DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA ELECTRÓNICA EN AUTOMATIZACIÓN Y CONTROL

TRABAJO DE TITULACIÓN, PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERÍA ELECTRÓNICA EN AUTOMATIZACIÓN Y CONTROL

TEMA: DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL INTEGRADO EN EL HOGAR BASADO EN EL CONCEPTO DEL INTERNET DE LAS COSAS INDUSTRIALES PARA UN DEPARTAMENTO TIPO SUITE

AUTOR: PAROLI MENDOZA MICHELE VALENTINO

DIRECTOR: PONCE MORQUECHO CARLOS ALBERTO

SANGOLQUÍ

2017

ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICO

INGENIERÍA ELECTRÓNICA EN AUTOMATIZACIÓN Y CONTROL

CERTIFICACIÓN

Certifico que el trabajo de titulación, **"DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL INTEGRADO EN EL HOGAR BASADO EN EL CONCEPTO DEL INTERNET DE LAS COSAS INDUSTRIALES PARA UN DEPARTAMENTO TIPO SUITE"** realizado por el señor **MICHELE VALENTINO PAROLI MENDOZA**, ha sido revisado en su totalidad y analizado por el software anti-plagio, el mismo cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, por lo tanto me permito acreditarlo y autorizar al señor **MICHELE VALENTINO PAROLI MENDOZA** para que lo sustente públicamente.

Sangolquí, 03 de marzo del 2017

CARLOS ALBERTO PONCE MORQUECHO
DIRECTOR

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICO

INGENIERÍA ELECTRÓNICA EN AUTOMATIZACIÓN Y CONTROL

AUTORÍA DE RESPONSABILIDAD

Yo, **MICHELE VALENTINO PAROLI MENDOZA**, con cédula de identidad N° 0801430620, declaro que este trabajo de titulación **"DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL INTEGRADO EN EL HOGAR BASADO EN EL CONCEPTO DEL INTERNET DE LAS COSAS INDUSTRIALES PARA UN DEPARTAMENTO TIPO SUITE"** ha sido desarrollado considerando los métodos de investigación existentes, así como también se ha respetado los derechos intelectuales de terceros considerándose en las citas bibliográficas.

Consecuentemente declaro que este trabajo es de mi autoría, en virtud de ello me declaro responsable del contenido, veracidad y alcance de la investigación mencionada.

Sangolquí, 03 de marzo del 2017

MICHELE VALENTINO PAROLI MENDOZA

C.C. 0801430620

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICO

INGENIERÍA ELECTRÓNICA EN AUTOMATIZACIÓN Y CONTROL

AUTORIZACIÓN

Yo, **MICHELE VALENTINO PAROLI MENDOZA**, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar en la biblioteca Virtual de la institución el presente trabajo de titulación **"DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL INTEGRADO EN EL HOGAR BASADO EN EL CONCEPTO DEL INTERNET DE LAS COSAS INDUSTRIALES PARA UN DEPARTAMENTO TIPO SUITE"** cuyo contenido, ideas y criterios son de mi autoría y responsabilidad.

Sangolquí, 03 de marzo del 2017

MICHELE VALENTINO PAROLI MENDOZA

C.C. 0801430620

DEDICATORIA

A mi hijo, la razón que me motiva a superarme con cada reto, afrontar cada obstáculo que se presenta y querer ser cada día mejor.

A mi esposa, la cual amo incondicionalmente, es mi soporte y apoyo en cada reto que emprendo y de la cual nunca me podre separar.

A mis padres, que me dieron la vida, me educaron y me convirtieron en la persona que soy, sin ellos habría sido imposible alcanzar los logros que he obtenido hasta el día de hoy.

AGRADECIMIENTO

En primer lugar, a Dios, por haberme dado la fortaleza de seguir cada paso dentro de este largo, pero satisfactorio camino que represento mi carrera. Por ser esa motivación que me ha impulsado a continuar y no dejarme desfallecer.

A mi familia, por estar siempre ahí, ofreciéndome su apoyo incondicional, por entenderme y alentarme en cada decisión, obstáculo y oportunidad.

A mi "Alma Mater", la Universidad de las Fuerzas Armadas ESPE, por haber inculcado en mi un gran sentido y sentimiento de profesionalismo y emprendimiento, como también haberme enriquecido con conocimiento e intelecto de calidad.

ÍNDICE

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

RESUMEN

El presente trabajo abarca el diseño y la implementación de un sistema de automatización para el hogar basado en el concepto del Internet de las Cosas Industriales o IIoT. Innovador concepto que se desprende del Internet de las Cosas y pretende la conexión de objetos, que son parte de un proceso de control y automatización, con Internet a través de una red de control basada en TCP/IP con arquitectura distribuida y comunicación punto-a-punto. La tecnología empleada para implementar el proyecto y cumplir con el concepto de IIoT es IzoT, que es una plataforma abierta para la programación y desarrollo de redes de control distribuidas sobre IP. IzoT trabaja sobre plataformas SBC y sistema operativo Linux, y sus servicios de control son desarrollados en Python. Dentro del proyecto, los SBC son la base de controladores especialmente construidos y que incluyen tarjetas de entradas y salidas que se conectan a través de un bus I2C. Los controladores han sido instalados en diferentes puntos de la vivienda e interconectados a través de una red de datos Ethernet, funcionan como puntos de integración y control de diferentes periféricos, implementan automatismos y funcionalidades para el hogar, se conectan con Servicios de Internet, y prestan el servicio de una API REST para la conexión de una aplicación de usuario.

PALABRAS CLAVES

- **INTERNET DE LAS COSAS INDUSTRIAL**
- **AUTOMATIZACIÓN EN EL HOGAR**
- **TECNOLOGÍA IZOT**
- **SERVICIOS DE INTERNET**
- **ODROID C2**

ABSTRACT

This work includes the design and implementation of a home automation system based on the Industrial Internet of Things concept (IIoT). Innovative concept that emerges from the Internet of Things and pretends the connection of objects, which are part of a control and automation process, with Internet through a control network based on TCP/IP with distributed architecture and point-to-point communication. The technology used to implement the project and comply with the IIoT concept is IzoT, which is an open platform for the programming and development of distributed control networks over IP. IzoT works on SBC platforms and Linux operating system, and uses Python for the development of its control services. Within the project, SBCs are the basis of specially constructed controllers that include input and output boards. An I2C bus connects the SBC and boards. The controllers are located at different points of the house and these interconnected through an Ethernet data network, they function as points of integration and control of different peripherals, implement automatisms and functionalities for the home, connect with Internet Services, and provide the API REST service for the connection of a user application.

KEYWORDS

- **INDUSTRIAL INTERNET OF THINGS**
- **HOME AUTOMATION**
- **IZOT TECHNOLOGY**
- **INTERNET SERVICES**
- **ODROID C2**

PRÓLOGO

El presente documento recoge el diseño e implementación de un sistema de control integrado para el hogar, basado en el concepto del Internet de las Cosas Industriales. Un innovador concepto que señala la automatización de procesos a través de dispositivos distribuidos y autónomos que se comunican a través de una red de datos TCP/IP, y que permite conectar los diferente elementos u objetos del proceso a internet y sus servicios de una forma rápida, flexible y escalable.

El Internet de las Cosas o IoT es un moderno concepto que pretende conectar los objetos que utilizamos a diario con la red mundial, con el objetivo de tener mayor control del mundo que nos rodea y aprovechar de una forma más eficiente y sostenible los recursos con los que cuenta nuestro entorno. El IIoT o Internet de las Cosas Industriales es una particularidad o aplicativo del IoT, que tiene como objetivo la comunicación de dispositivos que requieren intercambiar algún tipo de información entre ellos, con el fin de realizar un trabajo. En otras palabras, una comunicación punto-a-punto.

En el presente trabajo se orientó el concepto de las IIoT a la automatización e integración de varios subsistemas de control y seguridad de una vivienda tipo suite, como son la iluminación, accesos, consumo de energía, supervisión de incendios y vigilancia contra-intrusión. Esto con el objetivo de mejorar la calidad de vida, integridad y bienestar de los ocupantes, como también mejorar la eficiencia energética de la vivienda.

Para poder aplicar el concepto de las IIoT se utilizó la tecnología abierta IzoT para construcción de redes de control distribuidas por medios de comunicación basados en el modelo TCP/IP. IzoT permite de una forma modular, virtualizar el estado de periféricos como luminarias, puertas, ventanas, circuitos y sensores medidores de entorno, y conectarlos con controles, automatismos y funcionalidades de una forma

robusta y flexible. El protocolo de IzoT trabaja sobre sistemas computacionales de alto nivel y puede ser ejecutado sobre sistemas operativos basados en Linux. Esto lo hace especialmente ideal, ya que puede ser posible implementar proyectos con computadores de placa reducida o SBC, utilizados para el presente proyecto como núcleos de los controladores que conforman la columna vertebral del sistema de control integrado.

Gracias a la libertad de desarrollo que permite IzoT y los SBCs, parte fundamental del proyecto es la integración con servicios prestados en internet como correo electrónico, envió de mensajes de texto SMS y almacenamiento de datos en la nube. Esto permite que la vivienda pueda realizar notificaciones de incidencias de forma remota, como registrar datos en la nube que pueden ser supervisados y analizados con el fin de mejorar parámetros como el confort de una estancia o la demanda de energía, lo que podríamos llamar como una Aplicación IoT, cada vez más popular hoy en día.

Otro importante aspecto de este proyecto, es la utilización de tecnologías de internet como REST, o también llamado RESTful, que permite el consumo de servicios a través del protocolo HTTP. Gracias a que IzoT permite implementar un API REST para los dispositivos y controles contenidos en la red punto-a-puntos, posibilito el desarrollo de una aplicación cliente o interfaz de usuario que es posible desplegar desde un navegador web y gestionar la vivienda desde cualquier parte del mundo.

CAPÍTULO I

FUNDAMENTOS TEÓRICOS

1. FUNDAMENTOS TEÓRICOS

1.1. EL INTERNET DE LAS COSAS

Se calcula que para el 2020, aproximadamente 50.000 millones de dispositivos se conectarán a Internet con el fin de proporcionar a las personas una serie de servicios y aplicaciones inteligentes sin precedentes. Refrigeradoras que notifiquen la fecha de caducidad de los alimentos, plantas que dan a conocer en tiempo real sus necesidades, electrodomésticos que se pueden encender antes de llegar a casa, inodoros que realizan análisis de orina para recomendar la dieta más adecuada, lámparas que se encienden solas ante la ausencia de luz, zapatos que registren en la nube las estadísticas de cuánto corre una persona cada semana, marketing a través de Beacon o animales que pueden transmitir su ubicación son algunas posibles aplicaciones de lo que se conoce como el Internet de las Cosas o IoT por sus siglas en inglés, *Internet of Thing*, un concepto un poco abstracto que nació en el año de 1999 en el Auto-ID Center del Instituto de Tecnología de Massachusetts (Elena Sanz, s.f.).

Podríamos definir al Internet de las Cosas como “una red que interconecta objetos físicos valiéndose de las tecnologías de internet”. Dichos objetos están provistos, enlazados o acoplados de algún elemento de cómputo que le permite no solo la conectividad a Internet, sino que además programar eventos específicos en función de las tareas que le sean dictadas remotamente.

Cualquier tipo de objeto puede ser conectado a Internet con la tecnología adecuada. Diferentes medios pueden ser utilizados para llevar a la nube las condiciones y tareas de un objeto. Medios como transductores, sensores, motores, conmutadores e incluso tecnologías de identificación como códigos de barras, códigos QR o etiquetas RFID pueden conectar objetos con elementos de cómputo para luego ser enlazados con la red global, ya sea por medios de comunicación y protocolos nativos de internet o métodos indirectos a través de buses de campo, medios inalámbricos y pasarelas de comunicación. Aunque este último no sea precisamente lo más ideal.

1.1.1. Aplicaciones para el Internet de las Cosas

El Internet de las Cosas representa un sinfín de aplicaciones que se pueden desarrollar tanto en el sector industrial como doméstico, aplicaciones que conllevan grandes beneficios como reducir y controlar la sobre-producción, reducir la contaminación ambiental o hacer más eficiente el consumo de energía.

Según **Echelon Corporation** (2013) en su artículo *"Requirements for the Industrial Internet of Things",* las aplicaciones del Internet de las Cosas las podemos clasificar en tres grandes grupos:

La primera clase de aplicaciones son las del Internet de las Cosas orientadas al Consumidor. Estas aplicaciones se caracterizan por la interacción constante de un humano con un dispositivo. En el IoT del Consumidor, las comunicaciones se ejecutan dentro de una arquitectura cliente-servidor y a menudo están fluyendo grandes cantidades de datos.

La siguiente clase de aplicaciones son las del Internet de las Cosas orientadas a la adquisición y análisis de Datos. Un sistema de rastreo de vehículos, un sistema que monitorea las estructuras de un edificio en busca de signos de

desgaste, o una red de sensores climatológicos distribuidos por toda una ciudad son ejemplos de esta clase de aplicaciones. Las aplicaciones del IoT orientados a los Datos son llamadas a menudo como máquina-a-máquina o M2M, porque los dispositivos destinados al monitoreo se conectan con servicios para almacenar sus datos y posterior ser analizados por un operador.

Existe una tercera clase de aplicaciones mucho más exigente que las dos anteriores, son las aplicaciones del Internet de las Cosas orientadas a los Procesos o simplemente Internet de las Cosas Industriales o IIoT por sus siglas en inglés, *Industrial Internet of Things*. Estas clases de aplicaciones utilizan un control punto-a-punto con inteligencia distribuida y estas conformadas por redes de nodos autónomos que trabajan juntos para lograr tareas específicas.

Para estos sistemas, los requisitos de comunicaciones no son cliente-servidor. En su lugar, los nodos actúan como pares en la red, tomando sus propias decisiones e informando su estado al resto de nodos que lo requieran. Las transferencias de datos son frecuentes, pero generalmente no transmiten grandes cantidades de datos.

Además de realizar su tarea principal, estos sistemas también se conectan a un Sistema Empresarial como un SCADA[1], BMS[2], Base de Datos o Servicio de Internet, para emitir alarmas, almacenar información relevante o generar reportes de históricos y estadísticos. Esta conexión puede ser a través de una red de datos local o a través de Internet con el Sistema Empresarial alojado en la nube. Cuando se comunica con el Sistema Empresarial la arquitectura de comunicaciones, al igual que en la clase de aplicaciones orientadas a los Datos, se transforma en un modelo M2M o cliente-servidor.

[1] Siglas de *Supervisory Control And Data Acquisition.*
[2] Siglas de *Building Management System.*

1.1.2. El Internet de las Cosas Industrial

El IoT, como ya se ha expuesto con anterioridad, presenta generalmente una arquitectura de comunicaciones cliente-servidor y gestión en la nube. Esto para adaptar la conexión de los objetos a las redes actuales, tal cual como se ha venido trabajando con los dispositivos orientados a los datos e información como computadores personales, teléfonos inteligentes o tabletas.

