, el transceptor X10 CM15A, es el encargado de procesar los comandos provenientes de la unidad central (a través de una interfaz USB) convirtiéndolos en señales X10, las cuales que se propagan a través de la instalación eléctrica de la vivienda. Para el control del transceptor desde la unidad central se utiliza el **demonio** de Linux *MOCHAD (Multiple Online Controllers for Home Automation Daemon)* software de **código abierto** y compatible con los modelos de transceptor CM15A y CM19A.

MOCHAD es capaz de enviar todos los comandos definidos por el protocolo X10 además de capturar los mensajes de eventos que pudieran ser generados por dispositivos X10. Para realizar esto *MOCHAD* hace uso de un **socket TCP** que le permite comunicar el transceptor con cualquier cliente, local o remoto, facilitando el control y el monitoreo de los dispositivos X10 instalados en la residencia.

Por otro lado, los interruptores X10 XPS3 traducen la señal procedente del transceptor y ejecutan la instrucción indicada, encendiendo o apagando la lámpara.

Idealmente se buscan dispositivos que permitan controlar las lámparas utilizando el interruptor de pared convencional, proporcionando así dos mecanismos para el control de la iluminación; de otra forma, aquellas lámparas controladas por comandos X10 quedarían inhabilitadas en el caso en que el interruptor de pared se encuentre abierto. El interruptor XPS3 posee esta capacidad por lo que se prefiere ante los zócalos existentes que también funcionan con X10.

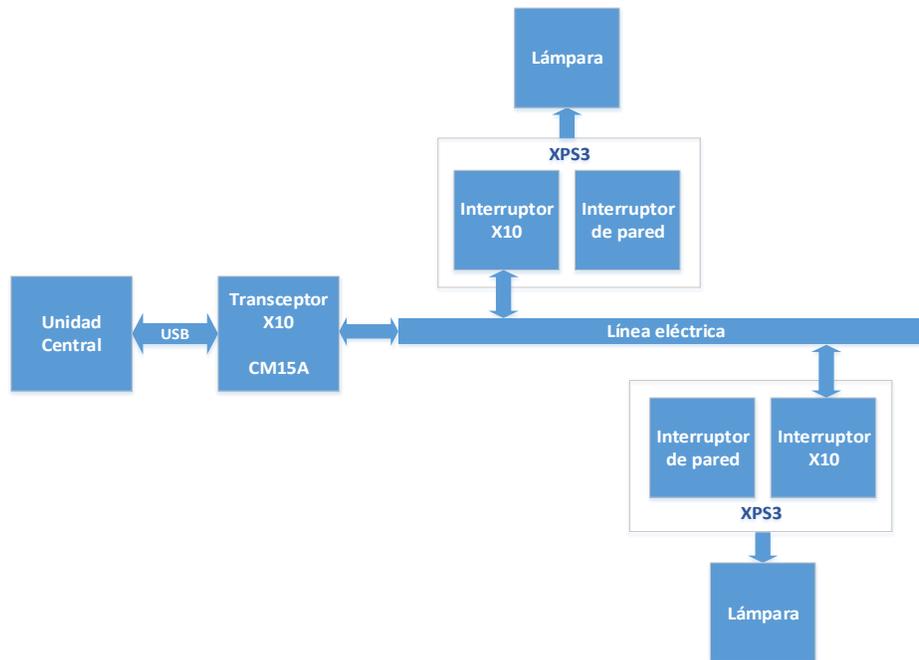


Ilustración 3.1 Estructura de la Unidad de Control y Automatización de Iluminación.

3.3 Protocolo de comunicación X10.

Como resultado de la investigación previa, el protocolo X10 se elige para este proyecto debido a su amplia adopción en el mercado y en consecuencia la gran variedad de productos que lo incorporan. En México, es la tecnología domótica más difundida, contando con compañías especializadas únicamente en la venta de éstos productos.

3.3.1 Definición del protocolo.

La comunicación entre dispositivos X10 se realiza a través del cableado de la red eléctrica de la residencia, montando un tren de pulsos de 120[kHz] de 1[ms] de duración en ella. Esta serie de pulsos se encuentran sincronizados con el cruce por cero de la señal de la línea eléctrica, con una tolerancia de 200 [μs], lo que permite a los dispositivos X10 saber en qué momento recibir o transmitir información. [11]

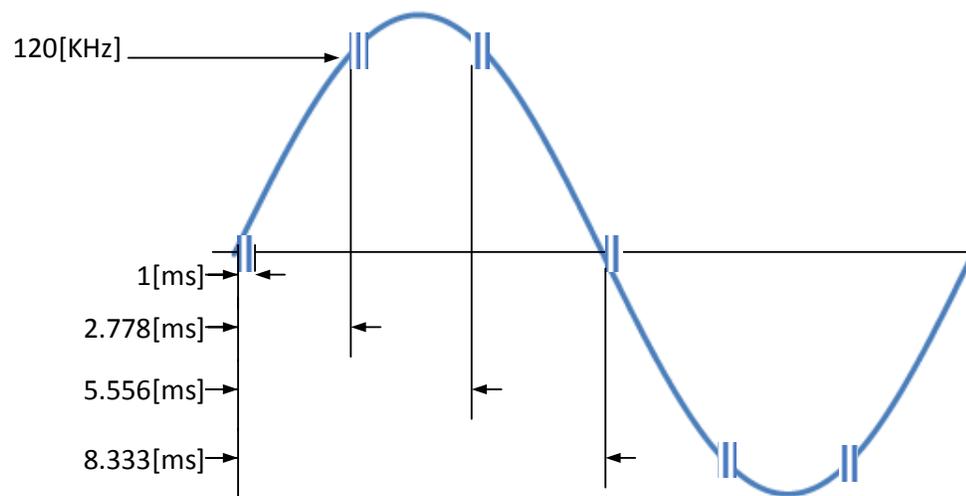


Ilustración 3.2 Onda de transmisión X10.

Para permitir el uso en redes trifásicas los trenes de pulsos de 120[kHz] se transmiten tres veces para coincidir con el cruce por cero de cada una de las fases.

En la ilustración 3.2 se muestra una onda típica de transmisión X10 donde se pueden apreciar los tres trenes de pulsos correspondientes a los tres cruces por cero de un sistema trifásico de 60 [Hz]. En caso de un sistema monofásico las dos últimas señales son ignoradas por el dispositivo X10.

Uno de los principales inconvenientes para la transmisión de señales a través de la línea eléctrica es la presencia de señales de ruido. Para dotar de cierta inmunidad al ruido, el protocolo X10 realiza la codificación de los datos binarios usando su complemento. Para enviar un bit se ocupan los dos cruces por cero que tiene un periodo; en el primero se envía el valor deseado y en el siguiente su complemento,

representado un “1” binario con un tren de impulsos y un “0” con la ausencia de este.

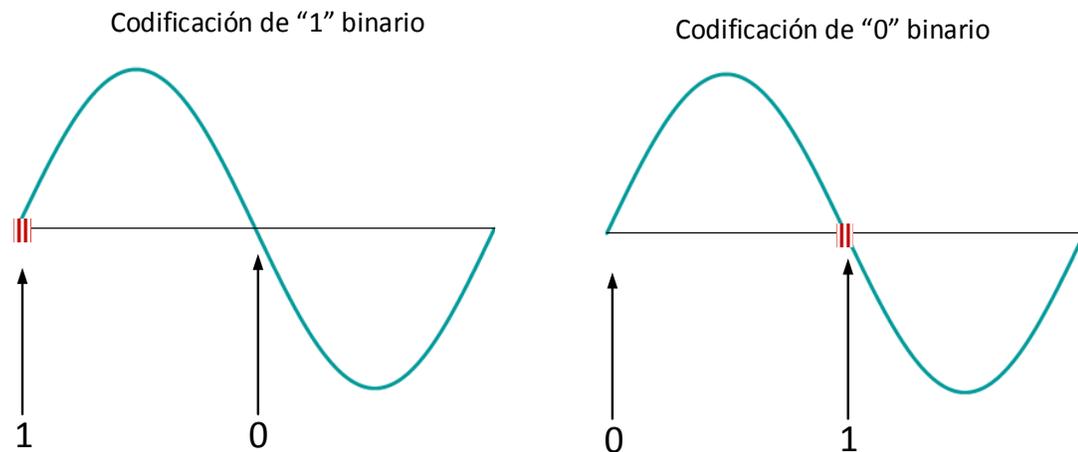


Ilustración 3.3 Envío de datos binarios.

X10 es un protocolo que funciona enviando tramas de información que contienen códigos de dispositivo y comandos específicos.

Una trama X10 está compuesta de cuatro partes básicas: un código de inicio, un código de casa, un código dispositivo o función y un código de fin. El código de inicio se representa con la secuencia “1110” transmitida solamente en dos ciclos, siendo la única excepción en la codificación complementaria de bits. En los siguientes cuatro ciclos se representan los cuatro bits del código de casa. Los siguientes cuatro están destinados a representar el código de dispositivo o de función. Por último se utiliza un ciclo para identificar si la trama corresponde a una función o una dirección de dispositivo.

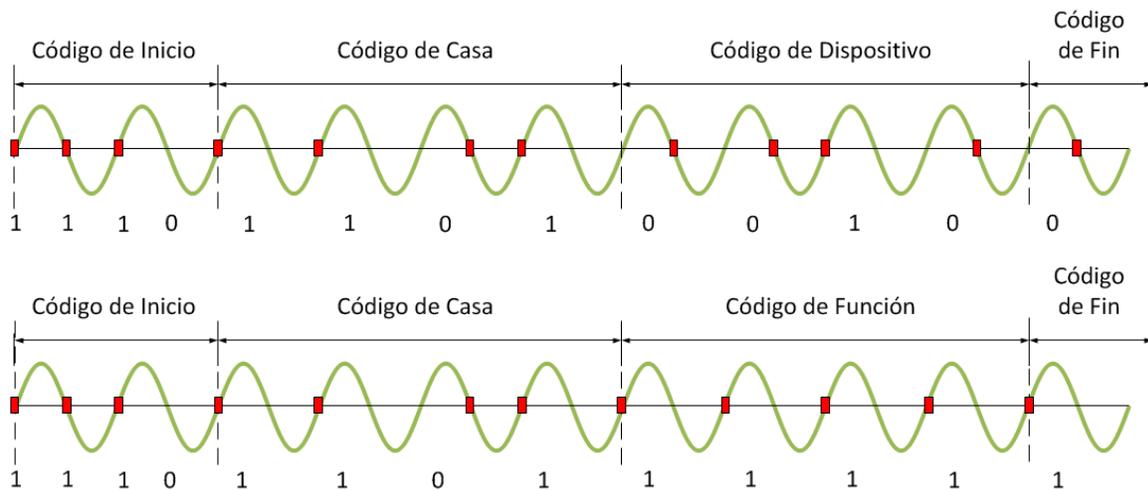


Ilustración 3.4 Estructura de las diferentes tramas X10.

Para enviar una orden X10 completa es necesario enviar una trama de dirección dos veces, lo cual activa el dispositivo deseado; esperar seis cruces por cero (exceptuando las funciones de incremento o disminución de intensidad de luz) y posteriormente enviar por duplicado una trama de función. Antes de enviar una nueva orden se debe esperar tres ciclos.

En la tabla 3.1 se muestran los comandos X10 disponibles así como su correspondiente código y su descripción.

Código	Función	Descripción
0 0 0 0	<i>All Units Off</i>	Apaga todos los dispositivos de iluminación con el código de la casa indicado.
0 0 0 1	<i>All Lights On</i>	Enciende todos los dispositivos de iluminación con el código de la casa indicado.
0 0 1 0	<i>On</i>	Enciende el dispositivo indicado.
0 0 1 1	<i>Off</i>	Apaga el dispositivo indicado.
0 1 0 0	<i>Dim</i>	Reduce la intensidad de la luz.
0 1 0 1	<i>Bright</i>	Aumenta la intensidad de la luz.
0 1 1 1	<i>Extended Code</i>	Código de Extensión.
1 0 0 0	<i>Hail Request</i>	Pide respuesta a los dispositivos con el código de la casa indicado en el mensaje.
1 0 0 1	<i>Hail Acknowledge</i>	Respuesta a la orden anterior <i>Hail Request</i> .
1 0 1 *	<i>Pre-Set Dim</i>	Permite la selección de dos niveles predefinidos de intensidad de la luz.
1 1 0 0	<i>Extended Data</i>	Datos adicionales (seguido por 8 bytes).
1 1 0 1	<i>Status is On</i>	Respuesta a la solicitud <i>Status Request</i> , indicando que el dispositivo está encendido
1 1 1 0	<i>Status is Off</i>	Respuesta a la solicitud <i>Status Request</i> que indica que el dispositivo está apagado
1 1 1 1	<i>Status Request</i>	Solicitud sobre el estado del dispositivo

Tabla 3.1: Lista de Comandos X10

El protocolo X10 es una buena opción a pesar de contar con inconvenientes tales como el hecho de no ser totalmente inmune al ruido en la línea eléctrica; sin embargo, existen en el mercado filtros dedicados a la eliminación del ruido producido por aparatos domésticos.

Al estar orientado específicamente a la comunicación entre dispositivos que forman un sistema domótico y al hecho de que su velocidad de transferencia es extremadamente baja (60 [b/s]), no es utilizable para el envío de otro tipo de información que no sea el control de dispositivos X10 compatibles.

Otro punto débil del protocolo X10 es la falta de algún sistema de encriptación, por lo que no se le debería confiar ninguna tarea que suponga un riesgo de seguridad, sin embargo para el control de iluminación de una residencia es una opción viable.

3.3.2 Implementación de dispositivos X10

Un dispositivo que funcione bajo el protocolo X10 puede ser un transmisor, un receptor o ambos (transceptor). En términos prácticos un dispositivo con la capacidad de recepción de comandos X10 debe ser capaz de detectar los cruces por cero de la señal de CA residencial y a la señal de 120 [kHz], mientras que un dispositivo transmisor debe poseer un generador de señales de la misma frecuencia y un método para montarla.

Una forma sencilla de implementar un detector de cruces por cero es utilizando un microcontrolador que posea diodos de protección en sus terminales de entrada que permitan sujetar el voltaje de 127 [V] CA, y así cuando este supere el voltaje de alimentación positivo (V_{DD}) sea interpretado como un '1' lógico, mientras que cuando sea inferior al voltaje de alimentación negativo (V_{SS}) se considere como un '0'. Para limitar la corriente que llega a la terminal del microcontrolador y que esta no supere el valor máximo soportado se conecta un resistor en serie de un valor considerablemente alto, del orden de $M\Omega$. Una vez conectado esto, es posible detectar el cruce por 0 a través de un cambio de estado lógico.

Para la implementación del detector de la señal portadora (120 [kHz]) se requiere un circuito de cuatro etapas. Las primeras dos se encargan de atenuar en su totalidad la señal de 60 [Hz], mediante un circuito de acoplamiento y un filtro pasa altas. La siguiente etapa amplifica la señal de 120 [kHz] para su posterior acondicionamiento en la última, la cual entrega una señal digital interpretable por un microcontrolador. Agregando este circuito al detector de cruces por 0 es posible diseñar un dispositivo que reciba comandos X10 siguiendo el protocolo definido con anterioridad.

Finalmente para transmitir los comandos X10 es necesario poder generar la señal de 120[kHz] y montarla en la de 60 [Hz] y 127 [V] CA. Para la generación de la señal existen varias opciones; se puede hacer uso de un oscilador controlado por un microcontrolador o mediante software generar la señal con las características

necesarias. Una vez generada la señal debe ser montada, lo que se realiza mediante dos etapas. La primera de ellas aísla al circuito generador de la señal de CA; la segunda acopla mediante un filtro pasa altas y permite el montaje de forma segura. [12]

3.4 Software

El software encargado de la iluminación realiza principalmente dos tareas: control de la iluminación en tiempo real y el control programado. El proceso de control en tiempo real consiste en la recepción de los comandos que el servidor web envía a través de instrucciones PHP, transmitidas directamente hacia del socket TCP que conecta al demonio *MOCHAD*.