El problema con arquitecturas cliente-servidor y gestión de la información en la nube es que las aplicaciones se transforman en Sistemas de Control Centralizados (todos los puntos de control dependen de un punto maestro), lo que genera un peligro constante a una caída absoluta del sistema ante una falla de funcionamiento en el servidor central o la red de comunicaciones. Es por esto que dentro del Internet de las Cosas nace el concepto del Internet de las Cosas Industriales o IIoT con el fin de manejar objetos con capacidad de conectarse a Internet y que al mismo tiempo forman parte de un sistema de control y automatización.

El concepto del Internet de las Cosas Industriales propone la aplicación de tecnologías que permitan a los puntos de control de un sistema interoperar entre si bajo una arquitectura de control y comunicación punto-a-punto o P2P, generando un sistema de control distribuido y dando una solución eficiente a los problemas antes expuestos.

Del artículo *“Connecting with the Industrial Internet of Things”* de **Moor Insights & Strategy** (2013) se puede extraer que el IIoT logra que los objetos que conforman el sistema de control puedan ser autónomos (no dependientes de un controlador central), una frecuente transmisión de datos en tiempo real, acceso a cada uno de ellos de forma independiente y conexión directa a Internet sin la necesidad de pasar por un equipo Gateway o Pasarela. La conexión directa a

Internet permite que los puntos de control interactúen con Sistemas Empresariales ubicados o alojados en cualquier parte del mundo dentro de un servidor o data center haciendo mucho más eficiente la gestión del sistema y ahorrando en infraestructura.

1.1.3. Requisitos del Internet de las Cosas Industrial

El modelo de protocolos TCP/IP tiene grandes ventajas, además de ser popular, también permite el enrutamiento de grandes volúmenes de datos, es fiable, escalable y abierto, e independiente del medio físico, además de tener un sinnúmero de dispositivos de infraestructura y administración en el mercado por parte de miles de fabricantes en todo el mundo. Pero también conlleva desventajas como ser propenso a eventuales saturaciones del medio físico, presentándose colisiones y perdidas de paquetes que podrían afectar al “tiempo real” de un sistema industrial o visto de una forma más simple, a la llegada oportuna de la información que intenta transmitir un nodo a otro.

Esto determina que la tecnología, plataforma y protocolo de comunicaciones que se utilice para la implementación de un sistema basado en el concepto del Internet de las Cosas Industriales debe cumplir un conjunto de requisitos que permitan tener una transmisión de datos confiable y oportuna en las peores circunstancias posible. Dichos requisitos se pueden clasificar en cuatro áreas principales (Echelon Corporation, 2013):

- Resiliencia en caso de fallas.
- Requisitos de conectividad física.
- Seguridad.
- Servicios de control.

1.2. LA TECNOLOGÍA IZOT

IzoT es una plataforma de control y comunicaciones multi-protocolo y multi-medio que trabaja sobre el modelo de protocolos TCP/IP[3]. El termino IzoT nació como una simplificación en la pronunciación del acrónimo I^2oT, que a su vez es una simplificación del acrónimo IIoT (*Industrial Internet of Thing*).

A diferencia de las plataformas de nivel del consumidor, la plataforma IzoT trata de forma integral los requerimientos específicos del IIoT como son, control distribuido, la fiabilidad de los sistemas industriales, soporte para el medio, escalabilidad, adaptabilidad, coexistencia y evolución.

IzoT fue desarrollado por la compañía norteamericana **Echelon**, la misma que hace más de 20 años desarrollo el protocolo de control punto-a-punto LonTalk, actualmente estandarizado bajo las normas ISO/IEC 14908 y ANSI 709.1, y extensamente utilizado en el sector industrial y de automatización de edificios. IzoT es en realidad una fusión de los conceptos de la norma IEC 14908 con el modelo TCP/IP para formar redes de control que convivan dentro de la misma infraestructura de las redes de datos sin perder su autonomía y horizontalidad (las redes de datos presentan arquitecturas verticales cliente-servidor con un fuerte punto de fallo). Los dispositivos provistos para interoperar en redes LonTalk clásico (IEC 14908-1) son completamente compatibles con la tecnología IzoT.

La base de la tecnología IzoT es el protocolo LonTalk/IP, el cual conecta a las comunidades de dispositivos (también llamados nodos) como periféricos, sensores o actuadores. LonTalk/IP permite generar ambientes de procesos en tiempo real con gran precisión y exactitud.

[3] Conjunto de protocolos de estándar defacto que son, hoy en día, la base del Internet.

1.2.1. Fundamentos del protocolo LonTalk/IP

La comunicación de datos en LonTalk/IP está estructurada por objetos llamados datapoints o punto de datos. Cada punto de dato puede representar una variable física como, por ejemplo, el estado de un interruptor o la velocidad de un motor. Los datapoints también pueden representar variables virtuales como alarmas, eventos o mediciones indirectas. Los datapoints pueden ser de entrada o salida, y están organizados dentro de bloques funcionales que son la representación virtual del dispositivo o periférico en sí.

Los puntos de datos y bloques funcionales están definidos por un amplio conjunto de Tipos de Datos y Perfiles Funcionales, muchos de ellos estandarizados dentro de la norma IEC 14908. La dinámica de LonTalk/IP consiste en la conexión virtual de los datapoints de salida con los de entrada. De esta manera los dispositivos interaccionan entre si y logran cumplir funcionalidades y automatismos con gran flexibilidad. Las conexiones virtuales se logran gracias a la utilización de herramientas de gestión y administración de redes LonTalk/IP. **Echelon** proporciona su propia herramienta llamada *OpenLNS Commissioning Tool*, la cual funciona como un plug-in de *Microsoft Visio 2010* desde el cual es posible crear redes, comisionar dispositivos a estas redes y conectar los datapoints de estos dispositivos (Echelon Corporation, 2016).

A continuación, se resume las características principales del Protocolo LonTalk/IP (Echelon Corporation, 2016):

- Entrega fiable de mensajes.
- Detección de mensajes duplicados.
- Bajo costo de dispositivo.
- Bajo costo de instalación y mantenimiento.
- El uso eficiente del ancho de banda del canal.

- Sistemas interoperables.
- Segmentación de sistemas.

1.2.2. Servicios de control de LonTalk/IP

LonTalk/IP es un protocolo que proporciona servicios de control sobre el protocolo estándar IP. Trabaja en las 7 capas del modelo OSI, donde las capas superiores dentro del modelo LonTalk/IP, son llamadas de Servicios de Control, y las inferiores de Servicio de Transporte. Las capas de Servicios de Control están definidas por las capas 4 a 7 del protocolo estándar para redes de control ISO/IEC 14908-1. Y las capas de Servicios de Transporte están definidas por el modelo de protocolos TCP/IP RFC-791/793. LonTalk/IP proporciona soporte para medios de comunicación definidos en el estándar IEEE 802 como Ethernet y WiFi (Echelon Corporation, 2016).

Los servicios de control son aquellos que permiten interoperar a los dispositivos dentro de una red. LonTalk/IP logra una comunicación punto-a-punto gracias a que cada dispositivo en la red puede ejercer acciones de publicación y suscripción a través de sus datapoints y conexiones. Cada dispositivo dentro de una red pude publicar o servir sus datos a otros dispositivos como también suscribirse o solicitar datos de otros dispositivos. De una forma más simple, cada dispositivo dentro de una red LonTalk/IP es cliente y servidor a la vez.

En una primera instancia, la suscripción y publicación de datos a través de conexiones virtuales determinan el servicio de control más importante. Junto a este se implementan otros servicios de control como la fiabilidad de las comunicaciones, la presentación de los datos y las herramientas y aplicaciones de gestión de la red y usuarios. Todos estos servicios se implementan en modelo de capas jerárquico determinados en las 4 capas superiores del estándar

ISO/IEC 14908. La Tabla 1 muestra una breve descripción de dichas capas y sus equivalentes con el modelo OSI.

Tabla 1

Descripción de capas superiores del protocolo ISO/IEC 14908

No.	Nombre de Capa OSI	Numero de Capa OSI	Capa ISO/IEC 14908	Descripción
1	Capa de Aplicación	7	Herramientas y Aplicaciones	Herramientas de gestión y análisis de red, Aplicaciones específicas como registro de datos, gestión de alarmas o programadores horarios.
2	Capa de Presentación	6	Interpretación de Datos	Datapoints, Bloques Funcionales
3	Capa de Sesión	5	Control	Suscripción/Publicación, Autentificación
4	Capa de Transporte	4	Fiabilidad en las Comunicaciones Extremo-a-Extremo	Entrega de mensajes con reconocimiento (Acknowledged) y sin reconocimiento (Unacknowledged), ordenamiento común, detección de mensajes duplicados.

Fuente: (Echelon Corporation, 2015).

En el **ANEXO A** de este documento se adjunta una guía completa de los elementos que determinan los servicios de control de LonTalk/IP, extraído del Manual de IzoT publicado por **Echelon Corporation**.

1.2.3. Desarrollo de dispositivos con IzoT

Para el desarrollo de dispositivos con IzoT, **Echelon** proporciona un SDK[4] de forma gratuita y de código abierto que es posible descargar desde la página

[4] Siglas de *Software Development Kit.*

http://www.echelon.com/izot-platform. El IzoT SDK puede ser instalado en dispositivos con arquitectura de procesadores ARM de 32 bits y 64 bits, con sistemas operativos basados en Linux como Debian o Ubuntu, y esta implementado sobre un Entorno Virtual de Python.

Un Entorno Virtual (comúnmente abreviado como “virtualenv”) es un directorio con una instalación de Python de una versión en particular, que permite ejecutar aplicaciones Python en segundo plano (PyAr, Python Argentina. Python Software Foundation, 2014).

El IzoT SDK incluye la pila LonTalk/IP copilada en una biblioteca de C/C++ y el IzoT Python Package desarrollado sobre el lenguaje de programación Python versión 3.4. El IzoT Python Package es un conjunto de módulos que permiten construir el Firmware del dispositivo. Incluye un Framework[5] para la biblioteca de la pila LonTalk/IP con el cual es posible controlar todas las funcionalidades y servicios del protocolo LonTalk/IP, y librerías de clases que definen los diferentes Tipos de Datos y Perfiles Funcionales estándares (Echelon Corporation, 2014).

Con el IzoT Python Package es posible implementar y desarrollar los datapoints, bloques funcionales y aplicaciones de un dispositivo o nodo de la red de forma sencilla y estructurada gracias a que Python es un lenguaje de programación interpretativo y de alto nivel (multiparadigma). Utilizando el IzoT Python Package, es como los dispositivos pueden ejecutar sus funcionalidades dentro de la red y conectarse con los puntos de datos de otros dispositivos tanto para publicar la información de sus periféricos asociados, como suscribirse a la información de los periféricos de los otros dispositivos. Se adjunta a este documento en su **ANEXO B** una descripción de los directorios del Entorno Virtual y de los comandos del SDK de IzoT (Echelon Corporation, 2016).

[5] Conjunto de módulos de programas que sirven como infraestructura para un software.

1.2.4. Servidor IzoT

El Servidor IzoT permite construir un servicio para aplicaciones Web y M2M[6] dentro de una red basada en tecnología IzoT. Con el fin de conectar una red IzoT con una interfaz de usuario o un sistema empresarial. El Servidor IzoT permite a los operadores o administradores de la red, acceder de forma local o remota para la supervisión y control de las funcionalidades de la comunidad de dispositivos y periféricos. El Servidor IzoT es instalado con el SDK de IzoT y se ejecuta desde el mismo Entorno Virtual de Python.

El Servidor IzoT principalmente proporciona de un API REST fácil de usar con una interfaz de baja sobrecarga, pero de gran alcance para la detección de dispositivos, como también monitoreo de datos y control de variables de entrada de dichos dispositivos. El API REST también proporciona de una gestión de roles y usuarios para el acceso a los recursos de la red de control por medio de una autentificación de usuario y contraseña.

REST es una tecnología que utiliza HTTP[7] como interfaz entre los sistemas para obtener datos o generar operaciones sobre esos datos, y puede manejar diversos tipos de formatos como XML[8] o JSON[9]. Siendo este último el más flexible y fácil de implementar y procesar. Tanto XML como JSON son reglas para la estructuración de datos que se transmiten en texto plano. Se adjunta en el **ANEXO C** de este documento una guía completa del API REST del Servidor IzoT. En este es posible apreciar los recursos que proporciona el API como también la utilización de los métodos para su conexión.

---

[6] Máquina-a-Máquina, concepto genérico para el intercambio de información entre dos máquinas de forma remota.

[7] Siglas de *Hypertext Transfer Protocol*, protocolo base para la transmisión de recursos para la web, ejemplos, una página html o un stream de video.

[8] Sigas de *eXtensible Markup Language.*

[9] Siglas de *JavaScript Object Notation.*

1.3. SERVICIOS DE INTERNET

Hoy en día, Internet es mucho más que WWW, la red posee una serie de servicios que, en mayor o menor medida, tienen que ver con las funciones de información, comunicación e interacción. Algunos de los servicios disponibles en Internet, aparte de la Web, son el acceso remoto a otros ordenadores (a través de telnet o siguiendo el modelo cliente-servidor), la transferencia de ficheros, el correo electrónico (e-mail), los boletines electrónicos y grupos de noticias, las redes sociales, los streaming de audio y video, los blogs de contenidos, la geolocalización en mapas, los foros de debate y las conversaciones en línea (chats).

Los servicios que hoy ofrece Internet no sólo se han multiplicado, sino que han evolucionado hacia nuevas y mejoradas funciones y han ganado en facilidad de uso y manejo. Todo esto es lo que hoy en día se denomina “Nube”. A este cambio han contribuido no sólo la velocidad de transferencia de los bits que permiten los transceptores y routers actuales, y la mayor eficiencia y capacidad de las líneas de telecomunicaciones con un gran ancho de banda, sino también, mejoras en el software y las aplicaciones (bases de datos integradas en la web, motores de búsqueda, agentes inteligentes, etc.) y en el hardware (mayor capacidad de almacenamiento y memoria, incremento exponencial de la velocidad de los procesadores, capacidad de tratar todo tipo de datos, no sólo los textuales, sino también los datos multimedia, etc.) (María Jesús Lamarca Lapuente, 2013).

Una de las grandes ventajas que hoy ofrecen los Servicios de Internet es la ventaja de controlar las funcionalidades de los mismos desde una API[10]. Dicha API ofrece la posibilidad de que otras aplicaciones o maquinas se conecten al servicio a través de Internet y consuman sus funcionalidades y recursos. Esto

[10] Siglas de *Application Programming Interface.*

representa un gran beneficio hoy en día ya que permite que nuevas aplicaciones no tenga que implementar funciones que ya existen en la red, ahorrando en tiempo y recursos. Gracias a la popularidad de REST, la mayoría de servicios de Internet que hoy se ofrecen implementan esta clase de API para su conexión.

Las API REST han logrado mucho más que la conexión de aplicaciones a servicios, logrando extender esto a la conexión de dispositivos como sensores o actuadores. Es aquí donde toma un papel fundamental el Internet de las Cosas, las API REST han logrado que las cosas u objetos (con algún elemento de computo e interfaz de comunicación) puedan consumir las funcionalidades que hoy en día se ofrecen en Internet. A continuación, se detallan tres servicios que se implementaran en el presente proyecto para el registro de datos y notificación de eventos, y que son válidos para que dispositivos puedan consumir desde Internet.

1.3.1. Servicio de correo electrónico con Gmail

El correo electrónico o e-mail (como se lo denomina en países angloparlantes) fue uno de los primeros Servicios de Internet y en la actualidad aún sigue siendo uno de los más usados para la comunicación de información tipo texto por la red (no limitándose a esta pudiendo transmitir otra clase de morfologías como audio, video o ficheros). El correo electrónico se ha transformado en el medio de comunicación más importante a nivel empresarial o institucional, incluso se lo considera como un elemento jurídico.

Los sistemas de correo electrónico se basan en un modelo de almacenamiento y reenvío, de modo que no es necesario que ambos extremos se encuentren conectados simultáneamente. Para ello se emplea un servidor de correo, que hace las funciones de intermediario, guardando temporalmente los mensajes

antes de enviarse a sus destinatarios. Su protocolo de comunicaciones base es SMTP[11].

Gmail es un servicio de correo electrónico gratuito, provisto por la compañía especializada en productos y servicios relacionados con Internet, **Google**. Entre las ventajas que posee Gmail se encuentran la gran velocidad de la transmisión de sus correos gracias a la enorme infraestructura que posee **Google**, la seguridad de llegada de sus correos gracias al reconocimiento de su dominio, el almacenamiento perpetuo en la nube de los correos entrantes y salientes, y su interfaz completamente web (no requiere de otros protocolos como POP3 o IMAP para la descarga o revisión de correo entrante). Todo esto lo hace idóneo para la transmisión de información crítica como alarmas notificadas por aplicaciones, dispositivos o sistemas empresariales.

1.3.2. Servicio de envío de SMS con Twilio

Twilio es un servicio que permite, entre diversas aplicaciones que hoy en día presenta, el envío masivo de mensajes SMS[12] desde un API REST.

El SMS es un tipo de mensajería utilizado en las redes de telefonía celular como un medio de comunicación orientado al intercambio de textos cortos entre personas o abonados. Hoy en día sigue siendo ampliamente utilizado y no se limita únicamente a la comunicación de persona-a-persona sino también de máquina-a-persona e incluso de máquina-a-máquina.

Twilio proporciona la capacidad de enviar SMSs desde Internet a través del protocolo HTTP a un bajo costo, ahorrando a sus clientes la adquisición y mantenimiento de equipos Gateways. **Twilio** implementa toda la conexión y

[11] Siglas de *Simple Mail Transfer Protocol.*
[12] Siglas de *Short Message Service.*

gestión de su servicio web con la red telefónica celular, proveyendo de una interfaz simple de usar para que otras máquinas se conecten. El API de **Twilio** ofrece toda la información relevante sobre el estado de cada SMS enviado permitiendo una mejor gestión. Es posible conocer si el SMS tuvo éxito o no y sus posibles causas y errores.

Twilio trabaja con la red celular de los Estados Unidos desde donde se envían los SMS a cualquier parte del mundo. Desde la web de **Twilio**, *http://www.twilio.com*, es posible contratar o comprar un numero celular y todos los gastos de mantenimiento de la línea y envió de mensajes pueden ser deducidos de una tarjeta de crédito. De momento el costo por cada SMS enviado a Ecuador es de nueve centavos de dólar y el costo de una línea es de un dólar al mes.

1.3.3. Servicio de registro de datos para IoT con ThingSpeak

ThingSpeak es una plataforma para el Internet de las Cosas que permite recopilar, almacenar, visualizar y actuar sobre los datos de sensores y actuadores con capacidad de conectarse a Internet y utilizar el protocolo HTTP. Esta provee de almacenamiento en la nube de datos numéricos que pueden ser enviados a través de su API REST. **ThingSpeak** actúa como una red social para las cosas en la que diversas personas pueden compartir la recolección de datos y mediciones de diferentes dispositivos localizados en cualquier parte del mundo.

De acuerdo a la *Sección 1.1.1. Aplicaciones para el Internet de las Cosas* de este documento, se habló de las diferentes aplicaciones para el Internet de las Cosas, donde se trató aplicaciones orientadas al registro y análisis de datos y sistemas M2M. **ThingSpeak** precisamente fue creado para satisfacer esta clase de aplicaciones, pero de una forma abierta y completamente disponible en Internet. La mayor ventaja de utilizar **ThingSpeak** es la de proporcionar una base

de datos completamente gratuita y completamente montada y funcional, ahorrando costos y tiempo de almacenamiento, instalación y administración de bases de datos.

ThingSpeak utiliza una interfaz completamente web para el manejo, visualización y análisis de datos. Es posible acceder desde *http://www.thingspeak.com* y solo basta con registrarse y crear un canal para que un dispositivo pueda comenzar a enviar sus datos. Dentro del modelado de **ThingSpeak** un canal es una entidad u objeto que puede ser asociado a un número indeterminado de mediciones. Cada cuenta o canal es provista de un numero ID único o “llave” para el direccionamiento de la información enviada.

Dentro de la web de **ThingSpeak** es posible visualizar los datos en tiempo real en forma de gráficos históricos según van llegando, como también hacer consultas entre fechas determinadas. **ThingSpeak** proporciona de otras herramientas para el análisis y cálculo de la información almacenada como es la conexión con *MATLAB*.

Un servicio de registro de datos para IoT proporciona de la capacidad de almacenar información relevante de un entorno como puede ser una vivienda para su posterior análisis. Datos como el consumo eléctrico, luminosidad, temperatura, humedad y alarmas técnicas y de seguridad pueden ser almacenados. El registro de información puede ayudar a mejorar el consumo de energía (Ej. Análisis de la curva de carga para encontrar picos de consumo o malos hábitos energéticos) o el confort de los ocupantes (Ej. Detectar periodos del día en el que incrementa la luz solar o temperatura puede ayudar a tomar medidas para mantener el calor dentro de la vivienda por la noche).

1.4. DESARROLLO DE APLICACIONES CON JAVAFX

JavaFX es una tecnología de software que, combinada con Java, permite crear y desplegar aplicaciones con un aspecto vanguardista y contenidos avanzados, audio y vídeo. JavaFX permite el desarrollo de “Aplicaciones Ricas para Internet” o RIAs por sus siglas en inglés. Los RIAs no son más que aplicaciones web con características y capacidades de aplicaciones de escritorio, incluyendo aplicaciones multimedia interactivas.

JavaFX apoya el patrón de arquitectura de software MVC (modelo, vista, controlador) y puede ser completamente programado en lenguaje Java, a pesar que implementa otro lenguaje de marcado para la construcción de las vistas conocido como FXML. Al igual que Java, JavaFX es completamente abierto, es multiplataforma y está orientado a objetos. Como también requiere de la Máquina Virtual de Java (JRE) para que sus aplicaciones puedan ser ejecutadas, y del Kit de Desarrollo de Java (JDK) para poder ser desarrolladas. Desde la versión 8 de Java, JavaFX viene como una parte integral de la distribución central de Java, y no requiere de descargas adicionales.

JavaFX fue lanzado por primera vez en el 2010 y ha sido repotenciado por **Oracle Corporation**, luego de que este adquiriera **Sun Microsystems**. En sus inicios, el objetivo de JavaFX era el de competir con otras tecnologías similares como *Flash* de **Adobe** y *Silverlight* de **Microsoft** (tecnologías de software prácticamente desaparecidas a día de hoy). Pero, hoy en día, se presenta como la principal opción para el desarrollo de aplicaciones gráficas para Java dejando prácticamente descontinuado a sus predecesores como Swing o AWT. Si bien el objetivo de JavaFX no es el de competir con tecnologías de presentación web como HTML5, es una opción bastante optima cuando se requiera que la aplicación sea ejecutada desde un navegador de Internet (John Martens, 2014).

1.5. LA AUTOMATIZACIÓN EN EL HOGAR

Dentro de los hogares, las personas realizan una serie de procesos que están íntimamente relacionados con la vida misma, por lo que en la automatización en el hogar es donde más se refleja el aumento de la calidad de vida que puede prestar esta clase de sistemas. La iluminación, climatización, accesos, seguridad, alimentación, higiene, estancia y entretenimiento son varios de los sistemas que se pueden encontrar en los hogares, y los cuales representan diversos procesos para su funcionamiento. Poder automatizar dichos sistemas representan un incremento sustancial en el confort, seguridad y ahorro de recursos de los ocupantes de una vivienda. Un ejemplo claro son las lavadoras y secadoras, las cuales pueden lavar y secar las prendas de vestir de forma automática y programada, el proceso manual de lavado de ropa representaba en la antigüedad un trabajo largo y tedioso, un desgaste en la salud y gastos excesivos de recursos hídricos.

1.5.1. Beneficios de la automatización en el hogar

Según Cesar Calderón Y Henry Sánchez (2016) en su publicación "*Domótica una mejor calidad de vida"*, la automatización en el hogar podría mejorar la calidad de vida y la calidad de tiempo. El uso de dispositivos que facilitan trabajos mejoran el consumo de recursos, reducen la contaminación ambiental y pueden tener un efecto directo en aspectos fundamentales de las personas como la salud, educación, cultura y felicidad.

A continuación, se describen las ventajas más importantes que se obtienen de un hogar automatizado:

A. Confort

El apagado de luces a una hora programada, el encendido automático del calefón cuando se abre una llave o la apertura de una puerta desde un portero eléctrico, reducen en las personas el grado de percepción del entorno, aumentado la condición de bienestar y comodidad.

B. Seguridad

Sistemas electrónicos de vigilancia conformados por sensores detectores de movimiento, humo, calor, gas, monóxido de carbono y cámaras de video ayudan a la prevención de incidencias catastróficas que amenazan la integridad de las personas y los bienes dentro de la vivienda como incendios, robos, fugas o envenenamientos.

C. Control

Mientras mayor control se tenga sobre un sistema mayor será su eficiencia, la medición del consumo de energía eléctrica o el encendido de luces a través de sensores de presencia hacen que las personas tengan mayor conciencia de su entorno y mayor autonomía personal.

D. Ahorro

La automatización produce que los procesos no solo funcionen por sí mismos, sino que funcionen con mayor precisión y exactitud, esto reduce el consumo de recursos físicos como la energía, el agua o el tiempo. Generando un beneficio económico y mejorando la calidad de vida.

E. Conectividad

Hoy en día, no se puede hablar de automatización sin comunicación. Todo sistema automatizado debe tener la capacidad de conectarse con otros sistemas

de igual forma automatizado. La conectividad permite que los sistemas se integren para compartir recurso e información. Y hacen que los sistemas sean adaptables a nuevas tecnologías. En pocas palabras, la conectividad aumenta los beneficios antes expuestos.

F. Sostenibilidad

Una de las principales ventajas en el desarrollo e implementación de sistemas cada vez más automatizados, es la de asegurar las necesidades del presente sin afectar o comprometer las necesidades de futuras generaciones. El uso más eficiente de recursos y la reducción sustancial de carga de trabajo en el hogar ayudara a que las próximas generaciones cuenten con los elementos necesarios con un mejor estilo de vida.

1.5.2. Sistemas a ser automatizados en el hogar

Si bien, en las últimas décadas ha aparecido cantidades de dispositivos automatizados para el hogar, estos no dejan de ser autónomos y carentes de comunicación con el exterior, entre los cuales se pueden contar los aires acondicionados o las máquinas de lavado. Dichos dispositivos han mejorado sin duda la calidad de vida de las personas, pero en el mundo conectado de hoy se enfrentan a nuevos retos como es el Internet de las Cosas.

Dejando de lado estos dispositivos autónomos, las viviendas se componen de otros sistemas mucho más primordiales que se podrían categorizar como de infraestructura, estamos hablando de sistemas como el de electricidad, iluminación, agua potable, gas, accesos, climatización, telefonía, y hoy en día los de datos y seguridades técnicas. Dichos sistemas han permanecido desatendidos, en una gran parte, de ser automatizados o controlados electrónicamente. Esto ya sea por motivos económicos, de conocimiento, de

indiferencia, o porque no se ha encontrado el foco de explotación adecuado. Si bien se viene hablando hace un buen tiempo sobre tecnologías como la Domótica o Inmótica, estas no se popularizaron lo suficiente como las redes de telefonía celulares o el Internet.

En el presente, la idea de automatizar la infraestructura de las viviendas se está convirtiendo cada vez más latente. Esto debido a que hoy en día, conceptos como la sostenibilidad, eficiencia energética, cuidado del medio ambiente y conservación de recursos, son cada vez más importantes y en el futuro tomaran un protagonismo sin igual. La automatización provee de los beneficios adecuados para ajustar los sistemas de infraestructura del hogar a estos nuevos conceptos, ya que los mismos utilizan recursos del exterior para proveer de servicios en el interior de la vivienda. Esto último hace necesario que la infraestructura del hogar consuma energía y recursos de forma más eficiente, la automatización logra esto adicionando otros beneficios como aumento del confort, la seguridad, el control y la integración.

1.5.3. El Internet de las Cosas Industriales como una solución para la automatización en el hogar

La automatización en el hogar es un concepto que se viene desarrollando por un largo tiempo, desde dispositivos como lavadoras, hornos de microondas, refrigeradoras, aires acondicionados, duchas eléctricas, calentadores de agua, detectores de movimiento, cerraduras eléctricas y aspersores de riego, hasta sistemas más avanzados como equipos de telecomunicaciones, computadores, sistemas contra-incendio, cámaras de vigilancia, domótica y servicios de Internet, han aumentado sustancialmente la calidad de vida de las personas como también simplificado trabajos que en el pasado representaban tiempos y gastos.

El Internet de las Cosas abre un nuevo mundo de posibilidades en el hogar, la capacidad de poder conectar todos los dispositivos anteriormente descritos a Internet representa una interacción del ser humano con su entorno como nunca antes se ha visto. Al conectar los dispositivos y sistemas del hogar con la red global es posible integrar recursos e información para mejorar y simplificar procesos, controlar remotamente funciones para mejorar la calidad del tiempo, consumir solo lo necesario para ahorrar dinero y reducir la carga de contaminación al ambiente.