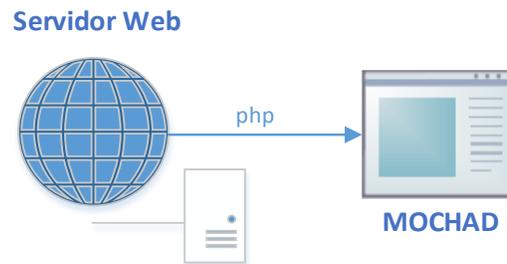


Ilustración 3.5. Proceso de control de iluminación en tiempo real.

La sintaxis de una línea ejecutable por *MOCHAD* consiste en:

1. Código de medio de transmisión, que puede ser:
 - pl – Línea eléctrica.
 - rf – Radio frecuencia.
 - tx – Transmitir.
 - rx – Recibir.
2. Código de casa (a-p).
3. Código de unidad (opcional) (1-16).
4. Instrucción X10.

Ejemplos de instrucciones hacia *MOCHAD*:

- `pl h 10 off`. Apaga el dispositivo de iluminación con código de casa “h” y de unidad “10” conectado a la línea eléctrica.
- `rf a all_units_off`. Apaga todos los dispositivos de iluminación con el código de casa “a” conectados a través de señales de radio frecuencia.

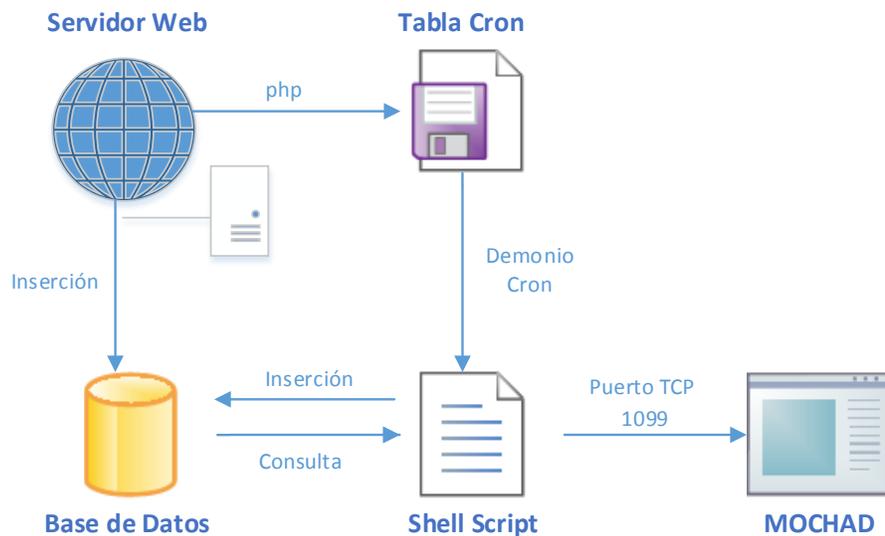


Ilustración 3.6. Proceso de control de iluminación programado.

El proceso de automatización de tareas de control de iluminación mostrado en la ilustración 3.6, inicia con la inserción en la base de datos por parte del servidor web de las acciones programadas y los dispositivos que las realizarán, asignándoles un identificador único de acciones. De la misma forma, el servidor escribe en la tabla **cron** el horario programado, indicando la fecha, hora y el identificador de acciones almacenado también en la base de datos. Cuando las condiciones de fecha y hora se cumplen, el demonio cron ejecuta un **shell script** que se comunicará con **MOCHAD** a través del **puerto TCP 1099**. El **script** consulta las acciones a realizar y los dispositivos asociados utilizando el identificador de acciones que recibe como parámetro desde la tabla cron, generando uno varios comandos de **MOCHAD**, además de actualizar en la base de datos el campo que almacena la fecha y hora de la última acción realizada exitosamente.

Una problemática que podría presentarse es la interrupción inesperada de energía eléctrica, que impida llevar a cabo una o varias de las acciones automatizadas. Si

por ejemplo un usuario programara el apagado de todas las lámparas de la residencia a cierta hora para ahorrar energía, y él no se encontrara para presenciar un corte en el suministro que impidiera al sistema apagar las lámparas, al momento de reanudarse la energía, el estado de los dispositivos automatizados regresaría al que fue momentos antes de la interrupción. De ser así el usuario se encontraría con que el sistema no cumplió con las tareas que él le había asignado en un principio. Esta situación evitaría que se lograra el objetivo del usuario al hacer uso del sistema, en este caso el ahorro energético.

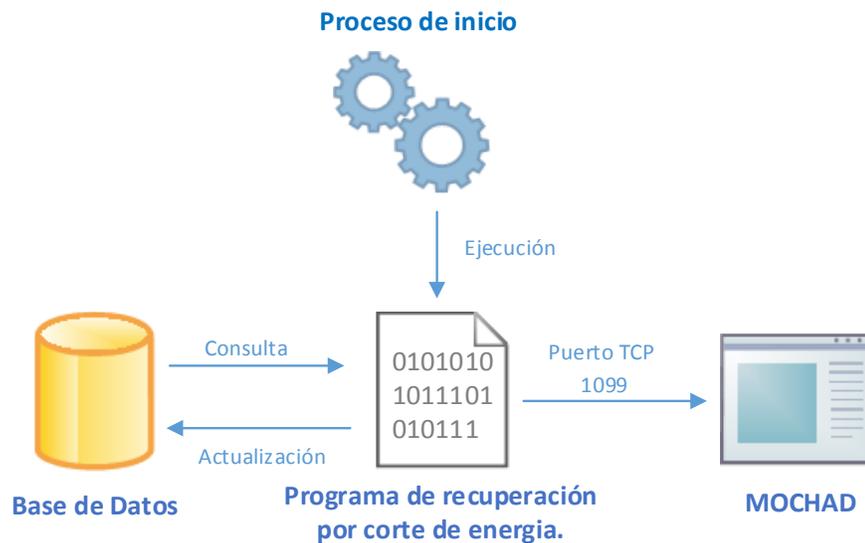


Ilustración 3.7. Proceso de recuperación posterior a un corte de energía.

Para mitigar los efectos de interrupciones de energía eléctrica el sistema realiza una rutina compleja al inicio, que comprueba cuales tareas programadas no fueron llevadas a cabo desde el momento de la última acción exitosa, para después obtener los estados finales de los dispositivos después de una ejecución hipotética de las tareas no ejecutadas. Posteriormente las envía secuencialmente al demonio *MOCHAD* a través de un socket TCP conectado al puerto 1099, en el orden que debieron ser ejecutadas. Finalmente y al igual que en el proceso de automatización actualiza la fecha de la última tarea realizada exitosamente.

Capítulo 4: Unidad de Detección de Intrusión y Videovigilancia

4.1 Objetivo.

La Unidad de Detección de Intrusión y Videovigilancia tiene dos objetivos principales. El primero de ellos es proporcionar a la unidad central un servicio de transmisión de video que el usuario pueda consultar a través de la aplicación web con el fin de monitorear los espacios que conforman la residencia y sus alrededores. El segundo consiste en alertar sobre cambios de estado en un sensor de presencia, pudiendo ser activados y desactivados en tiempo real o de forma automatizada desde la interfaz. La notificación de estos sensores será a través del envío de mensajes de texto a un número configurado previamente a través de la interfaz de usuario.

4.2 Estructura.

Esta unidad se compone de dos subsistemas. El subsistema de videovigilancia está formado únicamente por cámaras IP. Esta tecnología integra una computadora a una cámara de video, dándole la capacidad de soportar un servidor web que permita la interacción con los dispositivos que conforman la cámara (motores, lentes, fuentes de luz, etcétera) y principalmente la transmisión de video en tiempo real.

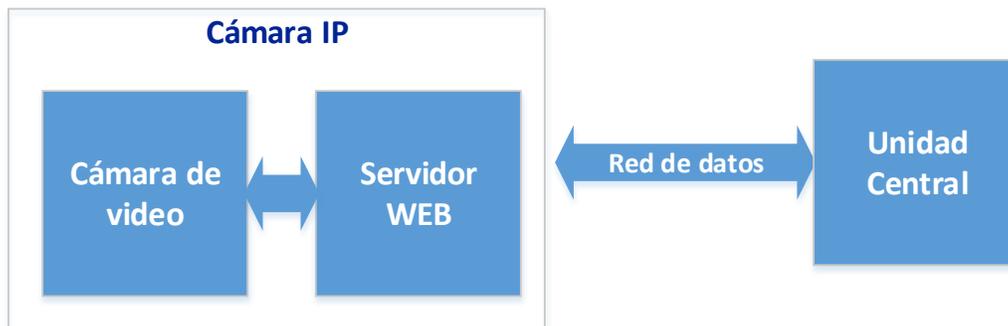


Ilustración 4.1 Estructura de la unidad de videovigilancia.

El subsistema de detección de intrusión se conforma de dos tipos de dispositivos: un transceptor CM15A y uno o varios detectores de movimiento X10. Al detectar el movimiento, los sensores envían una señal al transceptor, usando la variante de radiofrecuencia del protocolo X10 (X10 RF), el cual a su vez le comunica estos eventos a la unidad central. Esta, a través de MOCHAD captura la señal, y validándose del

software de **voz sobre IP** y mensajería instantánea Skype, alerta al usuario mediante un mensaje de texto corto (SMS).

El detector de movimiento X10 es un sensor infrarrojo pasivo, ya que no cuenta con ningún emisor de luz; en cambio capta la radiación infrarroja emitida por los objetos situados en su campo de visión.

Se decide utilizar el software Skype™ para entregar las notificaciones, pues permite el envío de mensajes de texto a teléfonos móviles sin necesidad de que el receptor cuente con una conexión a internet. El usuario puede configurar los números a los cuales desea recibir las notificaciones.

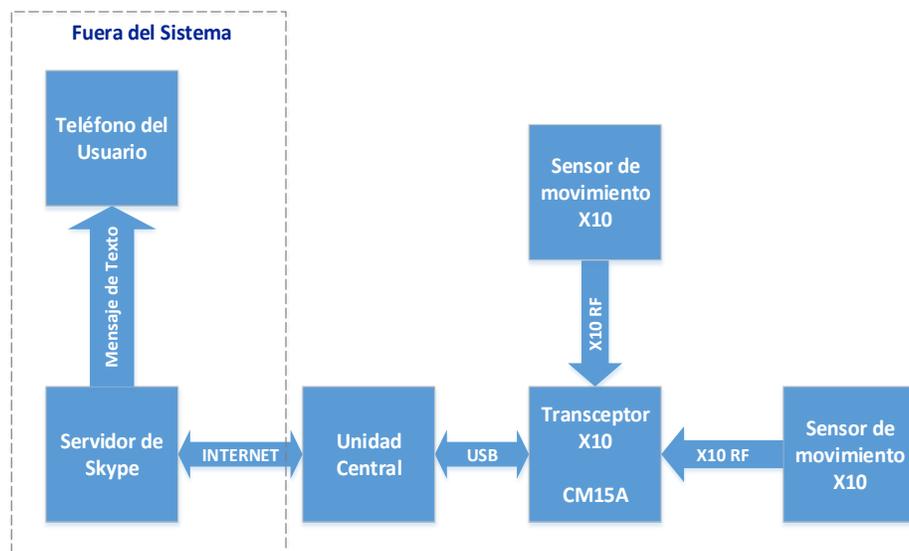


Ilustración 4.2 Estructura de la Unidad de Detección de Intrusión.

4.3 Sensores.

Como se mencionó anteriormente los sensores son parte fundamental de un sistema domótico pues proporcionan señales de entrada que permiten llevar a cabo acciones automatizadas a través los actuadores del sistema sean estos hardware o software.

El concepto de sensor involucra a aquellos dispositivos que ante la presencia de ciertos estímulos externos proveen de una respuesta proporcional a la entrada, permitiendo medir propiedades físicas. [15]

Existen una gran variedad de sensores que son capaces de medir distintas variables físicas mediante diferentes técnicas. De este gran conjunto los de uso doméstico más comunes son los detectores de gas y humo, medidores de temperatura y humedad, sensores de impacto, sensores de movimiento entre otros.

En el caso particular de este proyecto se utiliza un sensor de movimiento para detectar posibles intrusiones y posteriormente notificar al usuario. Existen dos tipos de detectores de movimientos: pasivos y activos. Los activos basan su funcionamiento en la emisión de alguna clase de energía (luz infrarroja, ultrasonido, microondas, etc.) y en el monitoreo del patrón de reflexión; si algo perturba este patrón el sensor identificará esta perturbación. Los pasivos por otro lado, simplemente detectan energía emitida por el intruso, generalmente la radiación infrarroja emitida en consecuencia de su temperatura corporal.

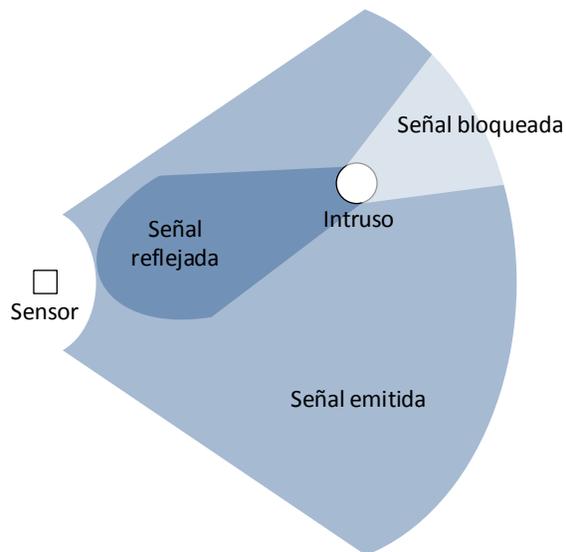


Ilustración 4.3: Funcionamiento de un sensor de movimiento activo.



Ilustración 4.4: Funcionamiento de un sensor de movimiento pasivo.

4.4 Protocolo X10 RF.

Con el fin de aprovechar las capacidades que ofrece el transceptor X10 CM15A, el cual es capaz de procesar señales de radio frecuencia, se decidió hacer uso de dispositivos compatibles. El protocolo X10 utilizado para dispositivos inalámbricos, tales como detectores de presencia y controles remotos es similar al protocolo NEC IR, usando ampliamente en controles remotos infrarrojos variando únicamente en la frecuencia de la señal portadora (de 38 [KHz], a 310 [MHz] para Norteamérica y 433.92[MHZ] para el resto del mundo) y en el medio de transmisión utilizado.

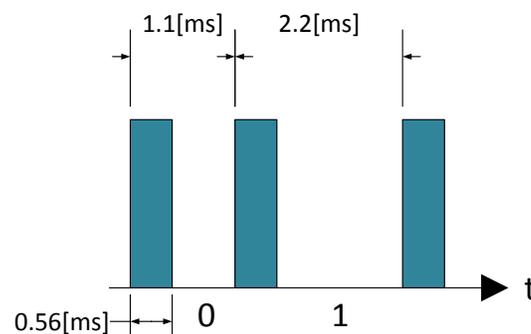


Ilustración 4.5: Modulación por posición de pulso.

El protocolo X10 RF utiliza la técnica de modulación por posición de pulso, que consiste en la variación del tiempo que existe entre un tren de pulsos de la señal portadora y el siguiente. Para este caso, la representación de un valor “0” binario se realiza con un espacio de tiempo de 1.1 [ms] entre cada tren de pulsos, mientras que el valor “1” se representa con un espacio de tiempo de 2.2 [ms], como se observa en la ilustración 4.5.

Cada trama de datos que define el protocolo X10 RF comienza con un tren de pulsos de 310 o 433.92 [MHz] de frecuencia y 8.8 [ms] de duración, seguido de un silencio de 4.4 [ms], lo que se conoce como “código líder”. A continuación se envían los bits que corresponden a la información que desea transmitirse, de acuerdo a la técnica de modulación antes mencionada.

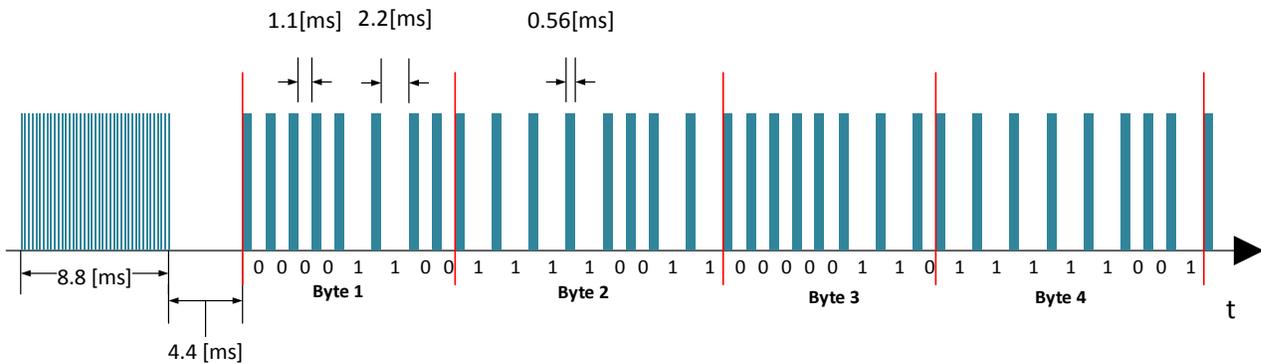


Ilustración 4.6: Ejemplo de una trama de datos X10 RF.

Los bytes 2 y 4 representan los complementos de los bytes 1 y 3, lo que permite a cada trama tener la misma duración y verificar posibles fallos al momento de transmitir los datos.

Para fines de este trabajo se considera al byte 1 como el primero en ser transmitido comenzando por su bit más significativo. Se considerará para todos los casos al bit menos significativo como bit 0.

Los comandos disponibles para la versión estándar del protocolo X10 RF son cuatro: apagar, encender, incrementar brillo y decrementar brillo. Cabe destacar que los dos últimos por definición de las tramas no permiten la selección de una unidad en específico, así que la unidad que responde a estos comandos es la última que recibió una orden X10.

El byte 1 está formado de la siguiente forma:

- Bit 7: Siempre 0.
- Bit 6: Siempre 0.
- Bit 5: Siempre 0.
- Bit 4: Es el bit 1 del número de dispositivo expresado en binario en caso de ser un comando de “Encendido o Apagado”. En caso contrario el valor es ‘1’.
- Bit 3: Es el bit 0 del número de dispositivo expresado en binario en caso de ser un comando de “Encendido o Apagado”. En caso contrario el valor es ‘1’ si el comando enviado es “Decrementar brillo”.
- Bit 2: ‘1’ para el comando “Apagar” y ‘0’ para todos los restantes.
- Bit 1: Es el bit 3 del número de dispositivo expresado en binario.

- Bit 0: '1' en caso de que el comando sea "Decrementar" o "Incrementar" brillo.

De igual manera el byte 3 tiene la siguiente estructura:

- Bit 7: Siempre 0.
- Bit 6: Siempre 0.
- Bit 5: Es el bit 3 del número de dispositivo expresado en binario en caso de ser un comando de "Encendido o Apagado". En caso contrario no importa el valor.
- Bit 4: Siempre 0.
- Bits 3-0: Corresponden al código de casa.

Para asegurar la correcta recepción de los datos la mayoría de los equipos transmisores envían cada trama un mínimo de 5 veces, con un espacio de tiempo de 40 [ms] entre ellas.

La versión de seguridad de X10 RF varía en dos aspectos principales con respecto a la estándar explicada con anterioridad. El primero es en la relación entre el byte 1 y el byte 2, pues mientras en la versión estándar estos eran complemento el uno del otro, en la versión de seguridad solo los cuatro bits más significativos se niegan del byte 1 al 2; el resto permanece sin cambios.

4.5 Cámaras IP

Una cámara IP es un dispositivo de captación de imágenes que a diferencia de una cámara convencional permite la transmisión de imágenes a través de una red de datos en tiempo real. Esto es posible gracias a que cuentan con una computadora integrada con interfaz de red con la capacidad de soportar la implementación diversos servidores, entre ellos **HTTP**, **FTP**, **SMTP**, etc.

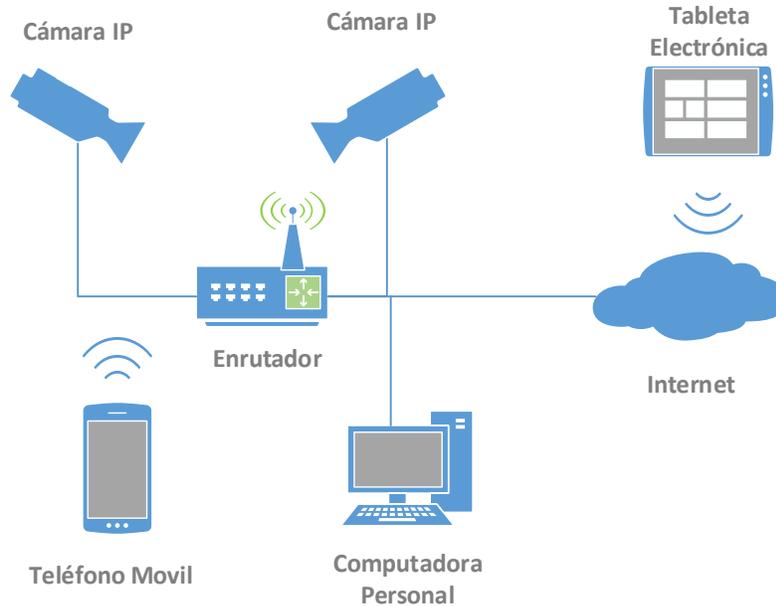


Ilustración 4.7: Estructura de un sistema de vigilancia con cámaras IP.

A continuación se mencionan las ventajas del uso de cámaras IP en comparación con otros sistemas de videovigilancia:

- **Acceso remoto:** Es quizá la mayor de sus ventajas pues ofrece la posibilidad de vigilar un espacio desde cualquier localización que tenga acceso a la red.
- **Multiplataforma:** El incorporar un servidor web permite el acceso al servicio desde una multiplicidad de dispositivos cuyo único requerimiento es contar con una interfaz de red y un navegador web.
- **Costo reducido de instalación:** Al contar con tecnologías como Wi-Fi y **POE** (Power Over Ethernet) el costo de instalación del sistema de video se reduce drásticamente pues no es necesario alterar la infraestructura existente añadiendo cableado adicional.
- **Facilidad de instalación:** Por lo mencionado en el punto anterior la dificultad de instalar cámaras IP es relativamente baja, mientras su configuración es bastante sencilla.
- **Escalabilidad:** Como cualquier dispositivo de red, el añadir cámaras al sistema no implica un cambio significativo en la red local.
- **Servicios adicionales:** Las modernas cámaras IP ofrecen varios servicios y funcionalidades, como envío automático de fotografías a través de correo

electrónico, grabación descentralizada de video, detección de movimiento entre otros.

En contraste, algunas desventajas en comparación con otros sistemas de videovigilancia son:

- Mayor costo por unidad: Las cámaras IP suelen tener un costo mayor a las de circuito cerrado de televisión (CCTV) llegando a costar hasta dos veces más.
- Posible vulnerabilidad de seguridad: Como en cualquier dispositivo conectado a una red de datos, si no se emprenden medidas adicionales de seguridad, el riesgo de intrusión es elevado, lo cual implicaría una brecha de seguridad importante.
- Menor calidad de video: Al utilizar una red de datos para transmitir imágenes, las cámaras IP necesitan un gran ancho de banda por lo que hacen uso de protocolos de transmisión con compresión reduciendo con esto la calidad del video.

Para complacer los objetivos y exigencias de este proyecto es necesaria la capacidad de transmisión remota y multiplataforma de las cámaras IP, por lo que se eligieron a pesar de sus posibles desventajas, las cuales sin embargo se pretenden mitigar llevando a cabo medidas de seguridad preventivas que impidan el acceso a usuarios no autorizados.

4.6 Software

El software desarrollado para esta unidad está conformado por dos programas: uno de ellos se encarga de monitorear el puerto TCP 1099 y realizar el posterior envío de mensajes de texto. MOCHAD retransmite a todos los programas conectados al socket los mensajes provenientes del transceptor X10, entre los cuales se encuentran las alertas de seguridad de los sensores. Una vez que el programa detecta que el mensaje recibido es una alerta, verifica en la base de datos que el sensor que enviado la alerta se encuentre activado; de ser así realiza una consulta adicional la base de datos para obtener el o los números que el usuario ha

configurado con anterioridad. Por cada uno de estos, lleva a cabo el envío de un mensaje de texto que contiene la alerta para el usuario.

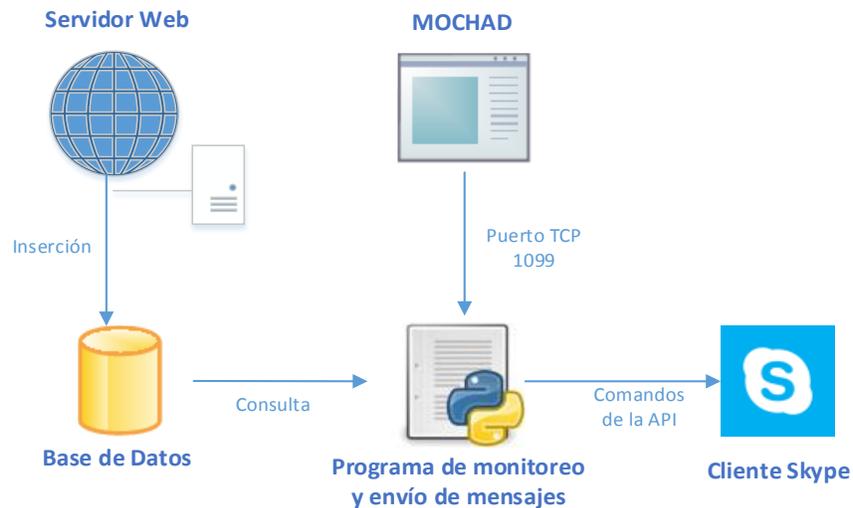


Ilustración 4.8: Proceso de envío de mensajes de texto.

Como se aprecia en la ilustración 4.8, para poder interactuar con el cliente de Skype™, el programa de envío de mensajes utiliza una **API** llamada *Skype4Py*.

Creada para ser utilizada con el lenguaje de programación *Python*, esta interfaz de software tiene como fin permitir que diversas aplicaciones puedan hacer uso de los servicios que ofrece Skype™, tales como llamadas de voz, acceso a los contactos, mensajería instantánea y envío de mensajes de texto, entre muchos otros. De esta forma el sistema es capaz de entregar los mensajes de alerta al usuario sin depender de que este cuente con una conexión a internet en su dispositivo móvil.

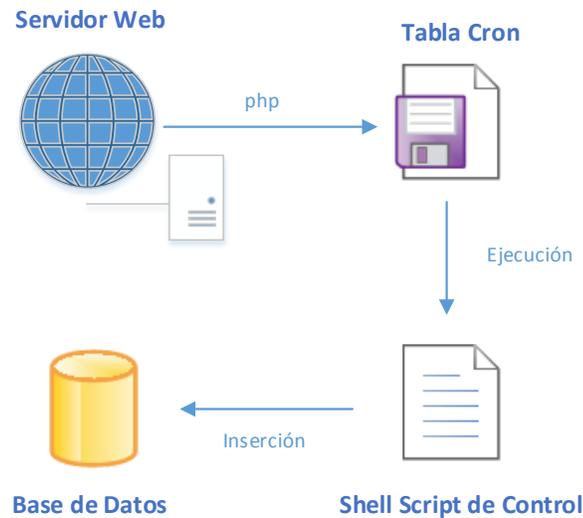


Ilustración 4.9: Proceso de activación y desactivación automática.

El segundo programa se encarga de la activación y desactivación programada de los sensores. Como se mencionó antes, que un sensor se encuentre activado o desactivado depende de un valor almacenado en la base de datos asociado a ese sensor en particular. Para cambiar el estado de activación del sensor de forma automática se utiliza la tabla cron mencionada en el capítulo anterior; esta ejecuta un *Shell script* que cambia el valor de activación almacenado del sensor en cuestión, proceso similar al que ejecuta la unidad de automatización de iluminación.

Capítulo 5: Unidad de gestión de llamadas a la puerta.

5.1 Objetivo y justificación.

Producir un evento detectable por la unidad central que desencadene un servicio que permita interacción audiovisual (local o remota) entre el visitante y el residente. La unidad de gestión de llamadas a la puerta también deberá permitir la apertura remota de la puerta de entrada de forma cómoda, sencilla y segura. Lo anterior representa un mecanismo seguridad, pues el usuario sabe de antemano quien llamó a la puerta y en consecuencia puede tomar diversas acciones, que van desde simplemente observar a la persona, interactuar con voz, hasta permitir su ingreso abriendo la puerta de entrada de la residencia. Estas acciones no dependen de la localización del usuario permitiendo abrir en caso de olvido de llaves, abrir a un invitado inesperado, simulación de presencia, etcétera.

Para lograr la interacción audiovisual se utiliza el software propietario Skype™ pues está disponible para una gran cantidad de plataformas y es bien conocido por ofrecer el servicio de videollamadas de forma gratuita y confiable.

5.2 Estructura.

La unidad de gestión de llamadas a la puerta consta de los dispositivos orientados a la realización de llamadas de video (cámara web, micrófono y altavoz), el botón de timbre y la cerradura con capacidad de apertura eléctrica.

Cuando un visitante presiona el botón de timbre, este dispara un evento en la unidad central, la cual mediante los dispositivos de videollamada la lleva a cabo hacia la cuenta de Skype™ definida en la aplicación web. Esta videollamada es el mecanismo que permite conocer a la persona que ha llamado a la puerta. De manera adicional, si el usuario lo decide, bastará con teclear un comando mediante una conversación instantánea asociada a la videollamada para abrir la puerta.

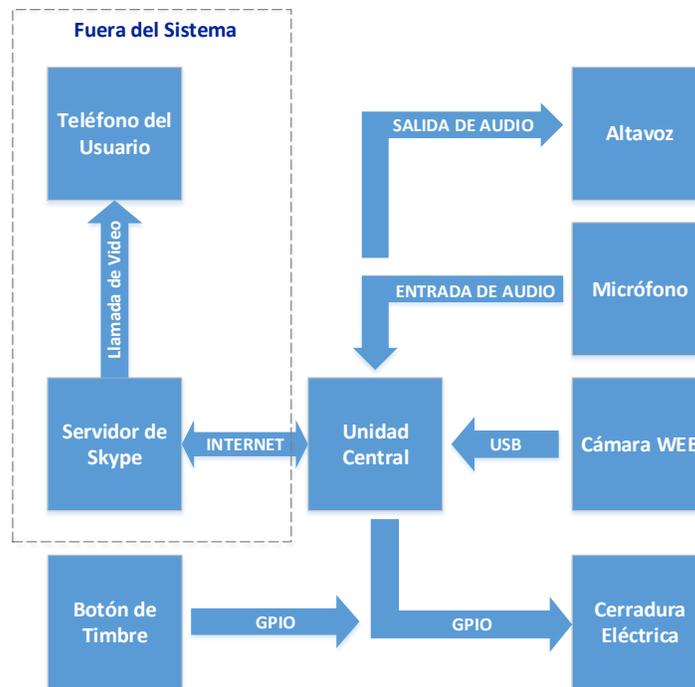


Ilustración 5.1: Estructura de la unidad de gestión de llamadas a la puerta.

5.3 Botón de timbre y cerradura eléctrica.

El botón de timbre proporciona la señal de entrada al sistema para llevar a cabo todas las funciones antes descritas siendo así el responsable de iniciar los servicios que ofrece la unidad.

Tanto el botón de timbre como la cerradura eléctrica se enlazan a la unidad central a través de los puertos denominados de entrada y salida de propósito general (por sus siglas en inglés *GPIO*).

5.3.1 Puertos de propósito general de entrada y salida

De forma general los *GPIO* son terminales de un circuito integrado que describen comportamientos de entrada y de salida, y que pueden ser controladas en cualquier momento. Son la forma más básica de interfaz de computadora haciendo posible a través de ellos la implementación de interfaces más complejas, desde funcionar como interruptores para el encendido o apagado de periféricos hasta protocolos de comunicación más sofisticados.

Su uso (aunque no exclusivamente) se da en la mayoría de las aplicaciones embebidas al proporcionar un mecanismo sencillo para interactuar con el medio físico a través de múltiples dispositivos de entrada como sensores, interruptores u otros circuitos integrados, o de salida como motores, pantallas de cristal líquido, diodos emisores de luz, dispositivos de audio, etc. Cabe destacar además su capacidad para generar interrupciones detectables.