Pero el conectar una gran cantidad de objetos a internet podría hacer que algunas cosas se simplifiquen y otras se vuelcan más complejas y costosas como el aumento del ancho de banda para la transmisión de datos y el crecimiento de las infraestructuras de red. La tendencia hoy en día es la de conectar dispositivos domésticos de forma aislada, con servicios en la nube para desde ahí gestionar su funcionamiento, estado y conexión con otros dispositivos (si es que el fabricante lo permite), esto generara una acumulación de sistemas que se comportan como clientes e intentan al mismo tiempo y por la misma salida transmitir sus datos con cada uno de sus servidores en el exterior. Esto representa grandes problemas no solo para los sistemas de transmisión de datos, sino también para las aplicaciones mismas como retrasos en el cumplimiento de sus funciones, perdidas de datos e incremento en la vulnerabilidad de la información.

Es aquí donde el Internet de las Cosas Industriales jugara un papel protagónico, permitiendo que la conexión entre dispositivos sea directa (punto-a-punto sin servidores intermedios), por eventos y con resiliencia, y que los dispositivos tengan una comunicación a servicios en la nube solo bajo demanda y cuando sea necesario. Esto permitirá que las cosas puedan convivir con las actuales redes de datos y que la creciente interconexión que en el futuro se va a generar en los hogares no sea costosa ni traumatizante.

CAPÍTULO II

DISEÑO DEL SISTEMA DE CONTROL INTEGRADO

2. DISEÑO DEL SISTEMA DE CONTROL INTEGRADO

2.1. ANÁLISIS DEL DEPARTAMENTO TIPO SUITE

El presente proyecto de grado contempla el diseño e implementación de un sistema de control en el hogar basado en los principios de Internet de las Cosas Industriales, innovador concepto que propone como principal dirección de diseño la distribución de la inteligencia entre los diferentes elementos que conforman el sistema, como su conexión directa a Internet a través de la misma infraestructura de comunicaciones para datos.

Para el desarrollo del proyecto de automatización se escogió un departamento tipo suite de 55.77 m^2 ubicado en el cuarto piso del edificio residencial Valtelina y con dirección entre las calles Elia Liut y Avenida Brasil, parroquia La Concepción, norte de la ciudad de Quito. En la **Figura 1** se aprecia la localización geo-referenciada del inmueble.

Para esta etapa de diseño se realizó un análisis de las condiciones actuales de la vivienda, como un levantamiento de las necesidades de automatización y seguridad que esta requiere, para posterior realizar una definición del sistema, red de control, dispositivos a ser empleados, instalación de los mismos, y diseño de los servicios de control y aplicación de interfaz de usuario.

Figura 1 Localización del Departamento Tipo Suite

Fuente: (Recuperado de Google Earth, 2016).

2.1.1. Estado actual de la vivienda

La vivienda está conformada por una cocina de 13.15 m^2, sala-comedor de 19.64 m^2, un dormitorio de 12.43 m^2, un baño de 4.03 m^2 y un cuarto de lavado de 2.53 m^2. Se encuentra completamente acondicionada y cuenta con servicios de energía eléctrica, agua potable, gas centralizado, telefonía e Internet de banda ancha con enlace de última milla de fibra óptica, conectado a un router de frontera que provee de una red inalámbrica WiFi para conexión de dispositivos en el interior del inmueble. La vivienda cuenta con sistema convencionales de iluminación, accesos, energía y ningún sistema de seguridad técnica ni contra intrusión.

El sistema de iluminación está conformado por 14 circuitos de luminarias, tanto para iluminación directa, indirecta y decorativa. El sistema de energía eléctrica es completamente convencional conformado por una red bifásica de tomacorrientes 110VAC 60Hz. La distribución y protección de los circuitos de fuerza e iluminación se lo realiza desde termo-magnéticos enchufados a un tablero eléctrico o térmico instalado en el cuarto de lavado, donde se encuentra la acometida eléctrica de la vivienda.

Actualmente el sistema de accesos está conformado por una puerta anti-vandálica como única vía de ingreso y salida de ocupantes, tres ventanas de vidrio batiente, y un sistema de llamado por pulsador-timbre. Las maquinas con las que cuenta la vivienda son un refrigerador, un horno de microondas, un calentador de agua a gas o calefón, una cocina con horno a gas, un extractor de olores para la cocina y una lavadora automática. El edificio donde se encuentra la vivienda provee de un sistema centralizado de GLP[13] que alimenta la cocina, su horno y el calefón, todos instalados en la cocina.

En el **APÉNDICE A** de este documento se puede encontrar un registro fotográfico del estado actual de la vivienda. Mientras que en el **APÉNDICE B** es posible apreciar en detalle los planos arquitectónicos, eléctricos y de iluminación del inmueble, como también el diagrama de distribución de los circuitos de la vivienda.

2.1.2. Levantamiento de información

De forma general la vivienda requiere de aplicaciones para el control de la iluminación, ocupación y accesos, seguridad contra intrusión e incendios, mejorar el consumo de energía eléctrica, notificaciones de eventos de forma remota,

[13] Siglas de *Gas Licuado de Petróleo.*

registro de variables en la nube (Aplicación IoT) y gestión de la vivienda desde una interfaz que se pueda ejecutar de forma remota.

A continuación, se recoge todas las necesidades de automatización y seguridad por estancia, habitación o zona que requiere la vivienda, y que servirán como directriz para el diseño de la red de control y servicios de control:

A. Cocina

Los puntos de interés en la cocina son dos circuitos de iluminación, una ventana y los puntos de alimentación de GLP para la cocina y el calefón. Como requerimientos tenemos:

- Control de ocupación con encendido automático de luces para el circuito de iluminación principal.
- Detección y aviso de incendios, fuga de gas y contaminación de monóxido de carbono, este último específicamente por la presencia del calefón.
- Control de ventilación de la zona. Debido a la presencia del calefón se requiere que la habitación se mantenga constantemente ventilada. El medio de ventilación es natural a través de la ventana (no es requerido ningún otro medio artificial), por lo que se requiere que la ventana se mantenga constantemente abierta cuando la vivienda este ocupada.
- Control contra-intrusión en la ventana cuando la vivienda se encuentre desocupada.

B. Sala-Comedor

La sala y el comedor de la vivienda se encuentran en una sola habitación, como también el acceso principal de la vivienda. Los puntos de interés que tenemos en estas zonas son tres circuitos de iluminación en la sala, un circuito de iluminación

sobre el comedor, un circuito de iluminación en el ingreso, la puerta de acceso y una ventana. Como requerimientos se tiene:

- Control de ocupación con encendido automático de luces para el circuito de iluminación principal de la sala.
- Funciones de encendido y apagado de los circuitos de iluminación para reducir número de mandos.
- Detección y aviso de incendio.
- Control de acceso en la puerta, con fácil apertura por medio de pulsadores tanto para el ingreso como para la salida. El cierre automático de la puerta no debe ser temporizado.
- Encendido de luces de bienvenida cuando se ingrese en la vivienda.
- Control contra-intrusión en la puerta y ventana cuando la vivienda se encuentre desocupada.
- Control del confort a través de la medición y registro de la luminosidad, temperatura y humedad relativa de la estancia.

C. Dormitorio

El dormitorio contempla puntos de interés como tres circuitos de iluminación y una ventana. Los requisitos encontrados en esta zona son:

- Control de ocupación con encendido automático de luces para el circuito de iluminación principal de la estancia.
- Desactivación del encendido automático de luces por control manual, específicamente para el instante de ir a dormir.

- Apagado automático de luces cuando sea de día, esto debido a que durante el periodo de sueño se deja la iluminación indirecta encendida.
- Detección y aviso de incendio.
- Control contra-intrusión en la ventana cuando la vivienda se encuentre desocupada.

D. Baño

Como único punto de interés en esta zona se tiene un circuito de iluminación integrado con un extractor de olores. Como requisito se tiene:

- Encendido temporizado del circuito de iluminación y extractor de olores.
- Aviso de una contaminación de monóxido de carbono, específicamente para periodos de ducha.

A parte de los requerimientos de automatización y seguridad antes expuestos se contemplan otros aplicados a toda la vivienda, como son:

- Control sobre la utilización de la energía eléctrica con apagado automático de luces no indispensables cuando exista un exceso de consumo.
- Control sobre la desconexión y reconexión de los circuitos de fuerza para reducir el consumo de corrientes “parasitas” de los dispositivos constantemente enchufados.
- Desactivación y activación de los encendidos automáticos de luces cuando sea de día o de noche respectivamente.
- Simulación de presencia en la vivienda para prevenir robos y atracos cuando la vivienda se encuentre desocupada.

- Notificación de alarmas de forma remota vía correo electrónico y mensajes SMS. Especialmente para alarmas de incendio, intrusión y exceso de consumo.
- Medición y registro de la demanda y consumo de la energía eléctrica para análisis de eficiencia energética con el fin de ahorrar energía y mejorar hábitos de consumo.
- Control y gestión remota de la vivienda a través de una interfaz de usuario

2.2. DEFINICIÓN DEL SISTEMA DE CONTROL INTEGRADO

De acuerdo a las necesidades recogidas en el levantamiento de información, *Sección 2.1.2 Levantamiento de información*, se definen los siguientes sistemas de control a ser diseñados e implementados para el presente proyecto. Las siguientes definiciones serán bases para el diseño de la red de control y de los servicios de control, descritos en las siguientes secciones de este capítulo. Cada uno de los sistemas definidos a continuación complementan al sistema de control integrado de la vivienda, y han sido pensados para cumplir con los principios de horizontalidad, control distribuido e integración de recursos.

2.2.1. Sistema de control de iluminación

El sistema de control de iluminación consiste básicamente en el control de los circuitos de iluminación a través de la conexión de sus retornos con conmutadores o relés de carga instalados en los controladores de la red. Esto permite virtualizar el encendido y apagado de las luminarias para poder integrarse a otros dispositivos, automatismos o funcionalidades por medio de los servicios de control de la red IzoT.

Las cargas de iluminación conectadas al sistema podrán ser controladas desde pulsadores, controladores de ocupación y aplicación cliente de la red de control.

2.2.2. Sistema de mandos

El sistema de mandos está conformado por pulsadores que sustituyen a los interruptores del sistema de iluminación convencional. Los pulsadores se conectan a entradas digitales de los controladores, desde donde es posible programar con el mismo mando más de una funcionalidad. Se definen dos funcionalidades de mando independientes por pulsador para el control de dos circuitos de iluminación o cualquier otro dispositivo, ajuste o funcionalidad que permita un control conmutado. Estas funciones son: pulsación corta y pulsación larga.

La pulsación corta es un liberado rápido del pulsador (menor a un segundo por defecto), mientras que la pulsación larga es un liberado retardado del mismo (mayor a un segundo por defecto). Cada función podrá ser ajustada para ejecutarse con retroalimentación (con confirmación del estado del elemento a ser controlado) o como interruptor (con activación y desactivación en cada lapso).

2.2.3. Sistema de control de ocupación

El control de ocupación consiste en la determinación de si una estancia se encuentra ocupada o no. Esto es posible de determinar a través de la conexión de sensores de movimiento para control automático y/o pulsadores para control manual. El fin del control de ocupación es el de encender las luces de una estancia ante la ocupación de la misma, como su apagado ante la desocupación con el fin de ahorrar energía y mejorar el confort.

Se definen dos formas de control de ocupación de una estancia: instantáneo e indefinido. La ocupación instantánea es una permanencia de corto plazo que es controlada a través de un temporizador que se activa cuando se detecta una desocupación. La indefinida, por el contrario, es una permanencia de largo plazo o sin límite de tiempo. Ambas formas de ocupación podrán ser operados desde un sensor detector de presencia o un pulsador dependiendo de la funcionalidad de la estancia o de la existencia del operador. Por ejemplo, en la cocina la ocupación instantánea puede llevarse a cabo por un sensor de movimiento y la indefinida por un pulsador, en otro caso, en el baño tanto la ocupación instantánea como la indefinida puede ser operada con un único pulsador (pulsación corta para la instantánea, pulsación larga para la indefinida).

Como parte del control de ocupación se encuentra la funcionalidad de “control crepuscular”. Este consiste en la detección de los periodos de luz y oscuridad de un día a través de un sensor de luminosidad para discriminar el encendido automático o instantáneo de las luminarias. En otras palabras, las luminarias no deben encenderse de forma automática durante el día (no hay necesidad por la presencia de luz natural) pero por la noche sí. Por conceptos de control, el encendido manual no podrá ser discriminado por el control crepuscular, aunque ante una detección del amanecer las luces encendidas se apagaran por precaución.

El control de ocupación también provee la funcionalidad de “simulación de presencia” para el caso en el que los ocupantes no se encuentren en la vivienda. Este consiste en el encendido y apagado aleatorio o programado de luces para dar la sensación de que la estancia se encuentra “ocupada”. La simulación de presencia se utiliza como medida de seguridad preventiva, alertando a posibles atracantes desde el exterior de que la vivienda se encuentra ocupada.

2.2.4. Sistema de control de eficiencia energética

Este sistema se encuentra conformado por tres funcionalidades bien definidas y está orientado a reducir y mejorar el consumo de energía eléctrica. La primera funcionalidad es el “Control Sobre la Demanda” que consiste en la medición de la corriente eléctrica de cada fase de la vivienda desde la acometida de entrada con el fin de realizar control automático de cargas. El control de demanda es llevado a cabo a través del apagado de circuitos de iluminación cuando se excede el valor de “máxima demanda” determinado para la vivienda. El sistema podrá ejercer control automatizado por demanda sobre dos grupos de luminarias con prioridad. Un primer grupo, conformado por los circuitos de iluminación menos críticos (luces indirectas y decorativas), y un segundo grupo conformado por cargas más prioritarias como las luces de la cocina o el comedor. Si luego de apagar ambos grupos de luminarias, no se consigue reducir el consumo con referencia del umbral de máxima demanda se dispara una alarma de sobre-demanda y notificaciones vía correo electrónico. No se considera la notificación vía SMS por no ser una alarma critica.

Como segunda funcionalidad del sistema se tiene el “Control Sobre Los Circuitos De Fuerza” de la vivienda (red de tomacorrientes). Un contactor instalado en la fase B de la acometida de la vivienda podrá desconectar o reconectar todos los circuitos de fuerza (a excepción de la refrigeradora y extractor de olores que se encuentran en la fase A), dependiendo de una señal de control. Esto con el objetivo de desconectar, ante una desocupación total de la vivienda o por la noche, todas las cargas conectadas a la red de tomacorrientes con el fin de cortar consumos “parásitos” de dispositivos inductivos o capacitivos.

La ultima funcionalidad es el análisis histórico y estadístico de la demanda y consumo de la energía con el fin de detectar hábitos y gestionar el ahorro. Esto se llevará a cabo con el registro periódico en la nube de las mediciones de

corrientes y potencia total (Aplicación IoT), y consultas y reportes desde la aplicación cliente.

2.2.5. Sistema de control contra-incendios

Su función fundamental es la detección y alarma de incendios a través de la supervisión de sensores detectores de humo y GLP. Como también el control sobre contaminaciones de monóxido de carbono a través de sensores detectores de CO. Ante un disparo de cualquiera de estas alarmas el sistema generara señales de estrobo y sirena para los circuitos de iluminación y timbre respectivamente, con el fin de comunicar a los ocupantes de la incidencia. Como también notificaciones vía SMS y correo electrónico.

Las funciones de estrobo y sirena consisten en generar parpadeos y pulsos sonoros en las luces y timbre de la vivienda respectivamente. Esto con el objetivo de comunicar la alarma o incidencia tanto de forma visual como acústica. Por convención y normativas los sistemas contra-incendios están obligados a dar ambas formas de aviso por si existe alguna persona con limitaciones sensitivas.

Para el caso de la alarma de fuga de gas no se contempla el aviso de estrobo debido a que puede generar un “arco eléctrico” en los terminales de las luminarias y generar un accidente aun peor. También, ante esta alarma se contempla la desconexión de los circuitos de fuerza por las mismas razones.

Una parte fundamental en un sistema de incendios es el monitoreo de la operación de los diferentes sensores, para lo cual el sistema medirá a través de una entrada digital la presencia de tensión de la línea de alimentación de los sensores. A través de una salida digital o relé de carga, el sistema controlara la alimentación de los sensores para inhibición de alarmas y reseteo de los dispositivos luego del reconocimiento de la incidencia. Dicha salida también será

supervisada por el sistema para detección de fallos en la reconexión de la alimentación. Los fallos por reconexión y falta de tensión en la línea de alimentación serán anunciados a través de la función de sirena y notificaciones vía correo electrónico.

Dentro del levantamiento de información se consideró la supervisión de la ventilación del calentador de agua o calefón que se encuentra instalado en la cocina de la vivienda. Como medida de prevención para un posible envenenamiento por exceso de CO. La ventilación del área se la realiza de forma natural a través de la ventana de la cocina, por lo que es necesario controlar que dicha ventana se mantenga abierta. Esta función se la realizara a través de un contacto magnético instalado en la ventana.

La función de control de ventilación de la cocina podría ser contradictorio a las funciones del sistema de control contra-intrusión. Pues este supervisa que los accesos de la vivienda se mantengan cerrados (incluyendo la ventana de la cocina). Para evitar esto, ambos sistemas se integrarán por medio de una señal de control para desactivar la supervisión de ventilación cuando se active o arme la vigilancia del sistema contra-intrusión.

2.2.6. Sistema de control de accesos

En una primera instancia, consiste en el control de una cerradura electromagnética instalada en la puerta de ingreso a la vivienda. Junto a este, un contacto magnético proporciona el estado de la puerta, permitiendo un control en lazo cerrado de la misma. El contacto magnético posibilitará un control más preciso de la puerta, permitiendo funciones como un cierre anticipado si se desiste de abrir la puerta después de una orden de apertura, no tener límite de tiempo cuando la puerta ha sido abierta, o una reapertura rápida luego de que la puerta ha sido cerrada.

Cuando se tiene una orden de apertura, el control desenergiza la cerradura electromagnética, seguido de una temporización corta para realizar la apertura manual (si este temporizador culmina sin una detección de apertura de la puerta, energiza nuevamente la cerradura). Una vez que los ocupantes hayan realizado el cierre manual, el contacto magnético da una señal de puerta cerrada, y un nuevo temporizador se ejecuta para dar paso a la energización de la cerradura. Si durante este tiempo existe una reapertura manual el proceso de cierre se reinicia. En cualquier caso, es posible enviar una orden de cerrado para energizar la cerradura. Las ordenes de apertura y cierre se las podrá realizar con la misma señal de control (Ej. Un pulsador).

El control incluye una función de “alarma de puerta abierta” que se activa luego de que el contacto magnético de una señal de puerta abierta. Un temporizador de larga duración se ejecuta para dar paso a un disparo de la alarma, si antes no se cierra la puerta. Esta alarma permitirá detectar problemas físicos con la puerta (Ej. puerta desnivelada), mal cierre de la misma e incluso fallos en el contacto magnético. La alarma de puerta abierta podrá ser inhibida con el correcto cierre de la puerta y podrá ser anunciada a través de la función sirena del timbre y correo electrónico.

El sistema de accesos incluye el control del dispositivo de timbre. La señal de control del timbre podrá ser ejercida desde el pulsador instalado en el exterior de la vivienda por medio de su función de pulsación corta.

La orden de apertura de la puerta puede ser ejercida por dos señales, uno para la apertura desde el interior de la vivienda y otro para la apertura desde el exterior de la vivienda. Ambas señales pueden ser un pulsador u otro dispositivo de acceso. Para la apertura de la puerta desde el interior de la vivienda se conectará una funcionalidad de uno o varios pulsadores de la vivienda. Para la orden de apertura dese el exterior, el control implementará una funcionalidad de

“candado”. El cual permite (candado abierto) o no (candado cerrado) a la señal de apertura desde el exterior tener control sobre la puerta. Esto con la idea de poder abrir la puerta con la función de pulsación larga desde el pulsador del timbre de la vivienda cuando la función de candado se encuentre “abierto”. El fin de la funcionalidad de candado es la de tener una posibilidad de apertura rápida cuando uno de los ocupantes haya abandonado la vivienda por un periodo corto. Tanto la función de candado como la apertura de la puerta se la podrá ejecutar desde la aplicación cliente de la red de control. Por otro lado, la orden de cierre manual de la puerta únicamente se la podrá ejecutar desde el interior de la vivienda.

Por último, el control de acceso proporciona de una función de “luces de bienvenida” para el encendido temporizado de un circuito de iluminación cuando se recibe una orden de apertura desde el exterior. La señal de luces de bienvenida se podrá conectar con el control de ocupación para adicionarle la función de “control crepuscular”.

2.2.7. Sistema de control contra-intrusión

El fin del sistema de control contra-intrusión es el de prevenir y alertar ingresos no permitidos e intentos de robos o atracos sobre la vivienda cuando esta no se encuentra ocupada. Para la detección de intrusiones el sistema recibe señales de todos los sensores de movimiento y contactos magnéticos instalados en puerta y ventanas. El control ejerce una función de vigilancia sobre las diferentes zonas de la vivienda, cuando una vigilancia detecta una actividad, se dispara una alarma que activa funciones de estrobo y sirena en los circuitos de iluminación y timbre respectivamente para ejercer pánico sobre los atracantes. Por otro lado, el control se integra con el sistema de notificaciones para aviso remoto de la incidencia vía SMS y correo electrónico.

La habilitación o armado de la vigilancia contra intrusos se la puede hacer de forma diferenciada, activando la vigilancia por separado para los sensores de movimiento, los contactos magnéticos en ventanas y el contacto magnético en puerta. El armado de las vigilancias se las realiza desde una señal de red que puede ser controlada desde la aplicación cliente.

El sistema contra intrusos incluye señales para integrarse con otros sistemas y ejercer funciones como desactivación de la supervisión de ventilación, simulación de presencia, cerrado de candado y desconexión de circuitos de fuerza.

2.2.8. Sistema de supervisión de confort

Sensores de luminosidad, temperatura y humedad relativa instalados en la sala-comedor de la vivienda, permitirán el monitoreo, análisis y registro en la nube (Aplicación IoT) de la condición de confort en la vivienda. La supervisión será llevada a cabo desde la aplicación cliente en tiempo real y a través de reportes de históricos y estadísticos, consultando al servicio de internet para registro de datos.

2.3. DISEÑO DE LA RED DE CONTROL DISTRIBUIDA

La red de control es esencialmente un conjunto de dispositivos o nodos interconectados con el fin de realizar funciones y automatismos, resultados de la interacción entre estos. Los dispositivos o nodos se clasifican en periféricos, controladores y elementos de red. Los periféricos son los elementos de entrada y salida de la red, están comprendidos por sensores, mandos y actuadores. Los controladores interconectan los periféricos y contienen los servicios de control, protocolo de comunicaciones y aplicaciones. Los elementos de red comprenden dispositivos como switches o routers que permiten la transferencia de paquetes y el enrutamiento de la comunicación.

Todos los elementos de la red trabajan en conjunto para dar como resultado el sistema de control integrado. La presente sección tiene como fin resolver y definir los componentes, conformación y distribución de la red de control a ser implementada, basándose en los principios del Internet de las Cosas Industriales y estructurada como una red de comunicación IEEE 802, TCP/IP e IzoT.

2.3.1. Arquitectura de la red de control

La red de control estará constituida por tres controladores basados en computadores de placa reducida o SBC[14]. Los SBCs son dispositivos con una alta capacidad de procesamiento en un reducido espacio, y pueden ejecutar un sistema operativo de alto nivel basado en Linux como Debian o Ubuntu. Su núcleo es un procesador de arquitectura ARM e integran alguna interfaz de comunicaciones basada en la norma IEEE 802 como Ethernet o WiFi. Los SBCs mantienen la mayoría de características de una computadora convencional y sin duda serán parte de la nueva generación de aplicaciones tecnológicas domesticas en el futuro.

Los SBCs no únicamente poseen características de hardware de alto nivel como interfaces USB[15], HDMI[16] o Ethernet, sino que también interfaces de bajo nivel como I2C[17], PWM[18], ADC[19] y GPIO[20]. Un reducido tamaño, comunicación de banda ancha e interfaces de bajo nivel los hacen idóneos para conformar aplicaciones de automatización y redes de control avanzadas con tecnología TCP/IP.

---

[14] Siglas de *Single Board Computer.*
[15] Siglas de *Universal Serial Bus.*
[16] Siglas de *High-Definition Multimedia Interface.*
[17] Siglas de *Inter-Integrated Circuit,* bus de comunicación inter-placa de muy alta velocidad.
[18] Siglas de *Pulse-Width Modulation.*
[19] Siglas de *Analog-To-Digital Converter.*
[20] Siglas de *General Purpose Input/Output.*

Dentro del presente proyecto se propone la construcción de controladores conformados por un SBC como unidad central de proceso (CPU) y módulos interfaces de I/O como placas de entradas digitales, placas de salidas por relés de carga y placas de entradas analógicas para señales de 0-5V y 0-10V, comunicados con la CPU por medio de un bus serie de datos I2C o puertos GPIO.

Estos estarán dentro de una carcasa plástica y se distribuirán en tres zonas de la vivienda: uno en la cocina, otro en la sala-comedor, y el ultimo en el dormitorio. Los controladores serán instalados sobre el “cielo falso” del tumbado (construido en Gypsum) y se podrá acceder a estos a través de compartimientos. A través de las interfaces de I/O, los controladores se conectarán con los periféricos, especialmente seleccionados para cumplir los sistemas de control propuestos en la *Sección 2.2 Definición del sistema de control integrado*.

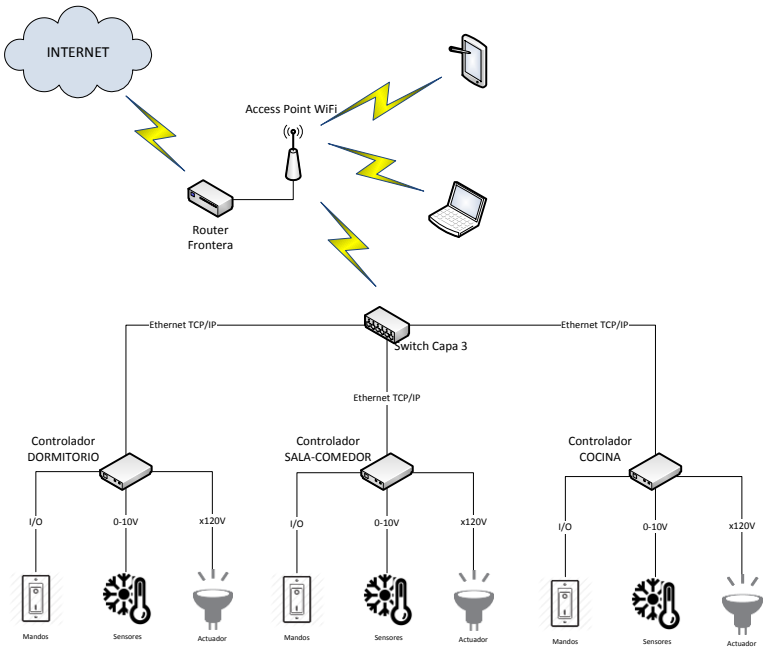


Figura 2 Arquitectura de la red de control distribuida

Fuente: (Elaboración propia en Microsoft Visio, 2016).

La columna vertebral de la red de control estará constituida de una red de datos Ethernet IEEE 802.3 en topología estrella, como se puede apreciar en el diagrama de la **Figura 2**. La red Ethernet estará inter-conectada con un switch repetidor WiFi/Ethernet 10/100/1000, que a su vez se conectará con el router de frontera del proveedor de internet de la vivienda para su eventual comunicación con el exterior. Esto a través de la red inalámbrica WiFi de la vivienda. También se dispondrá de dos fuentes de poder de 12 VDC y 24 VDC para la alimentación de los controladores y periféricos.

2.3.2. Descripción y selección de componentes

2.3.2.1. Controladores

En total se dispondrá de tres controladores ubicados en la cocina, sala-comedor y dormitorio de la vivienda. Cada nodo de control estará conformado por un SBC y módulos de interface I/O con la capacidad de recoger señales de los sensores y mando, y ejercer control sobre los elementos finales. De acuerdo al levantamiento de información realizado en la *Sección 2.1.2 Levantamiento de información* y de acuerdo a los productos que se pueden encontrar en el mercado se ha definido los siguientes componentes que darán forma a cada uno de los controladores:

A. Unidad central de procesos

El CPU de los controladores estará constituido por el ODROID C2 de marca **Hardkernel** de procedencia surcoreana. El ODROID C2 está provisto de un procesador Amlogic ARM® Cortex®-A53 (ARMv8) de 2Ghz y CPU de cuatro núcleos, 2Gbyte de memoria DDR3 SDRAM, socket para tarjetas Micro-SD y eMMC (sistema operativo y almacenamiento), un puerto Gigabit Ethernet y 40 pines para funciones de GPIO, PWM, ADC y bus I2C, entre otros adicionales

como 4 puertos USB, un puerto de video HDMI y un puerto de audio I2S. En la **Figura 3** se describe todas las partes y componentes del SBC.

El ODROID C2 es un potente SBC muy superior en características a varios de sus competidores como Raspberry Pi y BeagleBone. Posee un precio bastante cómodo de $40 y puede ser adquirido desde la tienda on-line del fabricante. (Hardkernel, 2013).

En el **ANEXO D** de este documento se puede encontrar un manual técnico completo del ODROID C2.