En el caso de una computadora que cuente con un sistema operativo, el núcleo es el encargado del manejo de los puertos, reservando el uso de los mismos para procesos de hardware internos. Para ello es necesario contar con el módulo del núcleo correspondiente a estas tareas. Los sistemas operativos basados en Linux permiten la manipulación de los *GPIO*'s desde el espacio de usuario a través del sistema de archivos virtual **sysfs**, el cual permite la manipulación de dispositivos de hardware como si se trataran de archivos almacenados en disco.

En el caso concreto de la computadora elegida para implementare la unidad central, la tarjeta *Gizmo*, cuenta con 8 terminales correspondientes a sus *GPIO*'s accesibles a través de la tarjeta de exploración, además de otras características de hardware adicionales, como se aprecia en la ilustración 5.2.

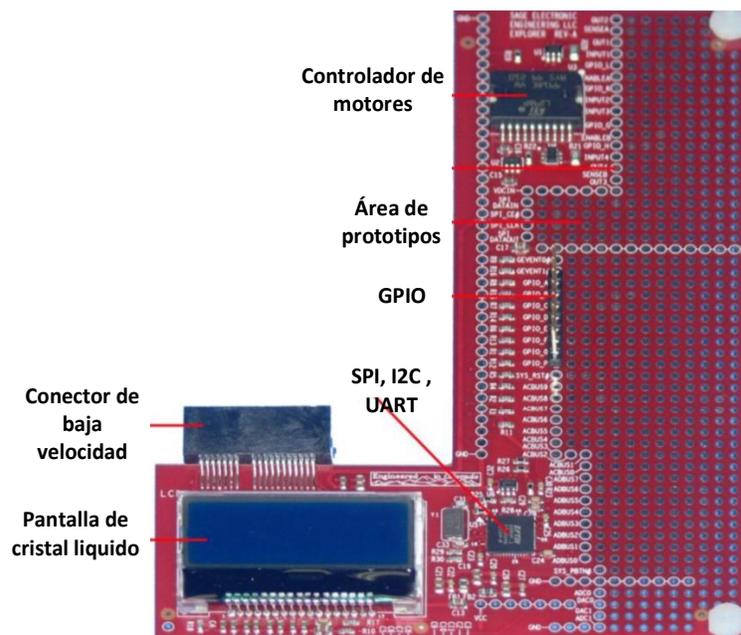


Ilustración 5.2: Vista superior con algunos de los componentes de la tarjeta de exploración.

En el caso particular del *chipset* AMD A55EFCH, utilizado por la tarjeta Gizmo, no se cuenta con un controlador que permita el completo uso de los *GPIO*'s. Por ello fue necesario generar uno, a partir un código parcialmente funcional de un controlador para este *Chipset* encontrado en la red.

5.3.2 Circuitos de interfaz.

El enlace tanto del botón del timbre como de la cerradura eléctrica con la unidad central puede reducirse a la conexión de un par de interruptores con los puertos de propósito general presentes en la tarjeta de exploración. Sin embargo, debido a las condiciones en las que operarán estos dispositivos, es necesaria la implementación de circuitos de interfaz que aseguren su correcto funcionamiento.

La señal necesaria para desencadenar las acciones del videoportero debe cumplir las condiciones de voltaje adecuadas para la unidad central (3.3 [V] para un señal de uno lógico y 0 [V] para un cero). Lo anterior implica una conexión directa a uno de los puertos de entrada/salida de propósito general, lo que sin embargo impediría el funcionamiento del timbre convencional de la residencia, que funciona con 127[V] de corriente alterna. Es por esto que se hace necesaria la implementación de un circuito adicional que ofrezca la posibilidad de manejar un voltaje digital y que a la vez active el timbre.

El elemento principal de este circuito es un *TRIAC* (del inglés *triode of alternating current*) o triodo para corriente alterna, que permite controlar la potencia de un señal de corriente alterna “recortando” la misma. Para ello, este dispositivo cuenta con una terminal conocida como compuerta, capaz de alterar su comportamiento de conducción. Esto es, si existe corriente en la compuerta, el *TRIAC* permite el paso de corriente a través de él en ambos sentidos, y caso contrario se comporta como un circuito abierto, impidiendo el paso de la corriente. Esto lo vuelve un dispositivo de uso adecuado cuando es necesario controlar señales de corriente alterna y generalmente de mayor potencia que las de un controlador digital.

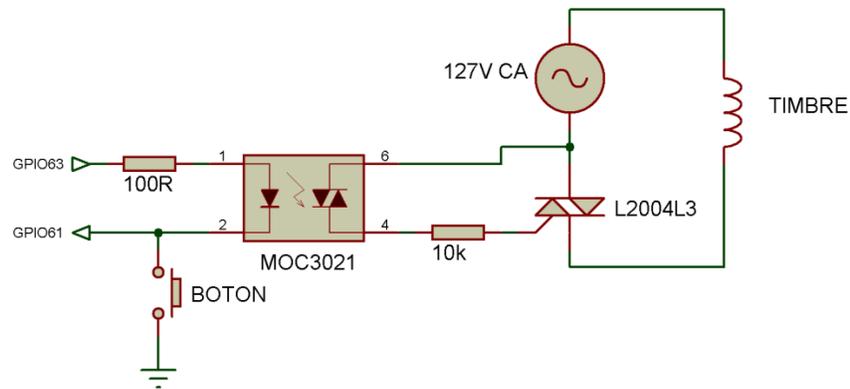


Ilustración 5.3: Circuito de interfaz del botón del timbre.

Una forma de conectar un controlador digital de tal forma que maneje a un *TRIAC*, es mediante un dispositivo adicional, conocido como opto-*TRIAC*. Este dispositivo sirve de interfaz entre señales de corriente directa y alterna, al funcionar mediante un diodo emisor de luz (*LED*) el cual interactúa con un receptor fotosensible que tiene la misma función de la compuerta en un *TRIAC* convencional. Este método presenta la ventaja de aislar las dos partes del circuito previniendo posibles daños al controlador.

Por otro lado, la cerradura eléctrica en específico requiere la conmutación de una señal de 12 [V] de corriente directa para liberar la puerta y permitir el acceso. Para esto, se utiliza un circuito de conmutación basado en un transistor de potencia, dispositivo capaz de interrumpir la corriente que pasa desde la chapa a tierra, encontrándose esta aislada del resto del circuito mediante un **optoacoplador**, de forma similar al circuito controlador del timbre.

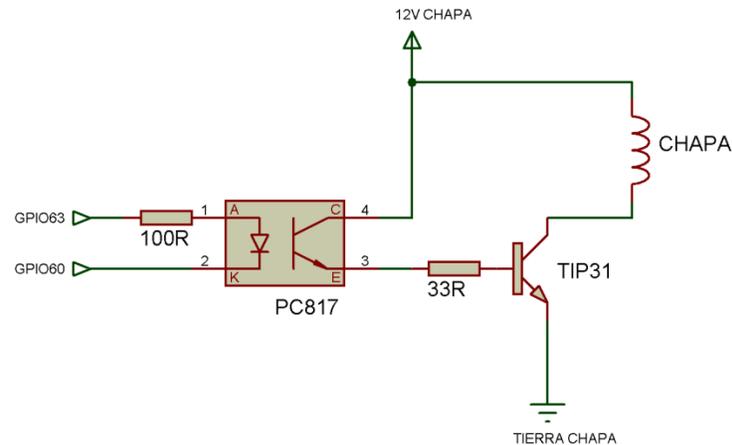


Ilustración 5.4: Circuito interfaz de la cerradura eléctrica.

Cabe señalar que el control tanto del timbre, como de la cerradura eléctrica, se realiza a través de pares de *GPIO*'s. Durante la construcción del circuito se encontró que la unidad central, al apagarse y permanecer polarizada, mantiene voltajes que no pueden ubicarse en un estado lógico determinado, provocando la activación innecesaria e inesperada de los circuitos mencionados. El control mediante un par de *GPIO*'s impide esta situación, manteniendo una diferencia de potencial cero en los diodos emisores de luz infrarroja de los optoacopladores, a pesar del estado de indeterminación lógica. De esta forma se tiene un control pleno de los circuitos de control conectados a los puertos de propósito general.

5.4 Circuitos de Audio.

5.4.1 Amplificador para altavoz

La porción restante de la unidad de gestión de llamadas a la puerta son los circuitos de audio que permiten interactuar con el visitante que ha llamado a la puerta, por lo que representan una parte importante del sistema.

Básicamente consisten en amplificadores de potencia de audio, necesarios para poder conectar a la unidad central un altavoz y un micrófono, considerando

además que son dispositivos cableados que se encuentran a una distancia considerable de la unidad central y no cuentan con un protocolo de comunicación que asegure la calidad de las señales.

El circuito de amplificación de audio para el altavoz se basa en el circuito integrado TA8227P además de los elementos y conexiones especificados en la hoja de datos del mismo. La elección de este circuito integrado se basó en tres criterios:

- Potencia de salida, considerando 2.5 [W] una potencia aceptable, e incluso superior a la mayoría de los intercomunicadores comerciales.
- Voltaje de alimentación, evitando circuitos que demanden alimentación con voltaje positivo y negativo.
- Número reducido de elementos externos, buscando minimizar la complejidad y espacio de la circuitería añadida a la unidad central.

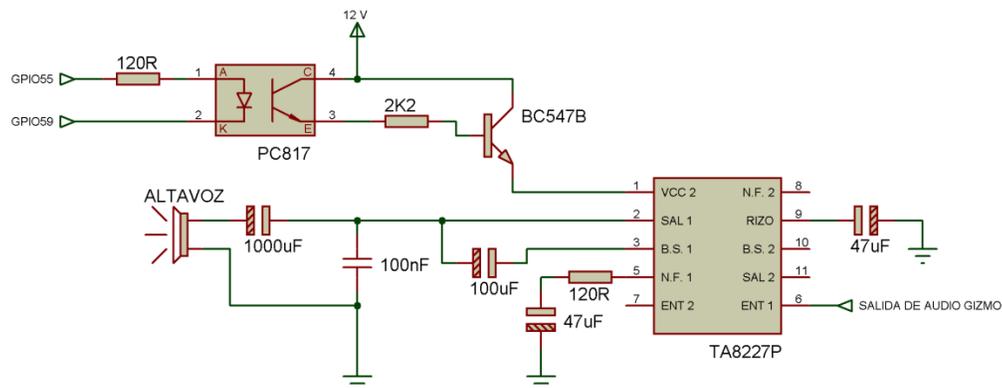


Ilustración 5.5: Circuito amplificador de audio.

Adicionalmente, como se observa en la ilustración 5.5, en la terminal 1 del circuito integrado se conecta un transistor optoacoplado que permite conmutar la señal de alimentación, y así apagar el amplificador cuando no está en uso, lo que evita consumo energético innecesario y posible ruido en el altavoz. Al igual que los circuitos de control del timbre y la cerradura mencionados anteriormente, el control del encendido se lleva a cabo a través de un par de puertos de entrada y salida de propósito general.

5.4.2 Preamplificador de micrófono.

La necesidad de diseñar o no un circuito preamplificador depende de la clase de micrófono a usar. Para esto se deben considerar sus condiciones de trabajo, así como la calidad del sonido deseada y su costo. En el caso de un intercomunicador, el micrófono opera en exteriores, exponiéndolo a las condiciones del medio ambiente como humedad y polvo. Por otro lado la calidad de sonido no es un aspecto crítico, pues simplemente se espera un funcionamiento aceptable en el rango de frecuencias de la voz humana.

Una forma de clasificación de micrófonos es según su **transductor**. Existen micrófonos dinámicos, electrostáticos, piezoeléctricos, magnéticos y de carbón. Los dinámicos o de bobina móvil representan una opción adecuada, pues cumplen con las características antes mencionadas a un bajo costo. Sin embargo, a diferencia de otros, los micrófonos dinámicos no cuentan con etapa de preamplificación, lo que justifica el diseño de un circuito de este tipo.

En la ilustración 5.6 se aprecia un circuito de preamplificación para un micrófono dinámico, que le proporciona a la señal la potencia necesaria para ser transmitida, capturada y posteriormente amplificada por el dispositivo encargado del manejo de audio en la unidad central (códec de audio). Este circuito demostró un mejor desempeño al comparar varios preamplificadores encontrados durante la investigación previa.

Al observar el circuito se aprecia que está compuesto por dos configuraciones de transistores; la primera de ellas, que utiliza al transistor BC549, es una de tipo emisor común, la cual se caracteriza por su alta ganancia de voltaje y por el hecho de que desfasa la señal 180 grados; sin embargo debido a que sus impedancias de entrada y salida se consideran medianas (del orden de $k\Omega$) es incapaz de acoplar su salida a la entrada del códec de audio. El otro transistor se encuentra conectado en una configuración de emisor seguidor, funcionando así como un acoplador de impedancias que evite las pérdidas de potencia al conectarse a la carga, en este caso la etapa de entrada del códec de audio (con una impedancia de 32 $[k\Omega]$). [16]

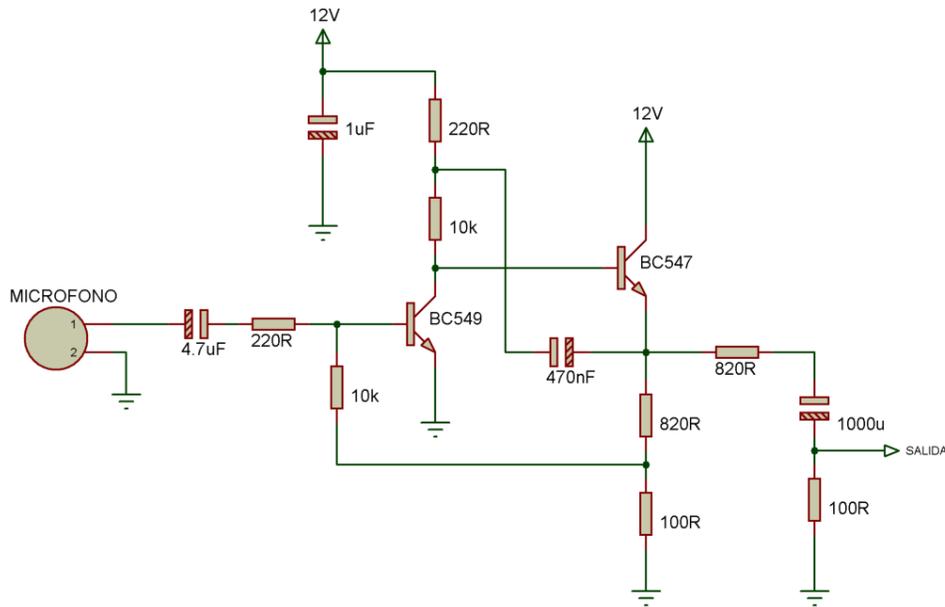


Ilustración 5.6 Circuito preamplificador de micrófono.

En conjunto el circuito tiene una ganancia de voltaje de aproximadamente 200 y una de corriente de 7.65, resultados obtenidos de manera experimental.

5.5 Software.

El software de esta unidad está conformado por dos programas: el primero de ellos se encarga de detectar un cambio de estado en el GPIO de interés, y como consecuencia realiza una consulta a la base de datos para obtener el o los nombres de usuario de Skype™ configurados previamente a los cuales se les realizará la videollamada. Una vez obtenidos procede a comunicarse con el cliente de Skype™ usando la API *Skype4Py* (mencionada en el capítulo anterior), para finalmente realizar la llamada. Si un usuario rechaza esta llamada, no se encuentra disponible o no la atiende en menos de dos minutos, el sistema llama a la siguiente cuenta en la lista y así sucesivamente hasta agotarla.

Por otra parte el segundo programa se encuentra a la espera de un código de apertura de puerta a través del cliente de mensajería instantánea proveniente de un contacto dado de alta mediante la interfaz. Solo un usuario que haya sido

aceptado previamente como contacto a través del cliente Skype™ puede usar los mensajes para enviar el comando de apertura, por lo que no basta con utilizar la interfaz web para añadir las cuentas. Esto brinda dos métodos de seguridad que impiden la apertura de la puerta por usuarios no deseados. En el momento que detecta este código, el programa envía el nuevo estado del GPIO haciendo uso del sistema de archivos virtual sysfs logrando de esta forma la apertura remota de la puerta.

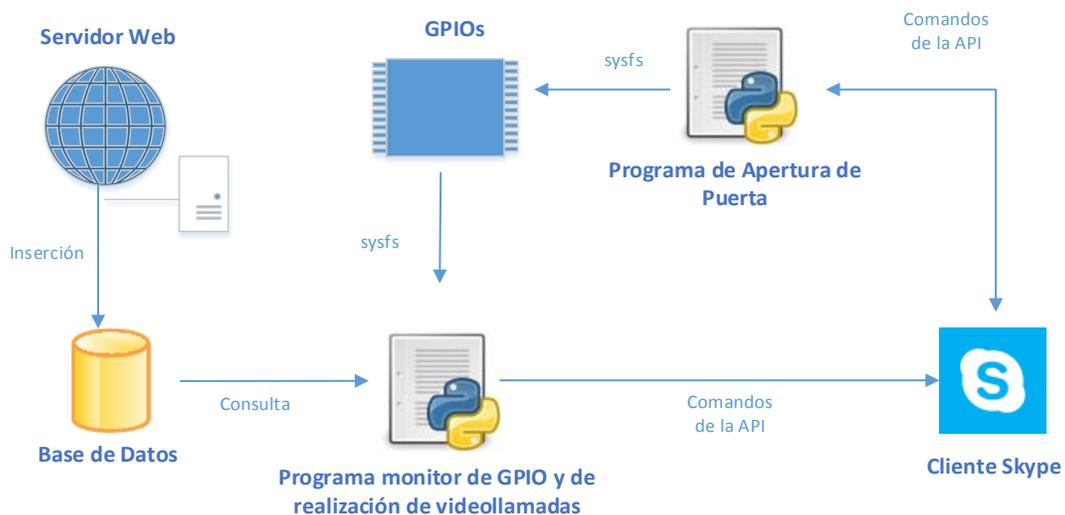


Ilustración 5.7: Proceso de gestión de llamadas a la puerta.

5.6 Construcción e implementación.

Para la implementación del videoportero y debido a que existe una distancia entre la unidad central y el frente, es necesario construir dos tarjetas de circuito impreso adicionales: una ubicada directamente sobre la unidad central, concretamente en la tarjeta de exploración, pasando así a formar parte de ella, y otra en el frente de la casa con interacción directa con los periféricos que conforman al videoportero, con excepción de la cámara web, la cual se encuentra directamente conectada a la unidad central a través de una de sus interfaces USB.

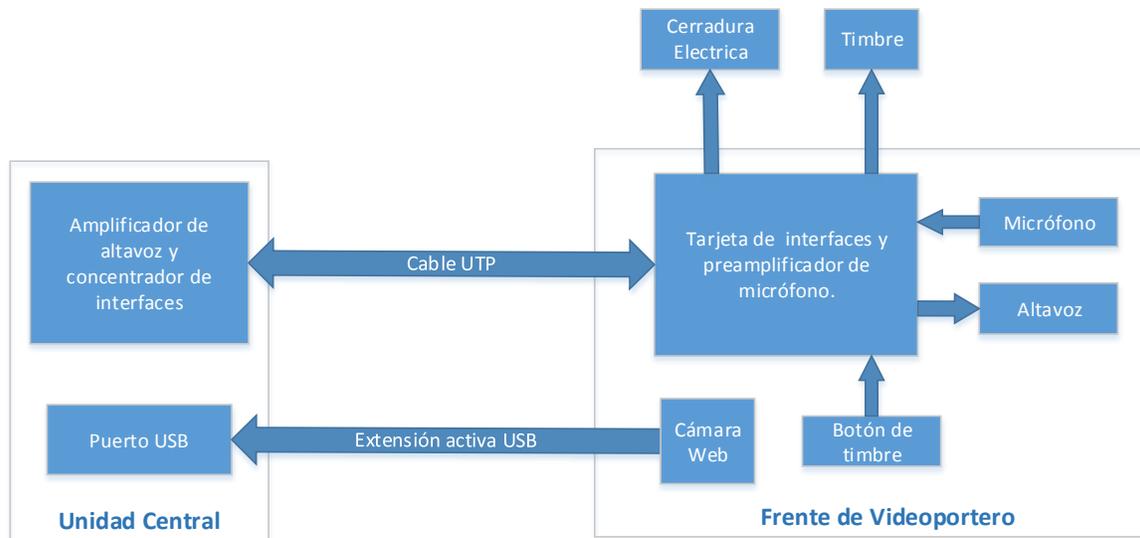


Ilustración 5.8: Estructura de conexiones físicas de la instalación del videoportero.

En las ilustraciones 5.9 y 5.10 se aprecia un modelo virtual en tres dimensiones de las tarjetas de circuito impreso diseñadas y construidas para la implementación de los circuitos descritos anteriormente.

La primera de ellas, como se mencionó, se conecta directamente en la tarjeta de exploración mediante un arreglo de conectores hembra que permite su inserción sin necesidad de soldarla. En ella se incluye el amplificador para altavoz y el circuito que lo controla. Sirve de intermediario entre la unidad central y un conector RJ45, al cual se conectan las señales que se envían desde y hacia la tarjeta de circuito ubicada en el frente del videoportero. Las que corresponden a:

1. Control de activación de la cerradura eléctrica y el timbre (GPIO 63).
2. Control negativo de la cerradura eléctrica (GPIO 60).
3. Señal del micrófono preamplificada.
4. Señal de altavoz, amplificada.
5. VCC del preamplificador (12 V).
6. Tierra de la de unidad central.
7. Señal de estado del botón de timbre (GPIO 61).

Estas señales se transmiten a través de cable **UTP** el cual se conecta a la segunda tarjeta ubicada en el frente de la casa.

Esta segunda tarjeta distribuye las señales de control y polarización de los periféricos, además de aquella que indica el estado del botón de timbre. Contiene a los circuitos de interfaz con este botón, preamplificador de micrófono y cerradura eléctrica. A la vez a esta tarjeta se conectan directamente el micrófono, el altavoz, el timbre, la cerradura y la polarización necesaria para accionar esta última.

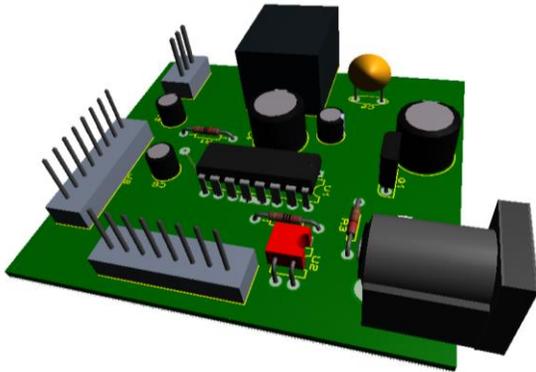


Ilustración 5.9: Tarjeta de interfaces del lado de la unidad central.

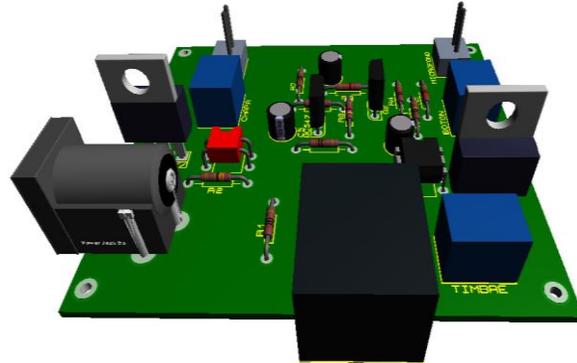


Ilustración 5.10: Tarjeta de interfaces del lado del videoportero.

Un aspecto a considerar es la cámara web necesaria para identificar al visitante. La norma USB 2.0 recomienda no conectar dispositivos a distancias mayores de 5 [m] pues las señales transmitidas tienden a atenuarse y a verse afectadas por el ruido. En respuesta a este inconveniente se utiliza una extensión activa USB la cual reconstruye las señales que pudieran haberse visto alteradas por la atenuación o el ruido en un cable de longitud considerable, en este caso 20 [m], lo que asegura la trasmisión correcta de la imagen proveniente de la cámara web.

Capítulo 6: Aplicación Web.

6.1 Objetivo y Justificación.

Proporcionar al usuario una forma de interactuar con el sistema, que permita: el control de la iluminación en tiempo real y la programación de acciones automáticas con el mismo propósito; el acceso a los recursos que proporcionen los dispositivos de videovigilancia, así como la configuración necesaria para el envío de las alertas desencadenadas por los sensores de movimiento; y la configuración del servicio de gestión de llamadas a la puerta.

Como parte importante en el desarrollo del proyecto, la aplicación web tiene el propósito de servir como interfaz entre el usuario y la totalidad del sistema, en un entorno sencillo y cómodo, capaz de mantener estas características en cualquier dispositivo, móvil o fijo, sin importar el sistema operativo o arquitectura de hardware en que sea utilizada, planteando como único requerimiento un navegador web relativamente reciente.

6.2 Configuraciones previas.

La implementación de la aplicación web que funcionará como interfaz de usuario implica la instalación y configuración *a priori* de algunos servicios necesarios para el funcionamiento de la misma. A continuación se describen.

6.2.1 Servidor Web.

Un servidor web es una aplicación que procesa y responde a las peticiones de un cliente (generalmente un navegador web), en forma de código interpretable a través de los protocolos correspondientes a la capa de aplicación del **modelo OSI**.

El término servidor web suele utilizarse también para designar a los equipos encargados de ejecutar la aplicación de servidor web, sin importar demasiado sus características técnicas. En el caso de este proyecto el servidor web es la unidad central, pues se encarga de recibir las peticiones de los dispositivos que utiliza el usuario para controlar el sistema y de esta forma manipular los diferentes periféricos.

Se ha optado por la utilización del servidor web Apache por sus notables características:

- Se trata de un distribución libre, por lo que su utilización no representa un costo adicional para la implementación del sistema.
- Puede ejecutarse en entornos UNIX y por lo tanto en el sistema GNU/Linux utilizado por la unidad central.
- Es modular y extensible. Lo que implica que solo se instalan las características indispensables para el proyecto y en caso de en el futuro requerir características adicionales, estas simplemente se añaden a la instalación existente.
- Es el servidor web más utilizado en el mundo, lo que se traduce en una gran comunidad que le brinda soporte.

Es importante notar que el servidor deberá solicitar a los usuarios una autenticación de seguridad, nombre de usuario y contraseña, con el fin de que usuarios no autorizados tengan acceso al sistema y con esto puedan controlar los dispositivos y las configuraciones.

6.2.2 Manejador de base de datos.

Un manejador de base de datos es un conjunto de aplicaciones y herramientas que permiten la interacción con una base de datos: añadir, modificar y eliminar usuarios, tablas, listas, procedimientos y registros.

Para la interacción con la base de datos de este sistema se eligió MySQL el cual es el software manejador de bases datos de código abierto más popular. Inicialmente se ofrece bajo la licencia **GNU GPL**; posteriormente si se desea hacer uso comercial de este manejador es necesario adquirir una licencia comercial. Incluye una gran cantidad de módulos que permiten su interacción con varios lenguajes de programación tales como: Java, C, C++, Python, etc. [17]

6.2.3 Manejo dinámico de DNS.

El acceso a la aplicación web ocurre a través de una red de datos, por ello es necesario que la unidad central esté conectada a la misma y que el usuario que

desea hacer uso de la aplicación tenga acceso a red de área local, sin importar su ubicación.

La puerta de entrada a una red de datos de área local es un enrutador. Este dispositivo cumple con varias funciones y en las redes residenciales es prácticamente el único dispositivo encargado de las conexiones de red, en especial la de internet.

Para acceder a un enrutador desde una red externa, es necesario conocer su dirección IP pública, única entre cada dispositivo conectado directamente a internet. Comúnmente los proveedores de acceso a internet de un usuario residencial le proporcionan a este una dirección IP pública dinámica para acceder a internet, lo que significa que cada cierto tiempo esta dirección cambiará. El inconveniente que esto representa es que un usuario debe conocer su dirección IP a lo largo del tiempo si es que quiere acceder a su red local.

En respuesta a esta problemática existen empresas dedicadas al manejo dinámico de **nombres de dominio**, los cuales representan la forma más sencilla para un usuario de acceder a un servicio web al eliminar la necesidad de memorizar direcciones IP. Un ejemplo de estas compañías es *No-IP.com*, la cual se vale de un cliente de software instalado en un equipo de la red local que envía periódicamente la IP pública al servidor de nombres de dominio, logrando que cualquier navegador web obtenga la dirección IP pública de la red local y con esta acceda a ella.

6.2.4 Configuración del enrutador.

Una vez que se tiene acceso al enrutador se vuelve necesario redirigir los paquetes de datos correspondientes hacia la unidad central. Para ello se hace uso de la técnica conocida como redirección de puertos (en inglés *port forwarding*) que permite la asociación de un puerto del enrutador a otro ubicado en un dispositivo específico dentro de la red local. A su vez es necesario, para permitir las conexiones entrantes, añadir una excepción de seguridad **Firewall** si es que se encuentra implementado dentro del enrutador. Concretamente se redireccionarán los paquetes dirigidos al puerto 80 (correspondiente al servicio Web) del enrutador hacia el puerto 80 de la unidad central. De la misma manera para acceder al recurso de video proveniente de las cámaras IP (mencionadas en el capítulo 4) es

necesario redireccionar desde algún puerto de la porción de puertos disponibles hacia la cámara en cuestión.

A pesar del manejo dinámico de **nombre de dominio** para algunos usuarios que se encuentran conectados a la red local es imposible utilizar el mismo nombre de dominio para acceder a la red. En este caso es necesario configurar el enrutador para que funja como servidor local de nombres de dominio, direccionando las peticiones directamente hacia el equipo correspondiente, en este caso la unidad central, en la cual reside la aplicación web.

En la ilustración 6.1 se muestran los servicios requeridos para la aplicación web en su conjunto así como sus formas de interacción.

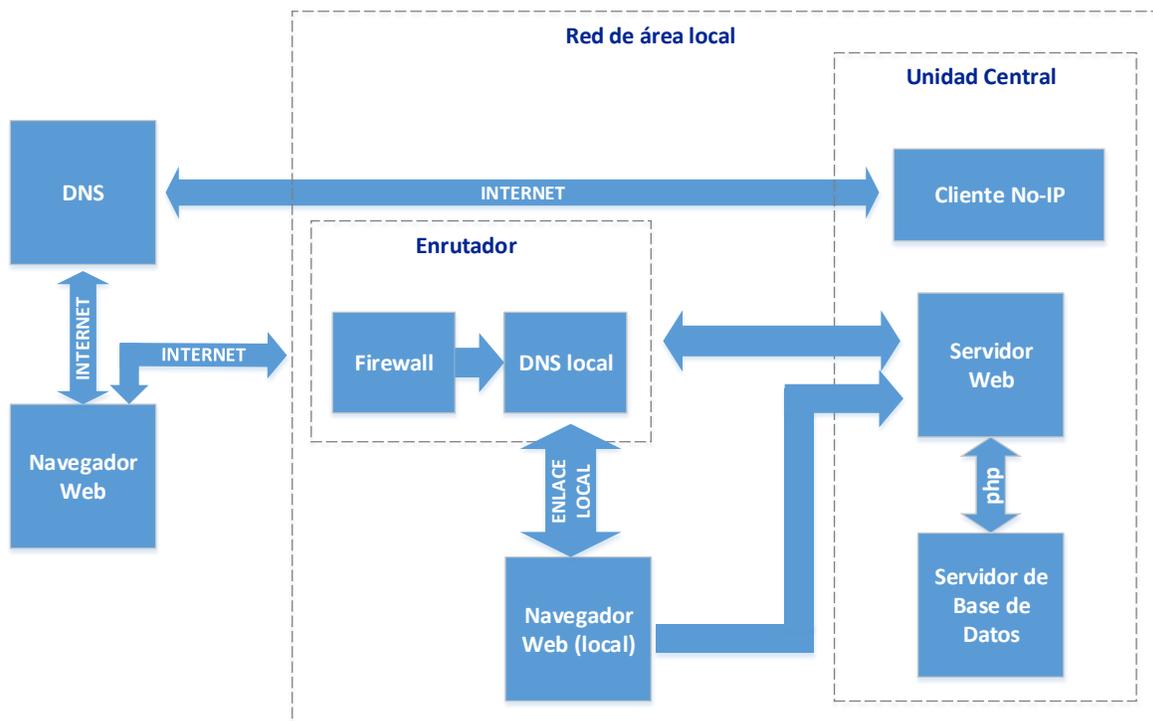


Ilustración 6.1: Servicios requeridos para la aplicación web.