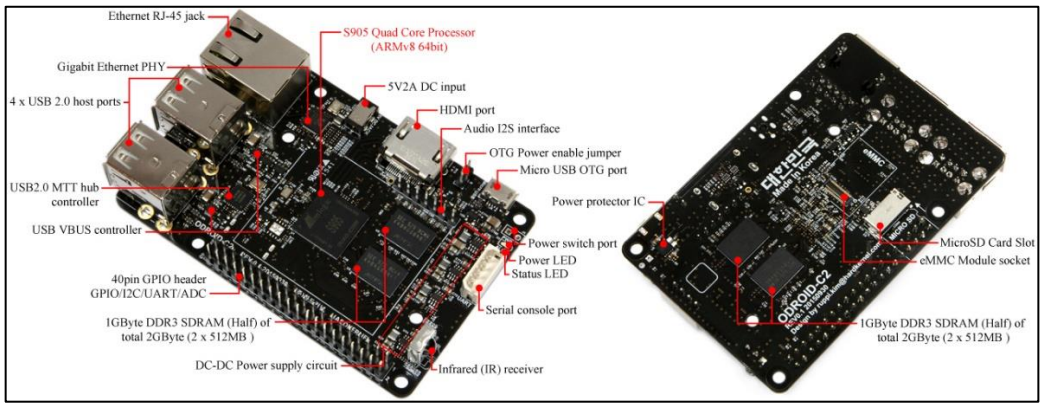


Figura 3 Partes del SBC ODROID C2

Fuente: (Hardkernel, Recuperado de http://www.hardkernel.com/main/_Files/prdt/2016/201602/C2BoardDetail_HR.jpg, 2016).

B. Módulo I2C de 8 entradas digitales

Para este módulo se cuenta con la placa marca **ERE** modelo I2C-IN830S de procedencia tailandesa. El I2C-IN830S es una placa electrónica basada en el controlador PCF8574, provista de 8 entradas detectoras de tensión opto-aisladas de 3 a 30 VDC. Su objetivo es el de recibir el estado de los contactos de

pulsadores, sensores o contactos magnéticos, y enviárselo al CPU a través del bus I2C (ERESHOP, 2017).

En la **Figura 4** se puede apreciar el módulo I2C descrito. También en el **ANEXO D** de este documento se adjunta la ficha técnica de este componente.

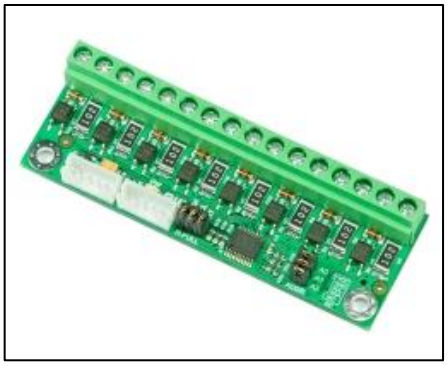


Figura 4 Vista frontal del módulo I2C de 8 entradas digitales

Fuente: (ERESHOP, Recuperado de https://www.ereshop.com/shop/images/medium/products/I2C-IN830S_MED.jpg, 2016).

C. Módulo I2C de 4 entradas analógicas

Para este módulo se cuenta con la placa electrónica marca **ERE** modelo I2C-AI418ML de procedencia tailandesa. Consiste en una placa electrónica basada en el controlador MCP3424, provista de 4 convertidores analógico-digital de hasta 18 bits. Cada entrada puede ser configurada para recibir señales de corriente o voltaje. Su objetivo es el de recibir las señales de sensores analógicos como los de temperatura, humedad relativa, luminosidad o corriente, y transmitirlas al CPU a través de bus I2C (ERESHOP, 2017).

En la **Figura 5** se muestra un ejemplo del módulo I2C descrito. En el **ANEXO D** de este documento se adjunta la ficha técnica de este componente.

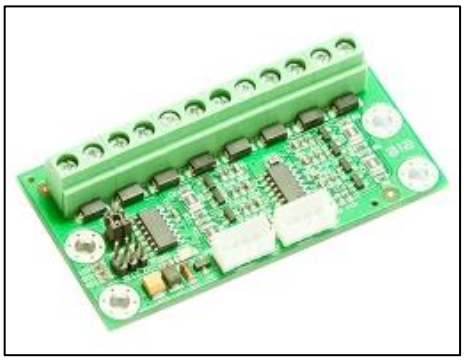


Figura 5 Vista frontal del módulo I2C de 4 entradas analógicas

Fuente: (ERESHOP, Recuperado de https://www.ereshop.com/shop/images/medium/products/I2C-AI418ML_MED.jpg, 2016).

D. Módulo I2C de 8 salidas por relé

Para cumplir con este módulo se cuenta con la placa modelo I2C-RL824M marca **ERE** de procedencia tailandesa. Esta es una placa electrónica basada en el controlador PCF8574, provista de 8 relés de carga de 120VAC 10A. Tiene el objetivo de controlar la conexión y desconexión de cargas como luces, cerraduras o timbre. Recibe la señal de control desde la CPU a través del bus I2C (ERESHOP, 2017).

En la **Figura 6** se puede apreciar el módulo I2C de 8 salidas por relé. Por otro lado, en el **ANEXO D** de este documento se adjunta la ficha técnica de este componente.

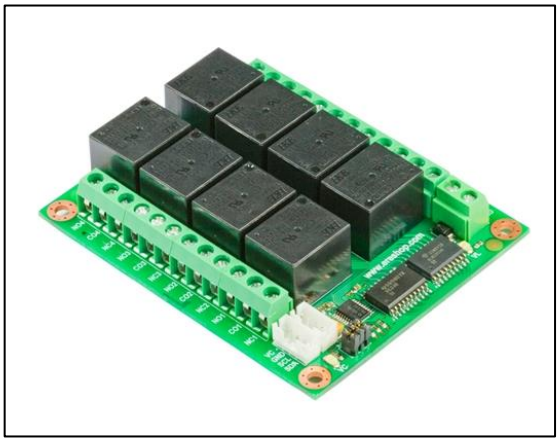


Figura 6 Vista frontal del módulo I2C de 8 salidas por relé

Fuente: (ERESHOP, Recuperado de https://www.ereshop.com/shop/images/large/products/I2C-RL812M_01_LRG.jpg, 2016).

E. Módulo TTL de 4 salidas por relé

Este módulo es una placa electrónica genérica de relés de carga controlados por entradas TTL (nivel de tensión para estado alto 5.0 VDC). Su objetivo es el de controlar cargas como luces, timbre o cerradura. Su control es a través de las salidas digitales GPIOs del CPU. En la **Figura 7** se refleja una imagen de este componente. La información obtenida de este módulo fue provista por el vendedor.

A continuación, algunas características de este módulo:

- 4 canales independientes protegidos con opto-acopladores.
- 4 relés de 1 polo 2 tiros de 120VAC 10A.
- Led indicador para cada canal (enciende cuando la bobina del relé esta activa).
- Activado mediante corriente, el circuito de control debe proveer una corriente de 15 a 20 mA
- Terminales de conexión de tornillo (borneras)

- Terminales de entrada de señal lógica con headers machos.

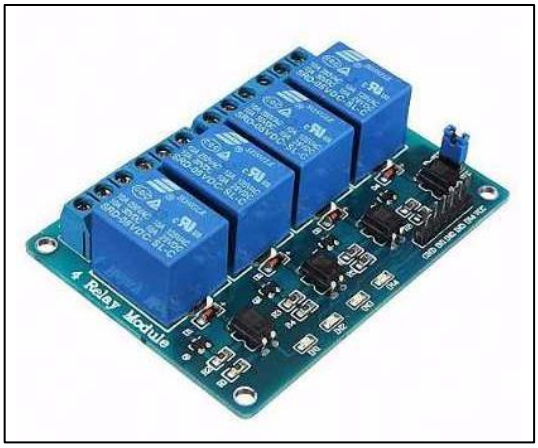


Figura 7 Vista frontal del módulo TTL de 4 salidas por relé

Fuente: (Genérico, Recuperado de http://cdn-tienda.bricogeek.com/3535-thickbox_default/modulo-4-reles-5v.jpg, 2016).

F. Módulo de alimentación DC/DC

Este módulo es una placa electrónica genérica convertidora de tensión DC a DC para la alimentación de la CPU. Su objetivo es el de recibir la tensión de la fuente de poder de 24 VDC y reducirla a 5 VDC para la alimentación de la CPU. En la **Figura 8** se contempla una imagen de esta placa. La información obtenida de este módulo fue provista por el vendedor.

A continuación, algunas características de este módulo:

- Indicador LED de alimentación.
- Soporta cargas de máximo 5A.
- Voltaje de entrada: 4-38VDC
- Voltaje de salida variable: 1.25-36VDC
- Temperatura de trabajo: -40 a +85 grados.

- Frecuencia de trabajo: 180KHz
- Eficacia de conversión: 96%

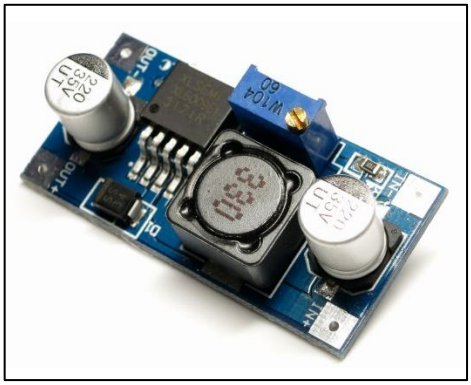


Figura 8 Vista frontal del módulo de alimentación DC/DC

Fuente: (Genérico, Recuperado de http://www.ilhanelektronik.com.tr/image/data/dc-dc%20converter%20xl6009.png, 2016).

2.3.2.2. Periféricos

Los periféricos están conformados por los mandos, sensores y actuadores a ser instalados en la vivienda, incluidos los circuitos de iluminación a ser automatizados. Se ha seleccionado los periféricos necesarios de acuerdo al levantamiento realizado en la *Sección 2.1.2 Levantamiento de información* y de acuerdo a los productos que se pueden encontrar en el mercado.

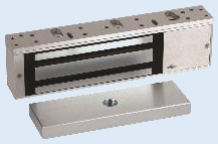
La **Tabla 2** muestra una descripción de dichos componentes, para mayor detalle referirse al **ANEXO E** de este documento donde se encuentran sus respectivas fichas técnicas.

Tabla 2

Lista de selección de periféricos

No.	Periférico	Imagen
1	**Placa de Pulsadores** **Marca:** Btcino **Modelo:** Mátix Placa modular de hasta 3 unidades para pulsadores NA 1-polo 127-227V 10A.	
2	**Sensor Detector de Movimiento** **Marca:** Paradox **Modelo:** Paradome DG467 Detector de movimiento infrarrojo con elemento de doble oposición, cobertura 360º 7m x 6m a una altura de 2.4m, salida por relé NC, alimentación 9-16VDC 29mA(12VDC).	
3	**Doble-Sensor de Movimiento y Luminosidad** **Marca:** E-Controls **Modelo:** E-Multisensor 0-10V ▪ Detector de movimiento PIR 360º, salida por relé con contacto libre de potencial 2-polos (NC-C-NO) 250VAC 5A. ▪ Sensor de luminosidad, rango de medida 5 a 1000 luxes, resolución 1 lux, salida de tensión variable 0-10VDC, alimentación 24VDC 26mA.	
4	**Sensor Detector de Humo** **Marca:** VEXHUM **Modelo:** VEXHUM Detector de humo fotoeléctrico 4 hilos, alimentación 9-35VDC 35mA(12VDC), salida por relé con contacto libre de potencial 2-polos (NC-C-NO).	

Continua

5	**Sensor Detector de GLP** **Marca:** Cofem **Modelo:** KEEPER DAGR110 Detector de gas licuado de petróleo, propano y butano, límite inferior de explosividad (LEL) 10%, alimentación 110VAC 3W, salida por relé con contacto libre de potencial 2-polos (NC-C-NO).	
6	**Sensor Detector de CO** **Marca:** Cofem **Modelo:** KEEPER DACOR110 Detector de monóxido de carbono, sensibilidad de detección 300ppm, alimentación 110VAC 3W, salida por relé con contacto libre de potencial 2-polos (NC-C-NO).	
7	**Doble-Sensor de Temperatura y Humedad Relativa** **Marca:** Guangzhou gzlexiang **Modelo:** AW3010 ▪ Sensor de Temperatura, rango de medida -20 a 80 ºC, resolución 0.1 ºC salida de tensión variable 0-10VDC. ▪ Sensor de Humedad Relativa, rango de medida 0 a 99.9 %RH, resolución 0.1 %RH, salida de tensión variable 0-10VDC, alimentación 15-36VDC 15mA.	
8	**Contacto Magnético** **Marca:** FLAIR **Modelo:** LMS13-1 Montaje en superficie, función de contacto NC, conexión eléctrica hasta 100VDC 500mA, colores blanco y marrón.	
9	**Cerradura Electromagnética** **Marca:** ZKSoftware **Modelo:** AL-280 Led Cerradura electromagnética 280kg(600Lbs), conexión eléctrica 12VDC 500mA.	

Continua

10	**Sensor Transductor de Corriente Eléctrica** **Marca:** Veris Industries **Modelo:** Hawkeye 922 Sensor transductor de corriente eléctrica, núcleo partidos, rango de medición seleccionable 0 a 30/60/120 AC 60Hz, salida de tensión variable 0-5VDC.	
11	**Contactor 3-Polos** **Marca:** SASSIN **Modelo:** LC1 - D9511 Contactor 3-Polos 95A, tensión de la bobina 120VAC 60Hz.	

Fuente: (Elaboración propia, 2016).

2.3.2.3. Infraestructura de red

La infraestructura de la red está conformada por el router de frontera, el switch repetidor para la interconexión física de la red Ethernet y las fuentes de poder para la alimentación de los controladores y periféricos. La **Tabla 3** especifica los elementos a ser utilizados, para mayor detalle referirse al **ANEXO F** de este documento.

2.3.3. Distribución de la red de control

En el **APÉNDICE B** de este documento se presentan los planos de diseño de la ubicación y conexión de los diferentes componentes de la red de control dentro de la vivienda. En el Plano de Control se puede observar la ubicación de los dispositivos de control como periféricos, controladores y elementos de red, como también la disposición de las líneas de control por donde se conectarán los periféricos con los controladores. Por otro lado, en el Plano de Potencia se pude observar la ubicación de los elementos finales de control que trabajan a niveles

de tensión de 120 VAC como luminarias y timbre. En este último, se observa la disposición de los circuitos de iluminación y timbre como la conexión de su retorno con su controlador respectivo. En ambos planos se puede distinguir la disposición de las líneas de comunicación y alimentación que conecta los controladores con el switch repetidor y la caja de control respectivamente.