6.3 Tecnologías para la construcción de aplicaciones web.

6.3.1 HTML.

Del inglés *HyperText Markup Language*, es el lenguaje que sirve como base para la construcción de páginas web. Es interpretado por los navegadores web para el despliegue de contenido de texto, multimedia e hipertexto. HTML se basa en la referenciación de recursos lo que significa que dentro del documento no se encuentran los recursos en sí sino referencias a sus ubicaciones que permiten al navegador acceder a ellos.

Cabe señalar que el concepto de página web se refiere en un sentido amplio a los documentos que tienen como único propósito la visualización de información, mientras que una aplicación web tiene un propósito específico distinto a este, como puede ser un editor de imágenes, un gestor de proyectos en línea, videojuegos o como es en este caso, una interfaz de control de un sistema domótico. Todo esto ha sido posible gracias a la aparición de nuevas tecnologías que se han adicionado a HTML para la creación de contenido web dinámico y de mayor complejidad. [18]

6.3.2 PHP

PHP por un lado es un módulo agregable a los servidores web que se encarga de preprocesar un archivo de procesamiento por lotes (*script*) para su posterior envío al cliente en formato HTML. Por el otro lado es un lenguaje de programación de *scripts* ejecutables del lado del servidor, de allí que sus siglas sean el acrónimo recursivo de *PHP: Hypertext Preprocessor*.

PHP permite ampliar la interacción entre un cliente y el servidor web, el cual como se mencionó en el apartado [6.2.1](#), en su forma más básica, únicamente se encarga de responder a las peticiones del cliente sin ir más allá de esta tarea. Al hacer uso de PHP el servidor ejecuta tareas previas a la respuesta que será entregada al cliente, desde una consulta a una base de datos (que posteriormente será parte de la respuesta al cliente) hasta la interacción con aplicaciones del sistema operativo del servidor.

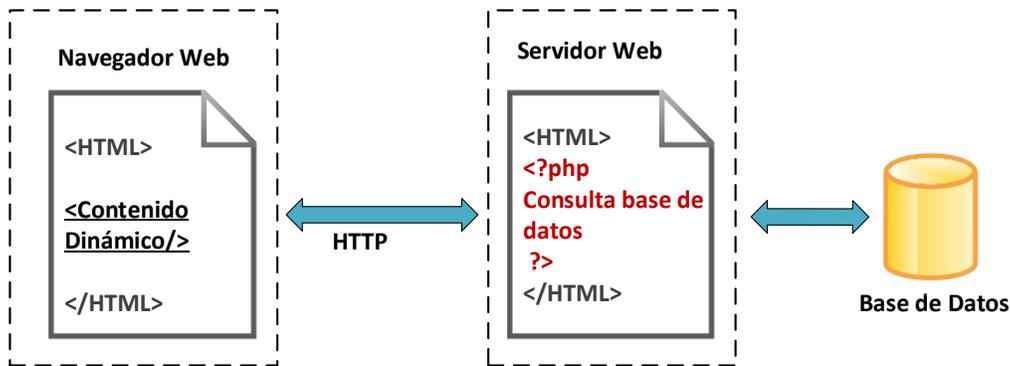


Ilustración 6.2: Ejemplo de funcionamiento de PHP.

6.3.3 AJAX.

Acrónimo de *Asynchronous JavaScript And XML* (JavaScript Asíncrono y XML), es una técnica que permite el intercambio asíncrono de una cantidad mínima de datos, de forma imperceptible, entre el navegador web y el servidor. Esto ofrece la capacidad de recargar parte del contenido de una página web sin la necesidad de recargar la página entera, lo que implica la creación de páginas web dinámicas, rápidas y con transiciones visualmente atractivas.

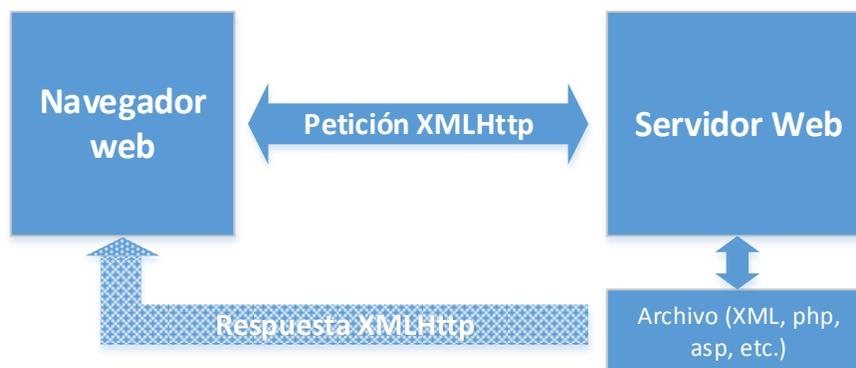


Ilustración 6.3: Arquitectura de comunicación por medio de AJAX.

El uso de AJAX ocurre de la siguiente forma: el navegador elabora una petición de tipo *XMLHttpRequest* la cual puede contener referencia a archivos con información en formato **XML**, texto plano o programas ejecutables del lado del servidor. En el caso de que se solicite un archivo de texto o XML, el servidor responderá con la información contenida en este archivo para el procesamiento posterior por parte del navegador. Por otro lado si la solicitud hace referencia a un programa (PHP,

ASP, etc.) el servidor se encargará de ejecutarlo; en el caso de que este programa contuviera una respuesta para el navegador, esta podría ser interpretada por el mismo, reflejándose en contenido dinámico.

6.3.4 JQuery.

JQuery es una biblioteca de *JavaScript* de **software libre** y código abierto (bajo las licencias MIT y pública general de GNU) que conjunta las funciones necesarias para la creación de páginas web más atractivas a través de manipulación de las partes de la página, manejo de eventos, modificaciones a la hoja de estilo mediante CSS, manejo simplificado de AJAX, efectos de animaciones, etc. [19]

JavaScript se usa para designar al lenguaje de *scripts* interpretado y a su intérprete. Es interpretado del lado del cliente por los navegadores web extendiendo las funcionalidades de interacción de las páginas web. Por ejemplo, es común que antes de enviar un formulario con información capturada por el usuario sea necesaria la validación de la misma; para esto se hace uso de *JavaScript*, que evita el envío incorrecto de datos y una subsecuente validación en el servidor que resulte en una respuesta negativa innecesaria. También permite la manipulación directa de los elementos que conforman el documento, como su posición, su formato, etc.

Por otro lado *CCS (Cascade Style Sheets)* es un lenguaje sencillo que define la presentación y el formato de los elementos de un documento web, por lo que combinado con *JavaScript* es posible conseguir los efectos y animaciones propios de *JQuery*.

6.4 Interfaz de usuario.

Apoyándose en las herramientas anteriores se espera cumplir con los objetivos planteados en un inicio. La interfaz de usuario se construye con un enfoque multiplataforma, es decir, que se pueda asegurar su correcta visualización en dispositivos móviles y equipos fijos. Se utiliza la versión móvil de *JQuery (jQuery)*

mobile) la cual proporciona la certeza de buen funcionamiento en ambos tipos de dispositivos (a diferencia de la versión convencional).

Cada una de las secciones de la interfaz, exceptuando el videoportero (gestor de llamadas a la puerta), está conformada por un conjunto de planos que representan los diferentes niveles de la residencia. Cada uno de ellos muestra la ubicación de los diferentes dispositivos que conforman el sistema domótico permitiendo el control, la programación y el monitoreo de acuerdo a las capacidades de cada uno de ellos. En la parte inferior un menú persistente permite la navegación entre los módulos de la interfaz.

6.4.1 Control y automatización de iluminación.



Ilustración 6.4: Pantalla del módulo de control y automatización de iluminación de la interfaz.

En la pantalla principal de este módulo se despliegan las lámparas controlables, ubicadas simbólicamente en las áreas que les corresponden físicamente dentro de la residencia, y el botón de edición de programaciones. En la ilustración 6.4 se aprecia una muestra de esta pantalla y sus componentes.

Al seleccionar una lámpara el sistema despliega un menú emergente con las opciones de encender, apagar y programar la lámpara en cuestión. La forma de operación de las opciones de encender y apagar consiste en realizar una petición al servidor mediante AJAX, la cual desencadena la ejecución por parte del módulo PHP de un programa que se comunica con MOCHAD a través del puerto TCP 1099.

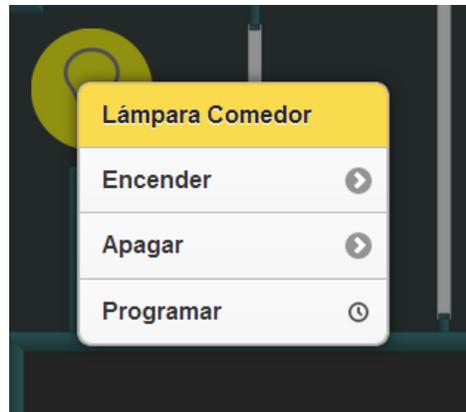


Ilustración 6.5: Menú de acciones por lámpara.

En este caso en particular la utilización de AJAX permite el encendido y/o apagado de las lámparas sin la necesidad de recargar la página. La ilustración 6.6 muestra la interacción del software que hace posible el control de lámparas desde la interfaz de usuario.

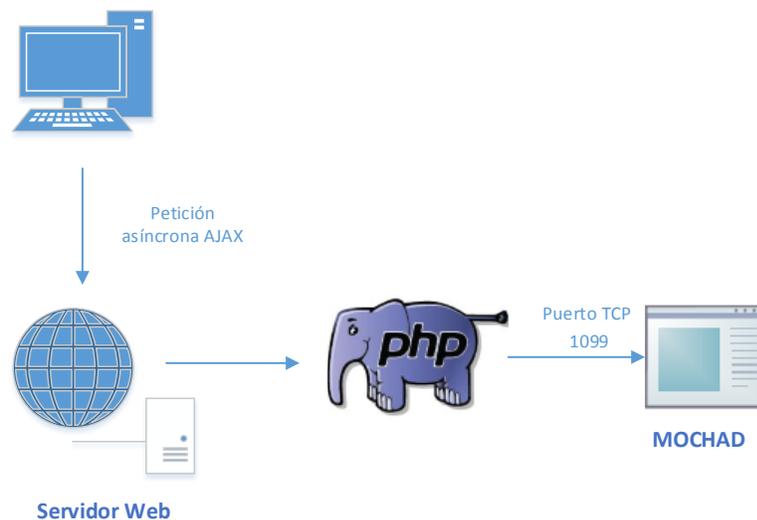
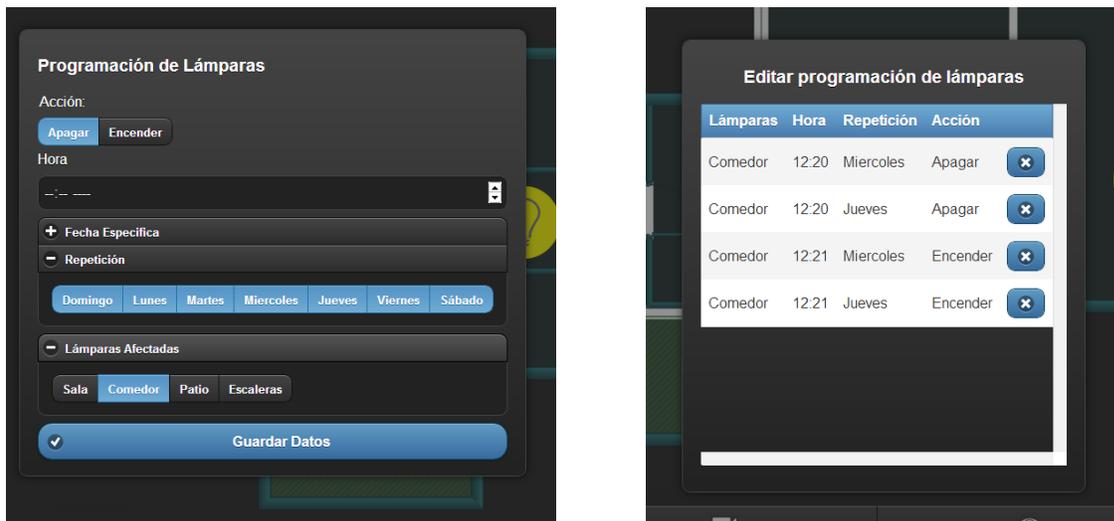


Ilustración 6.6: Estructura del software de control de iluminación desde la interfaz de usuario.

Por otro lado la programación del encendido y apagado de las lámparas se realiza mediante un formulario que solicita al usuario los parámetros de programación siguientes: la acción a desempeñar, la hora en la que se llevará a cabo y las lámparas afectadas por este evento; se ofrece al usuario la posibilidad agendar las acciones deseadas de acuerdo a dos criterios: una fecha en específico o repetición

semanal. Una vez guardada una o varias programaciones se pueden visualizar y eliminar a través del botón de “Editar Programación”, mencionado arriba.



Ilustraciones 6.7 y 6.8: Pantallas de Programación y Edición de programaciones.

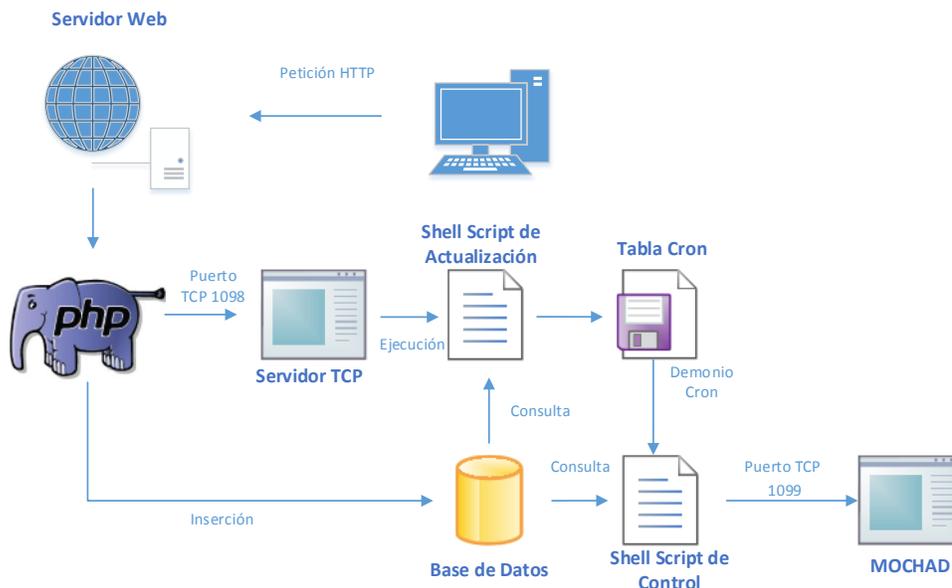


Ilustración 6.9: Estructura del software de programación de los dispositivos de iluminación desde la interfaz de usuario.

La porción del proceso de agregar y eliminar programaciones, ejecutada por la interfaz tiene como objetivo mantener los datos de la tabla cron actualizados y coherentes con lo deseado por el usuario. Para lograrlo realiza el almacenamiento y eliminación correspondiente de las programaciones en la base de datos, y

posteriormente notifica a través del puerto TCP 1098 que la tabla cron debe ser actualizada. A continuación un *Shell Script* vuelca todos los datos contenidos en la base de datos hacia la tabla cron reemplazándola en su totalidad. Esta forma de modificación a la tabla cron evita la necesidad de otorgar permisos adicionales al usuario **www-data**, hecho considerado como un fallo de seguridad que compromete directamente al sistema operativo, y así al resto de la unidad central.