Tabla 3

Lista de selección de la infraestructura de red

No.	Descripción	Imagen
1	**Router de Frontera (ISP)** **Marca:** Huawey **Modelo:** HG8045H Gateway ONT, interfaz óptica SC/APC interface Class B+, 2 antenas externas de 2dBi, red inalámbrica 802.11b/g/n.	
2	**Switch Repetidor WiFi/Ethernet** **Marca:** Linksys **Modelo:** RE6500 Router/Switch/Repetidor WiFi con 4 puertos Ethernet Gigabit 10/100/1000Mbps, 2 antenas externas de 2dBi y soporte para conexión con redes IEEE 802.11 a/b/g/n	
3	**Fuente de Poder de 12VDC** **Marca:** Mean Well **Modelo:** DR -120-12 Fuente de poder de 12VDC 10A 120W.	
4	**Fuente de Poder de 24VDC** **Marca:** Mean Well **Modelo:** DR -30-24 Fuente de poder de 24VDC 1.5A 36W.	

Los planos también demuestran la nomenclatura e identificación a ser empleada en cada uno de los componentes de la red de control con el fin de distinguir cada uno de ellos en el resto del diseño e implementación de todo el sistema.

2.3.4. Composición de la red de control

En el **APÉNDICE B** se expone el unifilar de la red de control con la disposición de sus elementos y líneas de conexión. En él se presentan tres tipos líneas:

Línea de Control: esta línea contiene los cables e hilos que transmiten las señales de estado y medición de los sensores como su alimentación. Para el caso de la cerradura electromagnética y los sensores de humo, sus líneas de control, transmiten la alimentación controlada de los mismos.

Línea de Potencia: esta línea contiene los retornos de los circuitos de iluminación y timbre a ser controlados por medio de la conexión y desconexión de su alimentación de 120 VAC. También contempla la alimentación de los sensores detectores de GLP y CO.

Línea de Comunicación: esta línea interconecta los controladores con el switch repetidor por medio de path cords.

Línea de Alimentación: esta línea transmite la alimentación de los controladores y periféricos desde las fuentes de poder en la caja de control. Dos niveles de tensión estarán dispuestos en la línea: 12 y 24 voltios.

En la **Tabla 4** y **Tabla 5** se resume la composición de la red de control. En estas se describe el número, cantidades e identificación de todos los componentes de la red.

2.3.5. Conexionado de los controladores

En el **APÉNDICE B** se muestran planos con la estructura y conexionado de cada uno de los controladores a ser instalados en la cocina, sala-comedor y dormitorio.

Tabla 4

Resumen del número e identificación de componentes por zona

No.	Componente	Cantidad	Identificación
	Cocina		
1	Controlador	1	CD01
2	Placa de Pulsadores (una unidad)	1	1B01-1
3	Sensor Detector de Movimiento	1	1MD01
4	Sensor Detector de Humo	1	1SD01
5	Sensor Detector de GLP	1	1GD01
6	Sensor Detector de CO	1	1CO01
7	Contacto Magnético	1	1MC01
8	Sensor Transductor de Corriente Eléctrica	2	1A01 1A02
9	Circuito de Iluminación	1	c
10	Timbre 120VAC	1	1DB01
11	Contactor 3-Polos 95A	1	1PC01
12	Switch Repetidor WiFi/Ethernet	1	SW01
13	Fuente de Poder 12VDC	1	PW12
14	Fuente de Poder 24VDC	1	PW24

Continua

Sala-Comedor			
15	Controlador	1	CD02
16	Placa de Pulsadores (una unidad)	2	2B01 2B02
17	Placa de Pulsadores (dos unidades)	1	2B03-1, 2B03-2
18	Doble-Sensor de Movimiento y Luminosidad	1	2TH01
19	Sensor Detector de Humo	1	2SD01
20	Doble-Sensor de Temperatura y Humedad Relativa	1	2ML01
21	Contacto Magnético	2	2MC01, 2MC02
22	Circuito de Iluminación	6	b, e, d, f, g, h
23	Cerradura Electromagnética	1	2DL01
Dormitorio-Baño			
24	Controlador	1	CD03
25	Placa de Pulsadores (una unidad)	2	3B01 3B02
26	Placa de Pulsadores (tres unidades)	1	3B03-1, 3B03-2, 3B03-3
27	Sensor Detector de Movimiento	1	3MD01
28	Sensor Detector de Humo	1	3SD01
29	Contacto Magnético	1	3MC01
30	Circuito de Iluminación	6	i, j, k, l, m, n

Fuente: (Elaboración propia, 2016).

Cada controlador está constituido de un CPU (SBC) y módulos interfaces de I/O que se interconectan a través de un bus serie I2C, con la única excepción del

nodo de la cocina, que se conecta con el módulo de salidas de relé a través de señales digitales o GPIOs.

Tabla 5

Resumen de cantidades de componentes utilizados

No.	Componente	Cantidad Total
1	Controlador	3
2	Placa de Pulsadores	7
3	Sensor Detector de Movimiento	2
4	Doble-Sensor de Movimiento y Luminosidad	1
5	Sensor Detector de Humo	3
6	Sensor Detector de GLP	1
7	Sensor Detector de CO	1
8	Doble-Sensor de Temperatura y Humedad Relativa	1
9	Contacto Magnético	4
10	Sensor Transductor de Corriente Eléctrica	2
11	Circuito de Iluminación	13
12	Cerradura Electromagnética	1
13	Contactor 3-Polos 95A	1
14	Switch Repetidor WiFi/Ethernet	1
15	Fuente de Poder 12VDC	1
16	Fuente de Poder 24VDC	1

Fuente: (Elaboración propia, 2016).

2.3.6. Conexionado de periféricos y elementos de red

En el **APÉNDICE C** también se muestran planos con la disposición del conexionado de los controladores con sus periféricos. Estos planos están destinados a identificar en que salida o entrada de cada controlador debe ser conectado cada periférico de la red.

Por otro lado, se presenta un plano con el diseño de la estructura, distribución y conexionado de la caja de control con sus elementos y accesorios, entre estos: los sensores transductores de corriente, las fuentes de poder, el contactor, bornas y termo-magnéticos.

2.4. DISEÑO DE LOS SERVICIOS DE CONTROL IZOT

En la *Sección 1.2.2. Servicios de control de LonTalk/IP* se describió y conceptualizo los servicios de control provistos por LonTalk/IP, como las bases para la creación de redes IzoT. Luego de definir los componentes, estructura e infraestructura de la red de control a ser instalada en la sección anterior, se procede con el diseño de los servicios de control, como también de las funcionalidades y automatismos del sistema de control según se expone en la *Sección 2.2. Definición del sistema de control integrado.*

El concepto de IzoT está basado en la utilización de bloques funcionales que representan los controles y periféricos del sistema de control, cada bloque funcional agrupa datapoints de entradas y salidas que representan, a su vez, a las variables físicas o lógicas del control o periférico al cual representan. Datapoints de salida se conectan virtualmente con datapoints de entrada para transmitir información, visto de una manera más computacional, los datapoints de entrada se suscriben a los datapoints de salida, cuando estos últimos publican un evento en la red, los datapoints de entrada reciben el evento y lo identifican

como suyo a través del direccionamiento detallado en la cabecera del evento o trama.

Los bloques funcionales se construyen a partir de un perfil funcional el cual se declara y programa a través de Python y clases heredadas del SDK de IzoT. En cada perfil funcional se declaran los datapoints con su tipo de dato correspondiente.

Cada bloque funcional está asociado a un programa funcional, el cual contiene la lógica de las funcionalidades y los automatismos a ser desempeñados por dicho bloque funcional. Estos programas se desarrollan en lenguaje Python y se instancian junto a los bloques funcionales en la aplicación principal del controlador.

Durante esta sección se definirá la estructura y modelado de los bloques funcionales, y se diseñará los programas funcionales de dichos bloques con el fin de cumplir con las funcionalidades y automatismos descritos en la *Sección 2.2. Definición del sistema de control integrado.*

2.4.1. Modelado de los bloques funcionales

En el **APÉNDICE C** de este documento es posible encontrar el Manual Funcional completo con cada uno de los bloques funcionales diseñados para este proyecto. En dicho Manual Funcional se puede apreciar un diagrama de cada bloque funcional, junto con una lista de los datapoints de entrada y salida, y su respectiva descripción.

2.4.2. Diseño de los programas funcionales

El diseño de los programas funcionales esta presentado en diagramas de flujos, que paso a paso muestran las condiciones y procesos de cada programa.

Como consideraciones de diseño, los procesos de los programas funcionales se desarrollan con un control de estados, en el cual, dependiendo de las condiciones de las entradas, un programa salta de un estado a otro, generando el resultado deseado en las salidas.

Otras de las consideraciones de diseño a tener en cuenta es el desarrollo de los procesos sin condiciones bloqueantes, todas las condiciones de un programa transcurren sin ejecutar secuencias de bucle o lazos, entrando y saliendo del programa durante cada interacción de la aplicación principal del controlador (la única sentencia en bucle es la de la aplicación principal del controlador que se ejecuta de forma infinita como se puede apreciar en la **Figura 9**). De esta manera cada controlador puede alojar una gran cantidad de bloques funcionales y programas, y responder en tiempo real a los eventos generados en la red. En la **Figura 9** de muestra un diagrama del funcionamiento de la aplicación principal de los controladores.

Figura 9 Diagrama de proceso de aplicación principal

Fuente: (Elaboración propia en Microsoft Power Point 2013, 2016).

Un detalle importante a tener en cuenta en los programas de los bloques funcionales es la utilización de temporizadores que realizan funciones de retardos temporales como armado de una vigilancia, espera de una condición o disparo de una alarma. Estos temporizadores tampoco son bloqueantes y se ejecutan en segundo plano, devolviendo su estado al programa que lo implementa cuando esta lo requiera. De esta manera, los programas pueden controlar y consultar a sus temporizadores a través de funciones como INICIAR, REINICIAR, EN EJECICIÓN o DETENER.

Los diagramas de flujo de los programas funcionales relacionado con cada uno de los bloques funcionales de la sección anterior se pueden apreciar en el **APÉNDICE D** de este documento.

2.4.3. Distribución de los servicios de control

En la *Sección 2.3. Diseño de la red de control distribuida* se realizó una definición y diseño de los controladores y periféricos a ser instalados en la vivienda, y que conformaran el sistema de control integrado. Cada controlador está destinado a la integración de los periféricos de su área como a la ejecución de las funcionalidades, protocolo de comunicación y servicios de control.

En la **Tabla 6**, **Tabla 7** y **Tabla 8** se hace una distribución de los bloques funcionales por zona o estancia, controlador, periférico al que estará asociado (si lo tiene) y nombre con la cual será instanciado e implementado dentro de la aplicación del controlador y base de datos de la red LonTalk/IP.

Tabla 6

Distribución de bloques funcionales del controlador CD01

Estancia		**Cocina**	
Controlador		CD01	
No.	**Bloque Funcional**	**Perfil Funcional**	**Descripción**
1	Button_1B01_1	UFPT_button	Asociado al pulsador 1B01-1.
2	Switch_1MD01	UFPT_switch	Asociado al detector de movimiento 1MD01.
3	Switch_1SD01	UFPT_switch	Asociado al detector de humo 1SD01.
4	Switch_1GD01	UFPT_switch	Asociado al detector de gas 1GD01.
5	Switch_1CO01	UFPT_switch	Asociado al detector de CO 1CO01.
6	Switch_1MC01	UFPT_switch	Asociado al contacto magnético 1MC01.
7	Switch_1DC01	UFPT_switch	Asociado al estado de la línea de alimentación de 12VDC.
8	Load_1PW01	UFPT_load	Asociado a la salida RO0 de CD01 que controla la alimentación de los sensores de incendios.
9	Current_1A01_A	UFPT_electric_current	Asociado al transductor de corriente eléctrica 1A01.
10	Current_1A02_B	UFPT_electric_current	Asociado al transductor de corriente eléctrica 1A02.
11	Doorbell_1DB01	UFPT_bell	Asociado al timbre 1DB01.
12	Load_1PC01	UFPT_load	Asociado al contactor de potencia 1PC01.
13	Lighting_C	UFPT_lighting	Asociado al circuito de iluminación *c*.
14	Occupancy_1S01	UFPT_occupancy_control	Implementa el control de ocupación para de principal circuito de iluminación de la cocina.
15	Fire_1S02	UFPT_fire_control	Implementa el control de detección de incendios.
16	Energy_1S03	UFPT_energy_control	Implementa el control de demanda.

Tabla 7

Distribución de bloques funcionales del controlador CD02

Estancias	**Sala-Comedor**		
Controlador	CD02		
No.	**Bloque Funcional**	**Perfil Funcional**	**Descripción**
1	Button_2B01_1	UFPT_button	Asociado al pulsador 2B01-1.
2	Button_2B02_1	UFPT_button	Asociado al pulsador 2B02-1.
3	Button_2B03_1	UFPT_button	Asociado al pulsador 2B03-1.
4	Button_2B03_2	UFPT_button	Asociado al pulsador 2B03-2.
5	Switch_2ML01	UFPT_switch	Asociado al detector de movimiento 2ML01.
6	Brightness_2ML01	UFPT_brightness	Asociado al sensor de luminosidad 2ML01.
7	Temp_2TH01	UFPT_temperature	Asociado al sensor de temperatura 2TH01.
8	RH_2TH01	UFPT_relative_humidity	Asociado al sensor de humedad relativa 2TH01.
9	Switch_2SD01	UFPT_switch	Asociado al detector de humo 2SD01.
10	Switch_2MC01	UFPT_switch	Asociado al contacto magnético 2MC01.
11	Switch_2MC02	UFPT_switch	Asociado al contacto magnético 2MC02.
12	Load_2PW01	UFPT_load	Asociado a la alimentación de los sensores de incendios.
13	Load_2DL01	UFPT_load	Asociado a la cerradura electromagnética 2DL01.
14	Lighting_B	UFPT_lighting	Asociado al circuito de iluminación *b*.
15	Lighting_D	UFPT_lighting	Asociado al circuito de iluminación *d*.

Continua

16	Lighting_E	UFPT_lighting	Asociado al circuito de iluminación *e.*
17	Lighting_F	UFPT_lighting	Asociado al circuito de iluminación *f.*
18	Lighting_G	UFPT_lighting	Asociado al circuito de iluminación *g.*
19	Lighting_H	UFPT_lighting	Asociado al circuito de iluminación *h.*
20	Occupancy_2S01	UFPT_occupancy_control	Implementa el control de ocupación para el principal circuito de iluminación de la sala.
21	Occupancy_2S02	UFPT_occupancy_control	Implementa el control de ocupación de entrada.
22	Access_2S03	UFPT_access_control	Implementa el control de accesos para la puerta.
23	Vigilance_2S04	UFPT_vigilance_control	Implementa el control de vigilancia de intrusión.

Fuente: (Elaboración propia, 2016).