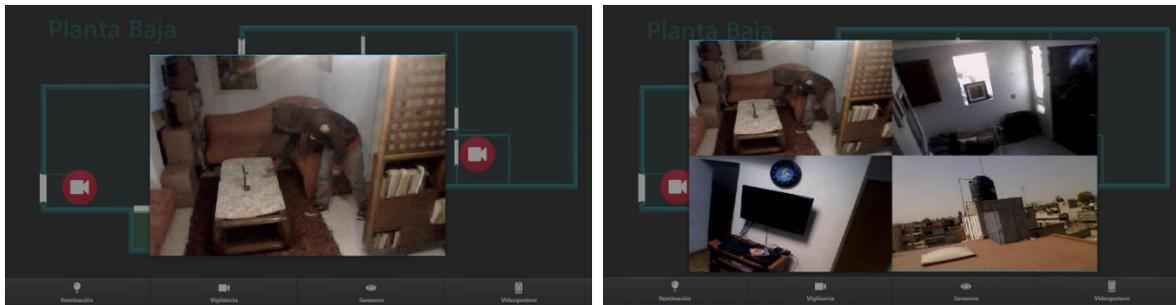
6.4.2 Videovigilancia

El módulo de videovigilancia de manera análoga al anterior muestra la distribución física de las cámaras IP en la residencia; al seleccionar alguna de ellas el usuario es capaz de visualizar el video que la cámara transmite en tiempo real. Para esto la aplicación web hace uso de la capacidad de referenciación de contenido multimedia de HTML, permitiendo visualizar el mismo a pesar de encontrarse en otros servidores web, en este caso cada una de las cámaras IP.



Ilustración 6.10: Pantalla del Módulo de Videovigilancia.

Al presionar el botón “Ver todas las cámaras” se muestra un mosaico con la transmisión simultánea de todas las cámaras.



Ilustraciones 6.11 y 6.12: Pantalla de visualización individual de cámara IP y Pantalla de visualización simultánea de cámara.

6.4.3 Detección de intrusión.

Este módulo al igual que sus predecesores despliega la localización de los sensores, mostrando el estado de activación de cada uno de ellos al momento de cargar la página. En la ilustración 6.13 se aprecia que el sensor que se encuentra en la parte izquierda del plano está activado mientras que el derecho está desactivado.

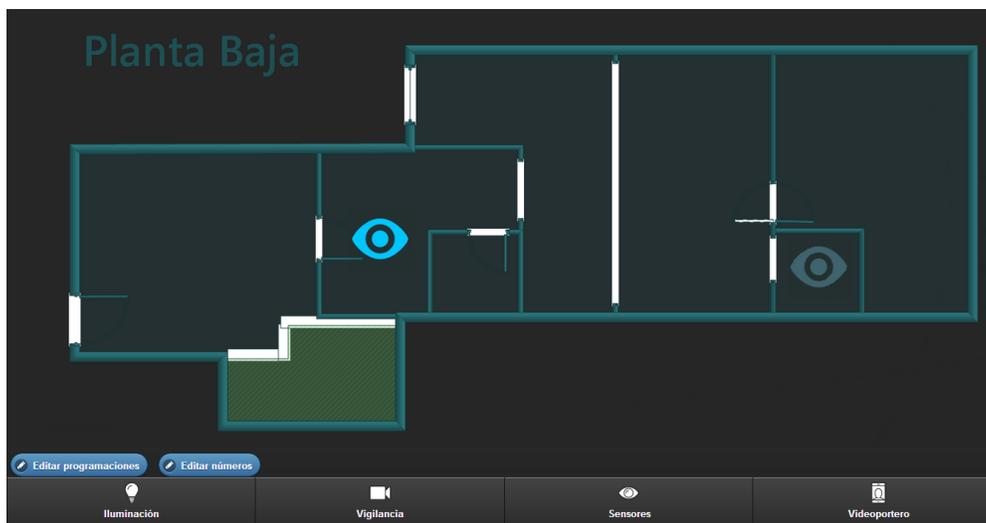


Ilustración 6.13: Pantalla del módulo de detección de intrusión.

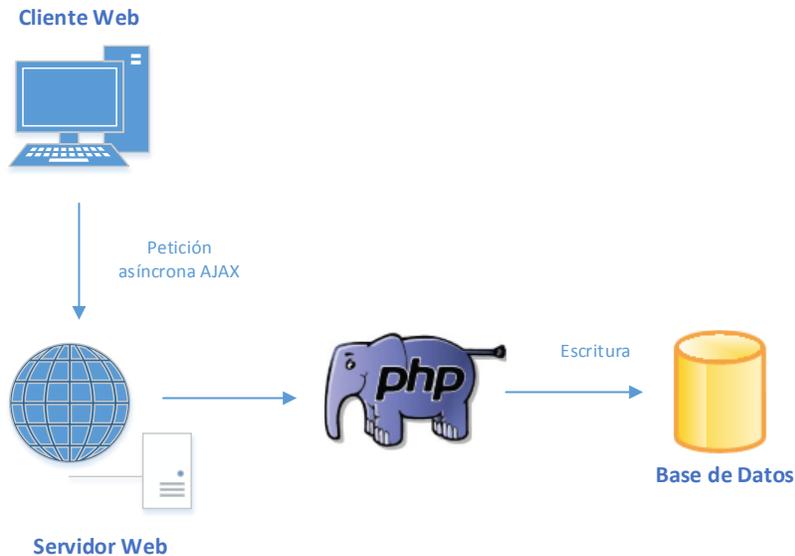


Ilustración 6.14: Proceso de activación y desactivación de los desde interfaz de usuario.

La interfaz de activación, desactivación y programación de los sensores guardan una similitud estrecha con el módulo iluminación (ver ilustraciones 6.5, 6.7 y 6.8). Sin embargo la forma de operación es algo distinta. Para activar o desactivar un sensor en tiempo real basta con realizar una escritura a la base de datos indicando el nuevo valor correspondiente al estado de activación del sensor. Tal y como se aprecia en la ilustración 6.14.

El proceso de programación es de forma similar a la programación del control de lámparas, con la única diferencia que el *Shell script* de control, ejecutado en el horario programado en la tabla cron, no tiene comunicación con MOCHAD sino únicamente con la base de datos realizando las escrituras correspondientes para actualizar el estado del sensor.

Por último, el botón de “Editar números de notificación” muestra un listado de los números telefónicos registrados para la notificación de las alertas de intrusión, permitiendo agregar hasta cinco de ellos, así como modificarlos o eliminarlos de la base de datos.

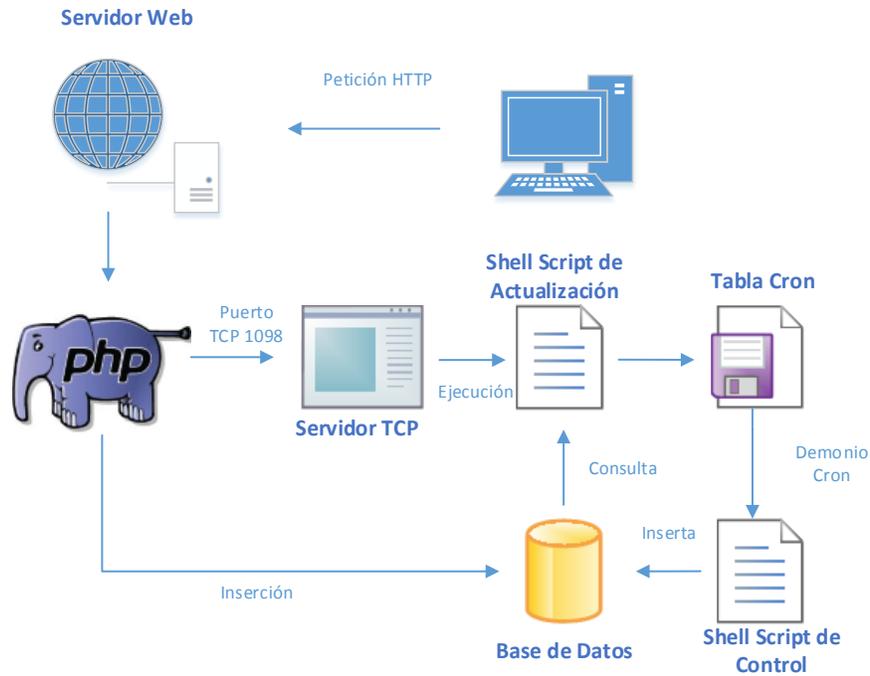


Ilustración 6.15: Proceso de programación de los sensores desde la interfaz de usuario.

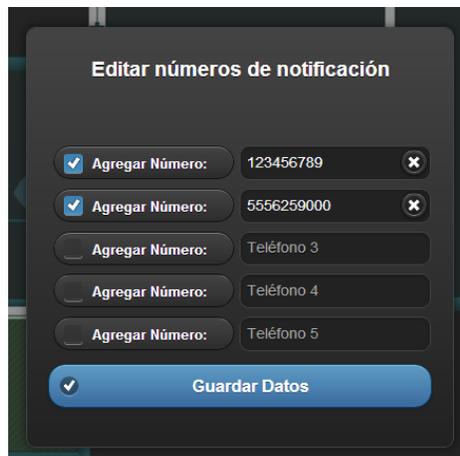


Ilustración 6.16: Menú de edición de números para notificación por mensaje de texto.

6.4.4 Gestión de llamadas a la puerta.

Este módulo puede considerarse el más sencillo de todos, pues únicamente permite agregar, editar o eliminar las cuentas de usuario de Skype™ a las cuales se les realizarán las videollamadas que notifiquen la llegada de un visitante a la puerta.



Ilustración 6.17: Menú de edición de cuentas con acceso y notificación al videoportero.

Capítulo 7: Resultados, conclusiones y trabajo a futuro.

7.1 Pruebas y resultados.

Siguiendo la metodología se vuelve importante realizar una serie de pruebas para verificar la funcionalidad y la compatibilidad de la interfaz de usuario con diversos dispositivos de diferentes fabricantes, resoluciones de pantalla, y navegadores web.

En concreto los dispositivos en los cuales se probó el manejo del sistema son:

- Apple iPod touch generación 5 con iOS7 en los exploradores Google Chrome y Safari, utilizando una resolución de 568 x 320 pixeles.
- Apple iPad generación 1 con iOS5 en el explorador Safari y una resolución de 1024 x 644 pixeles.
- Samsung Galaxy S2 con Android 4.2 (JellyBean), en los exploradores Google Chrome y Mozilla Firefox y una resolución de 534 x 247 pixeles.
- Nokia Lumia con Windows Phone 7 en el navegador Internet Explorer; y con 480 x 320 pixeles de resolución de pantalla.
- Laptop Acer Aspire 4752 con Windows 7 en los exploradores Google Chrome, Internet Explorer, Mozilla Firefox y Safari con una resolución de despliegue de 1349 x 667 pixeles.

En la tabla 7.1 se aprecian los resultados de esta primera prueba de compatibilidad de la aplicación web. Cabe mencionar que en la parte que refiere a la unidad de gestión de llamadas a la puerta, solo es necesario verificar la disponibilidad de una versión de Skype™ para el dispositivo en cuestión, pues de haberla se asegura el correcto funcionamiento; sin embargo, para utilizar todas funciones de interacción es necesario contar con un micrófono en el dispositivo.

Adicionalmente a los equipos mencionados, es importante destacar que se probó la aplicación web en un teléfono Nokia C6 con el sistema operativo *Symbian* y el navegador propio este. Esto resultó en compatibilidad limitada en la función de programación de tareas. De igual manera, al no contar con una versión de Skype™ para este sistema operativo, el videoportero no puede ser utilizado.

Se considera que un equipo es compatible con el sistema si es capaz de realizar todas las tareas con al menos un navegador web.

	Despliegue				Control en tiempo real				Programación de tareas				Despliegue de cámaras				Videoportero
iPad 1G	NA	NA	NA	✓	NA	NA	NA	✓	NA	NA	NA	✓	NA	NA	NA	✓	✓
iPod Touch 5G	✓	NA	NA	✓	✓	NA	NA	✓	✓	NA	NA	✓	✗	NA	NA	✗	✓
Galaxy S2	✓	NA	⚠	NA	✓	NA	✓	NA	✓	NA	✓	NA	✓	NA	✓	NA	✓
PC	✓	✓	⚠	✓	✓	✓	✓	✓	✓	⚠	⚠	⚠	✓	✗	✓	✓	✓
Nokia Lumia	NA	✓	NA	NA	NA	✓	NA	NA	NA	⚠	NA	NA	NA		NA	NA	✓

Tabla 7.1: Resultados de las pruebas de compatibilidad.

- Símbolo Descripción**
- ✓ Funcionamiento correcto.
 - ⚠ Funcionamiento parcial.
 - ✗ Incompatible.
 - NA El navegador no está disponible.

7.2 Conclusiones.

Lograr que un sistema de software sea totalmente multiplataforma es una tarea complicada, pues la diversidad de dispositivos, principalmente móviles, que un usuario podría utilizar para acceder a una aplicación web es enorme. Si bien el sistema es capaz de interactuar con todos ellos, el problema radica en el despliegue de la interfaz. Inicialmente se buscó obtener una interfaz visualmente atractiva por igual en todos los dispositivos, móviles y fijos (aspecto normalmente desatendido en las versiones móviles de los sistemas domóticos), sin embargo las herramientas orientadas a la estandarización y desarrollo de las aplicaciones web limitan hasta cierto punto el proceso creativo. En ese sentido es complicado reducir la brecha entre el diseño deseado, el alcance de las herramientas y las capacidades de despliegue de los diversos dispositivos. Una parte significativa del esfuerzo dedicado al desarrollo de la interfaz se centró en la resolución de este inconveniente, en la búsqueda de un balance entre el uso recomendado de la herramienta y técnicas independientes de desarrollo que permitieran que la aplicación final fuera lo más cercana al diseño inicial, conservando la capacidad de ser visualizado por la mayoría de los equipos. Se concluye entonces que el objetivo planteado en un inicio de obtener una interfaz multiplataforma fue cumplido en su mayoría, a pesar de las excepciones apreciadas en los resultados.

Por otro lado, para conseguir que un sistema embebido sea totalmente automático no basta con que sea capaz de realizar sus tareas programadas de forma automática, sino que tiene que poseer la capacidad de funcionar de manera autónoma y continua. Es decir, tiene que ser prácticamente inmune a los fallos y si alguno se presentase debe ser capaz de tomar las acciones para recuperarse y continuar con su funcionamiento. También debe estar preparado para hacer frente a situaciones externas inesperadas tales como interrupción de energía eléctrica o indisponibilidad de la red de datos. Adicionalmente debe poder comprobar e instalar actualizaciones al sistema operativo o al propio sistema de domótica, de forma automática. Para enfrentar estas problemáticas se implementaron varias acciones. La primera de ellas (descrita al final del capítulo 3) consta de un proceso encargado de recuperar las programaciones que pudieran no haberse realizado debido a un corte de energía eléctrica, respetando lo más posible las tareas que

tuviera programadas. También se tomaron precauciones para que los programas que conforman el sistema pudieran continuar sin problemas a pesar de fallas en la conexión a internet y se evitó que el videoportero atendiera mensajes retrasados que estuvieron en espera mientras la conexión a internet no estaba disponible, esto con el fin de evitar aperturas inesperadas en algún momento de reconexión. Adicionalmente se desarrolló un proceso que monitorea la temperatura de la unidad central en todo momento y en caso de superarse la temperatura crítica definida (80 [C]) alerta de esto al usuario y apaga a la unidad central. Por todo lo anterior, se puede decir que el sistema logra un grado de autonomía considerable y el objetivo se cumple.

Al estar basado en web, el sistema hereda la capacidad para ser accedido, controlado y configurado desde cualquier ubicación con acceso a internet. En particular la unidad de gestión de llamadas a la puerta ofrece servicios que en ningún producto se dan de forma remota, pues generalmente constan de un monitor fijo dentro de la residencia y la interacción con el visitante únicamente se puede realizar mediante este dispositivo.

Siguiendo con el análisis de los objetivos, en cuanto a las características de adaptabilidad y escalabilidad (crecimiento a futuro) cumple con los criterios necesarios, pues por un lado el uso de la tecnología X10 permite una instalación muy poco invasiva de la unidad de iluminación, la cual contempla una porción considerable del sistema; de igual forma las demás unidades también pueden resultar poco invasivas, al utilizar cámaras IP inalámbricas y sensores inalámbricos. Además de este aspecto el sistema es modular, lo que implica que no es necesario instalarlo en su totalidad si el usuario no lo requiere, y si este requiriera en el futuro adiciones, no resulta complicado agregarlas, adaptándose a sus necesidades. La unidad central tiene la capacidad de integrar diversos periféricos de diferentes tecnologías realizando cambios mínimos al sistema. El mayor trabajo que el sistema exigiría para poder adaptarse a diferentes entornos es el desarrollo de la interfaz web, pues es necesario modelar tanto la residencia como los dispositivos que utilizaría el sistema, sin embargo está diseñado para que estos cambios sean lo más sencillos posibles.

En el contexto de la domótica el presente trabajo se distingue debido a que se buscó aprovechar las mejores características de diversas tecnologías para crear un sistema cuyas capacidades superaran a sistemas ya existentes, resultando en uno que posee características poco usuales, destacando la existencia de actuadores implementados no solamente en hardware sino en software, concepto que como tal no existe aún, aprovechando las tecnologías recientes y la tendencia actual hacia los equipos móviles y el acceso remoto a los recursos.

Con respecto al conocimiento aportado por el trabajo para futuros desarrollos, se destaca el relacionado al área de los sistemas embebidos. En concreto en la invitación que se hace al desarrollo de tecnología basada en tarjetas de placa simple modernas de características complejas, cuyo alcance rebaza la idea obsoleta de que los sistemas embebidos tienen capacidades reducidas. Las opciones son cada vez más amplias, pues este tipo de tarjetas se encuentran en auge, provenientes de una gran cantidad de fabricantes. Inclusive durante el desarrollo de este trabajo (2013-2014) se lanzaron al mercado varias de estas que podrían remplazar a la unidad central elegida en un inicio.

En conclusión se puede decir cada objetivo planteado en un inicio se cumplió de forma exitosa. Sin embargo como en todo desarrollo tecnológico es posible mejorar varios aspectos del mismo y con la idea de que pudiese convertirse en un producto comercial competitivo, se continuará con su desarrollo en el futuro.

7.3 Trabajo a futuro.

7.3.1 Mejoras.

- **Compatibilidad completa con todos los dispositivos que lo permitan.** Se deberá continuar con el desarrollo de la interfaz de usuario en la búsqueda de que la mayoría de los dispositivos existentes en el mercado sean capaces de ejecutarla en su totalidad, lo que implica una labor continua a la par de los lanzamientos de nuevos equipos.
- **Alcance de los sensores de movimiento.** En concreto, y de acuerdo a lo observado durante las pruebas, los sensores de movimiento X10 utilizados poseen un rango de transmisión limitado que además se ve afectado por los objetos interpuestos entre el sensor y el transceptor. Este inconveniente ya ha sido documentado por algunos usuarios de X10 llegando a la conclusión de que se debe a un diseño pobre de la antena receptora del transceptor. Existen varias posibles soluciones para esto, la primera de ellas implica la modificación del transceptor y su antena para aumentar el rango de recepción. Lo otra sugiere utilizar repetidores de señales X10 RF y así extender el rango de los sensores. La última significa utilizar otro tipo de tecnología para este propósito, aprovechando la capacidad, antes mencionada de la unidad central para adaptarse a diferentes tecnologías.

7.3.2 Cambios.

- **Reemplazar a Skype™ como medio de realización de las videollamadas y el envío de mensajes de texto.** Es preferible el desarrollo y uso de tecnologías propias, pues se tiene un control total sobre lo que se puede hacer con él, además de que puede ser desarrollado de tal forma que se ajuste totalmente a las necesidades del proyecto, impidiendo características faltantes o sobrantes, aumentando la eficiencia y eficacia del sistema; además, se evitan restricciones de uso que pudieran existir en software propietario, consideración muy importante si se piensa comercializar el sistema en un futuro.

- **Cambiar equipo de unidad central.** Este aspecto está estrechamente relacionado con el punto anterior, pues una de las razones de peso para la elección de la unidad central fue la capacidad de ejecutar el cliente de Skype™. Al dejar de utilizar este software, se elimina la necesidad de una unidad central con conjunto de instrucciones x86, pudiendo entonces implementarla en computadoras de placa simple de arquitectura ARM de menor costo.

7.3.3 Adiciones.

- **Agregar a interfaz gráfica métodos del monitoreo del sistema.** Esta adición permitiría monitorear de forma remota y sencilla las condiciones actuales del sistema, lo cual facilitaría la detección, diagnóstico y corrección de posibles fallos.
- **Fuente auxiliar de energía y detector de corte de energía.** Algunas unidades del sistema están estrechamente relacionadas con la seguridad de los usuarios por lo que un corte inesperado de energía no debería inhabilitarlas. La adición consistiría en el uso de una fuente auxiliar a la cual estarían conectados la unidad central, el transceptor y el enrutador, lo que permitiría el funcionamiento de sensores y cámaras de seguridad. También consiste de un método de detección de cortes de energía, para con esto notificar al usuario y así este pueda tomar medidas pertinentes.
- **Opción de modificar las programaciones hechas sin tener que eliminarlas.** Actualmente la aplicación web es capaz de permitir la programación de tareas y la eliminación de estas, sin embargo una adición útil consiste en permitir la modificación de los parámetros de las tareas programadas con anterioridad, sin esto implique eliminarlas y crearlas de nuevo.
- **Cambio dinámico de tamaño de pantalla en PCs.** Algunos elementos de la interfaz se muestran de manera relativa en la pantalla de acuerdo a la resolución actual de la ventana del navegador. Este cálculo se realiza al momento de cargar la página, ofreciendo un despliegue dinámico de los elementos que conforman la interfaz; sin embargo, si en un equipo fijo se carga la página mientras la ventana tiene cierto tamaño y luego este es

cambiado, los elementos permanecen fijos de acuerdo al primer cálculo; por esto es necesario trabajar en un método que le permita a la página recalcular las nuevas posiciones sin afectar la interacción que se tiene con los dispositivos móviles.

- **Integrar diversas tecnologías domóticas.** Para aprovechar al máximo las capacidades de adaptabilidad y escalabilidad del sistema es necesario que se estudien otras tecnologías de domótica existentes en el mercado adicionalmente a las utilizadas en este trabajo, lo que supone desarrollo adicional en software para integrar el nuevo hardware. Este punto involucra además del uso de tecnologías existentes, el diseño y desarrollo de tecnologías propias lo que a su vez llevaría a reemplazar poco a poco software y hardware de terceros, pudiendo así introducir ideas innovadoras y vanguardistas.

Referencias por capítulo

Capítulo 1:

- [1] R. A. Española, «Real Academia Española,» 2001. [En línea]. Available: <http://lema.rae.es/drae/?val=dom%C3%B3tica>. [Último acceso: 1 Julio 2013].
- [2] Mohd. Rihan y M. Salim Beg, «Evolution of Home Automation Technology,» BVICAM's International Journal of Information Technology, vol. I, n° 2, 2009.
- [3] B. AJ Brush, B. Lee, R. Mahajan, S. Agarwal, S. Saroiu y C. Dixon, «Home Automation in the Wild: Challenges and Opportunities,» Microsoft research, 2011.
- [4] P. Ejnar Rovsing, P. Gorm Larsen, T. Skjødeberg Toftegaard y D. Lux, «A Reality Check on Home,» Journal of Green Engineering, pp. 303-327, 2011.

Capítulo 2:

- [5] V. Jonathan, «Embedded Systems: Real-Time Operating Systems for the ARM® Cortex™-M3 Volume 3,» 2012. [En línea]. Available: <http://users.ece.utexas.edu/~valvano/>. [Último acceso: 2013 07 24].
- [6] S. Heath, Embedded Systems Design, Newnes, 2002.
- [7] GizmoSphere, «gizmosphere.org,» 26 07 2013. [En línea]. Available: <http://www.gizmospher.org/wp-content/uploads/2013/07/Gizmo-Explorer-Kit-User-Guide-Rev-2.6.pdf>. [Último acceso: 16 09 2013].
- [8] Inte's Open Source Technology Center; Circuitco, «minnowboard.org,» 27 08 2013. [En línea]. Available: http://files.minnowboard.org/pdfs/minnowBoard_Flyer_2013-08-27.pdf. [Último acceso: 16 09 2013].
- [9] Shenzhen Norco Intelligent Technology, «User Guide Norco BIS-6630 Development Kit,» Marzo 2012. [En línea]. Available: <https://edc.intel.com/Download.aspx?id=6593&returnurl=/Embedded/DesingCenter/Tools/Seed-Board-Program/Default.aspx>. [Último acceso: 31 07 2013].
- [10] S. y. A. A. A. Haldar, Operating Systems, 2010.

Capítulo 3:

- [11] EuroX10, «EuroX10: Transmission Theory,» 2013. [En línea]. Available: <http://www.eurox10.com/Content/X10SignalTheory.htm>. [Último acceso: 29 09 2013].
- [12] Microchip Technology Inc., «X10 Home Automation Using the PIC16F877A,» 2002. [En línea]. Available: <http://ww1.microchip.com/downloads/en/appnotes/00236a.pdf>. [Último acceso: 29 09 2013].

Capítulo 4:

- [13] E. P. Cheung, «Data format for X-10 wireless units,» 04 1997. [En línea]. Available: <http://www.linuxha.com/athome/common/x10.rf.txt>. [Último acceso: 28 11 2013].
- [14] D. Houston, «X-10 RF Protocol,» 2006. [En línea]. Available: <http://davehouston.org/fr.htm>. [Último acceso: 28 11 2013].
- [15] R. Pallas Areny, Sensores y Acondicionadore de Señal, Barcelona: Marcombo, 2003.

Capítulo 5:

- [16] R. L. Boylestad y L. Nashelsky, Electrónica: Teoría de Circuitos y Dispositivos Electrónicos, México: Pearson Educación, 2009.

Capítulo 6:

- [17] Oracle, «Why MySQL?,» 2013. [En línea]. Available: <http://www.mysql.com/why-mysql/>. [Último acceso: 12 2013].
- [18] W3Schools, «W3Schools Online Web Tutorials,» 2013. [En línea]. Available: <http://www.w3schools.com/>. [Último acceso: 12 2013].
- [19] The JQuery Foundation, «About | JQuery Mobile,» The JQuery Foundation, 2014. [En línea]. Available: <http://jquerymobile.com/about/>. [Último acceso: 2 2014].

Anexo 1: Glosario de términos.

API: Del inglés *Application Programming Interface*, interfaz de programación de aplicaciones. Conjunto de funciones y procedimientos que permiten agregar una capa de abstracción al proceso de programación, comunicando componentes de software de forma transparente.

Arquitectura cliente-servidor: Modelo de software que describe comunicación entre equipos en una red de datos, donde uno de ellos atiende peticiones provenientes de los otros equipos proporcionándoles servicios o recursos.

ASP: Tecnología perteneciente a Microsoft que permite la creación de contenido web dinámico ejecutado, al igual que PHP, del lado del servidor.

Capa de abstracción: Concepto referente al ocultamiento de ciertas funcionalidades, aplicable a todos los niveles de un sistema computacional, con el fin de simplificar su entendimiento.

Conjunto de instrucciones x86: Conjunto de instrucciones interpretables por una CPU derivadas de aquellas diseñadas en un principio para el procesador Intel 8086.

CPU: Del inglés *Central Processing Unit*, unidad de procesamiento central. Dispositivo principal de una computadora encargado de llevar a cabo el procesamiento de todos los datos.

Cron: Servicio disponible en sistemas UNIX y derivados, encargado de ejecutar tareas programadas de acuerdo a una tabla de programaciones definida por el usuario.

Ethernet: Estándar de redes de datos de área local que implementa las capas física y de enlace de datos del modelo OSI.

Firewall: Cortafuegos en español. Dispositivo de hardware/software que impide conexiones no autorizadas a una red de datos privada provenientes de una pública, principalmente internet.

FTP: Del inglés *File Transfer Protocol*. Protocolo de red diseñado para la transferencia de archivos en una **arquitectura cliente-servidor**.

Gigabit Ethernet: Aplicación del estándar de red Ethernet que alcanza la velocidad de transmisión de datos de 1[Gb/s].

GNU GPL: Del inglés GNU *General Public License*. Licencia que declara como software libre a aquel que la porta, garantizando la libertad de usar, estudiar, compartir y modificar dicho software.

GPU: Procesador auxiliar encargado del procesamiento de gráficos y operaciones vectoriales y de coma flotante.

HTTP: Del inglés *HyperText Transfer Protocol*, protocolo de transferencia de hipertexto. Protocolo utilizado en internet para la transferencia de contenido web texto, multimedia e hipertexto, desde el servidor web hacia el navegador del cliente.

Kernel: Componente central de un sistema operativo encargado de las tareas de gestión de acceso al hardware y la utilización de recursos.

Memoria Caché: Memoria de alta velocidad y baja capacidad de almacenamiento utilizada por el procesador para acceder de forma rápida a datos usados con frecuencia.

Memoria Flash: Memoria de acceso aleatorio no volátil, utilizada comúnmente como unidad de almacenamiento secundario.

Microcontrolador: Circuito integrado que contiene todos los componentes básicos que conforman una computadora.

Modelo OSI: Modelo de referencia para la implementación de redes de datos, que divide las operaciones de red en 7 capas que van desde la conexión física hasta las aplicaciones con las que interactúa el usuario.

Nombre de dominio: Herramienta que permite asociar una dirección IP a una cadena de caracteres más fáciles de recordar por la mayoría de los usuarios.

ONE-NET: Estándar de código abierto para redes inalámbricas, diseñado para implementación de redes de datos de bajo consumo energético y costo, ideal para aplicaciones en domótica.

Optoacoplador: Dispositivo que se vale del efecto fotoeléctrico para acoplar dos etapas de un circuito proporcionando la protección que esto implica.

POE: *Power Over Ethernet* Tecnología que permite la polarización de un dispositivo de red mediante el cableado propio de esta.

Proceso demonio. Proceso computacional de un sistema operativo que se distingue por su nula interacción con el usuario y por mantenerse en ejecución en segundo plano todo el tiempo.

Puerto TCP: Número que identifica a una aplicación que utiliza el **protocolo TCP**.

PWM: Del inglés *Pulse-Width Modulation*, modulación por ancho de pulso. Técnica de modulación de señales en la cual se modifica el ciclo de trabajo de una señal para controlar su potencia.

Script: Archivo que contiene una serie de comandos simples, generalmente interpretados, para ejecutarse de forma secuencial.

Shell Script: Archivo *Script* ejecutado por el intérprete de comandos *Shell* de los sistemas UNIX.

SMTP: Del inglés *Simple Mail Transfer Protocol*, protocolo de transferencia simple de correo electrónico. Protocolo diseñado para realizar envío de correos electrónicos para su posterior recuperación en un servidor.

Socket TCP: Método de interconexión mediante el cual dos aplicaciones, que pueden residir o no en el mismo equipo, realizan el intercambio de datos utilizando el protocolo TCP.

Software de código abierto: Tipo de software en el que sus creadores decidieron proporcionar el código fuente de manera libre para que pueda ser modificado por los usuarios.

Software libre: Tipo de software cuya licencia permite el uso, copia, estudio y la redistribución de este.

Syfs: Sistema de archivos virtual proporcionado por el núcleo Linux que permite el acceso a dispositivos de hardware desde el espacio de usuario a través de una serie de archivos.

TCP: Protocolo de red para conexiones entre aplicaciones perteneciente a la capa cuatro del modelo OSI, que asegura que los datos serán enviados sin errores ni pérdidas y de forma segura.

Trama Ethernet: Unidad mínima de transmisión de datos, con formato establecido por Ethernet, perteneciente a la capa 2 del modelo OSI.

Transceptor: Dispositivo bidireccional que cumple con las funciones de transmisor y receptor de datos, pudiendo servir además como interfaz entre dos medios de comunicación diferentes.

Transductor. Dispositivo que transforma la energía de un fenómeno físico en otro tipo de energía, generalmente de menor magnitud.

Usuario www-data. Usuario utilizado para ejecutar los procesos del servidor web Apache, por lo que carece de prácticamente todos los permisos de un usuario típico en un sistema operativo tipo UNIX.

UTP: Tipo de cable trenzado en pares de ocho terminales no apantallado, muy utilizado en la implementación de redes de datos.

Voz sobre IP: Servicio de red que hace posible la transmisión de voz a través de la red utilizando el protocolo de internet (IP).

XML: Lenguaje para la elaboración de documentos que almacenen información de fácil extracción.

Zigbee: Especificación de red para comunicaciones inalámbricas de bajo consumo de energía, definida por el estándar IEEE 802.15.4, altamente vinculado al campo de la domótica.

Z-wave: Es un protocolo de comunicaciones inalámbricas diseñado específicamente para tareas de control y automatización residencial.