Tabla 8

Distribución de bloques funcionales del controlador CD03

Estancias		**Dormitorio-Baño**	
Controlador		CD03	
No.	**Bloque Funcional**	**Perfil Funcional**	**Descripción**
1	Button_3B01_1	UFPT_button	Asociado al pulsador 3B01-1.
2	Button_3B02_1	UFPT_button	Asociado al pulsador 3B02-1.
3	Button_3B03_1	UFPT_button	Asociado al pulsador 3B03-1.
4	Button_3B03_2	UFPT_button	Asociado al pulsador 3B03-2.
5	Button_3B03_3	UFPT_button	Asociado al pulsador 3B03-3.

Continua

6	Switch_3MD01	UFPT_switch	Asociado al detector de movimiento 3MD01.
7	Switch_3SD01	UFPT_switch	Asociado al detector de humo 3SD01.
8	Switch_3MC01	UFPT_switch	Asociado al contacto magnético 3MC01.
9	Load_3PW01	UFPT_load	Asociado a la salida RO0 de CD03 que controla la alimentación de los sensores de incendios.
10	Lighting_I	UFPT_lighting	Asociado al circuito de iluminación *i*.
11	Lighting_J	UFPT_lighting	Asociado al circuito de iluminación *j*.
12	Lighting_K	UFPT_lighting	Asociado al circuito de iluminación *k*.
13	Lighting_L	UFPT_lighting	Asociado al circuito de iluminación *l*.
14	Lighting_M	UFPT_lighting	Asociado al circuito de iluminación *m*.
15	Lighting_N	UFPT_lighting	Asociado al circuito de iluminación *n*.
16	Occupancy_3S01	UFPT_occupancy_control	Implementa el control de ocupación para principal circuito de iluminación el dormitorio.
17	Occupancy_3S02	UFPT_occupancy_control	Implementa el control de ocupación para el circuito de iluminación del baño.
18	EMail_3S03	UFPT_email_notification	Implementa el módulo de notificaciones de incidencias por correo electrónico con el servicio de Gmail.
19	SMS_3S04	UFPT_sms_notification	Implementa el módulo de notificaciones de incidencias por mensajes de texto SMS con el servicio de Twilio.
20	IOT_3S05	UFPT_iot_application	Implementa la aplicación IoT para el registro de variables en la nube con el servicio e ThingSpeak.

Fuente: (Elaboración propia, 2016).

2.5. DISEÑO DE LA APLICACIÓN CLIENTE MULTIPLATAFORMA

2.5.1. Fundamentos de la aplicación cliente

Para el desarrollo y programación de la aplicación cliente de la red de control integrada con tecnología IzoT, se utilizará la tecnología de software JavaFX. La aplicación será multiplataforma pero su principal objetivo es el de ser operado desde un sitio web y accedido desde cualquier parte del mundo.

A pesar que la red de control funciona bajo una arquitectura punto-a-punto, la aplicación cliente trabajará bajo una arquitectura cliente-servidor con el API REST del Servidor IzoT y hará las veces de un Sistema Empresarial y Aplicación IoT de la vivienda como se puede apreciar en la **Figura 10**. Desde la aplicación cliente se podrá monitorear y controlar todos los estados y alarmas de los dispositivos del sistema, administrar los controles y seguridades implementados, y generar reportes y estadísticos de la supervisión del confort y eficiencia energética.

La aplicación cliente también estará provista de un control de acceso por usuario/contraseña, esto gracias a que el Servidor IzoT provee de una gestión de usuarios para el manejo de la red de control a través del API REST. Para poder utilizar el API REST de la red de control es necesario estar en una sesión con un usuario dado de alta y con los permisos adecuados. El API REST del Servidor IzoT permite también la gestión de usuarios como altas y bajas. La aplicación cliente utilizará este API para iniciar sesión tanto en la misma aplicación como en la red de control, como también para la configuración de usuarios.

En resumen, la aplicación cliente funcionará como una aplicación nativa o programa de un sistema operativo (Windows y Linux) y como una aplicación web en un navegador de internet, se conectará a los API REST tanto del Servidor IzoT

de la red de control para la lectura/escritura de los controladores y datapoints, como al de la plataforma ThingSpeak para la consulta de los datos registrados de luminosidad, temperatura y humedad relativa, como también de demanda y potencia consumida de la vivienda.

2.5.2. Arquitectura de la aplicación cliente

JavaFX apoya el desarrollo de aplicaciones a través del patrón de diseño y arquitectura MVC o Modelo – Vista – Controlador. La arquitectura MVC propone la separación del código de un software en tres capas, cada una con una responsabilidad definida y denominadas modelos, vistas y controladores. El MVC permite que las aplicaciones con alguna interfaz de usuario grafica tengan mayor facilidad de mantenimiento, reutilización de código y separación de conceptos.

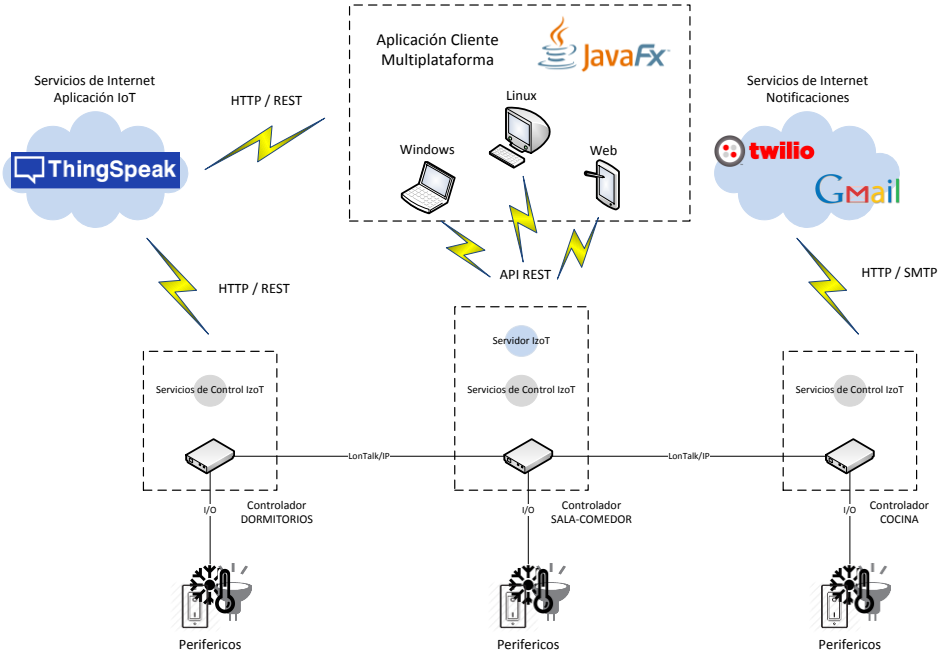


Figura 10 Arquitectura virtual del Sistema Integrado

Fuente: (Elaboración propia en Microsoft Visio 2013, 2016).

El patrón de diseño MVC es ampliamente utilizado en la ingeniería de software como una arquitectura de procesos que permite un desarrollo definido y modular, separando los procesos y tareas para el ingreso y salidas de datos (lógica de negocio), la transformación y aplicación de los datos (lógica de la aplicación) y la presentación de los datos.

En la **Figura 11** se muestra un diagrama con cada una de las capas del MVC, en este se puede apreciar como cada capa puede comunicarse con sus otras dos capas para compartir o solicitar datos e información. La entrada del programa es la capa de Controladores, la cual obtiene los datos necesarios de la capa de Modelos, los procesa y los pasa a la capa de Vistas para su presentación, siendo esta la salida del programa. La capa de Modelos obtiene los datos de algún agente externo como una base de datos o servicio, visto de otra forma, el Modelo es la capa que implementa interfaces y conectores con otros programas. La capa de Vistas también puede solicitar directamente los datos a la capa de Modelos para su presentación directa, esto cuando los datos deban presentarse de forma automática y sin ningún proceso intermedio.

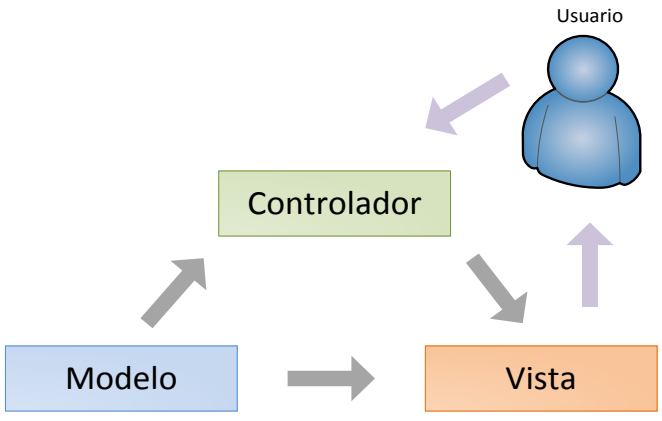


Figura 11 Funcionamiento de la arquitectura MVC

Fuente: (Elaboración propia en Microsoft Visio 2013, 2016).

La aplicación cliente será desarrollada con la tecnología JavaFX y utilizará el patrón de diseño MVC como arquitectura del programa. Java permite la agrupación y separación de las tareas, funciones, objetos y procesos de un programa en clases y paquetes. Por lo tanto, Java permite que se puede definir cada una de las capas del MVC en paquetes y clases. Clases de una capa puede ser instanciada por clases de otra capa cumpliendo así los principios de comunicación entre capas y reutilización de código.

2.5.3. Definición de la aplicación cliente

Dentro de un contexto global la aplicación cliente realizará tres tareas bien definidas: la supervisión y control de los diferentes sistemas y elementos que conforman la red de control, la administración y gestión de usuarios de la aplicación y red de control, y la generación de reportes y estadísticos de datos enviados al servicio de internet ThingSpeak para análisis de confort y eficiencia energética en la vivienda (Aplicación IoT). La **Figura 12** muestra un cuadro con una descripción de las diferentes funciones a ser realizadas por cada una de las tareas descritas anteriormente.

La aplicación cliente será desarrollada bajo una arquitectura de capas MVC, en cada capa se desarrollan clases y módulos que implementan las lógicas y funciones de cada una de las tareas. A continuación, se describe una definición de cada capa de la aplicación con sus módulos, clases y funciones.

2.5.3.1. Definición del modelo

La capa de modelos es aquella donde se trabaja con los datos, proporciona mecanismos para acceder a la información y también para actualizar su estado. También es la capa donde se implementa la lógica de negocio (reglas y discriminantes para la validación de los datos).

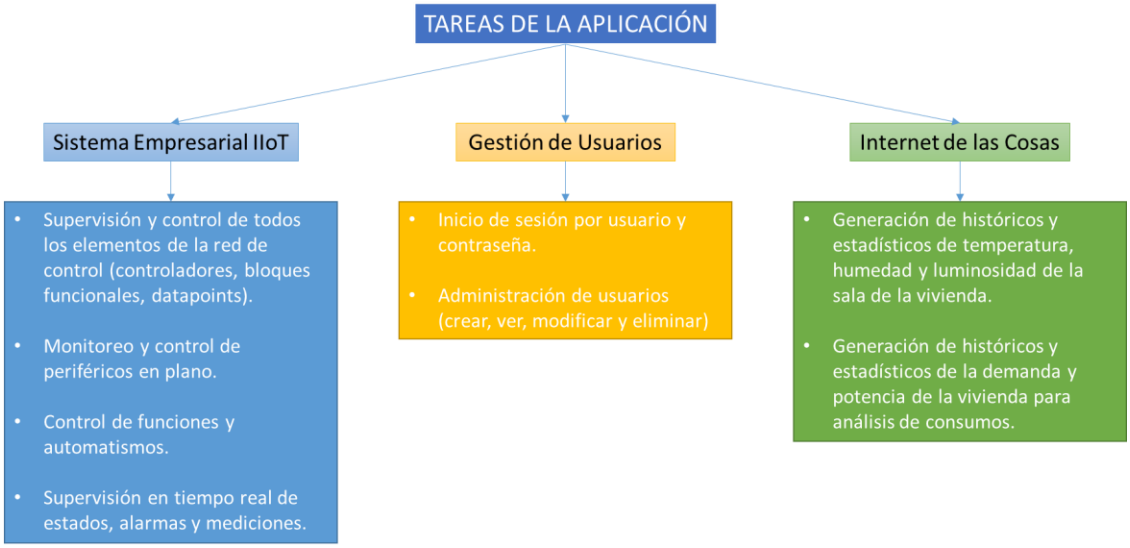


Figura 12 Funciones y tareas de la aplicación cliente

Fuente: (Elaboración propia en Microsoft Power Point 2013, 2016).

El modelo de la aplicación cliente proporcionará de clases para la manipulación de los datos, envió y recepción de los datos, y validación de los mismos. Básicamente la capa de modelos se clasifica en cuatro subcapas que se describen a continuación:

A. Value-Objects

Los Value-Objects o VO son las clases o beans que se utilizan como representación de los datos y sirven para el flujo de la información entre capas. La **Tabla 9** muestra una descripción de las clases objetos a ser implementados.

Tabla 9

Clases VO del modelo de la aplicación cliente

No.	Clase	Descripción
		Variables
1	Switch	Un enumerador que contiene los estados de una variable tipo SNVT_switch.
2	Alarm	Un enumerador que contiene los estados de un variable tipo SNVT_switch. Se utilizara como valor de los datapoint que cumple la función de transmitir una alarma.
3	Occupancy	Un enumerador que contiene los estados de una variable tipo SNVT_occupancy.
4	Setting	Un enumerador que contiene los estados de una variable tipo SNVT_setting.
5	Measure	Clase padre para la construcción de objetos que representen variables de tipo escalar.
6	Temp	Clase que contiene el valor de una variable tipo SNVT_temp_p.
7	LevPercent	Clase que contiene el valor de una variable tipo SNVT_lev_percent.
8	Lux	Clase que contiene el valor de una variable tipo SNVT_lux.
9	Amp	Clase que contiene el valor de una variable tipo SNVT_amp.
10	Power	Clase que contiene el valor de una variable tipo SNVT_power
		Usuario
11	User	Clase bean con los atributos de un usuario de la red de control: nombre d usuario, contraseña, nombre, apellido e e-mail.
		Red de Control
12	Node	Clase padre para la construcción de los objetos que representan los elementos de la red de control.
13	Datapoint	Clase que representa un datapoint, contiene atributos como nombre, tipo de dato, dirección (entrada o salida) y el valor que puede ser un objeto descrito en el apartado Variables de esta tabla.

Continua

14	Block	Clase que representa un bloque funcional, contiene una lista de clases Datapoint.
15	Controller	Clase que representa un controlador de la red, contiene una lista de clases Block.
16	Network	Clase que representa la red de control. Contiene una lista de clases Controller.
	Internet de las Cosas	
17	Feed	Es una clase con un par timestamp-value que contiene un dato o medida de un campo del canal de la vivienda de ThingSpeak.
18	Field	Clase que representa un campo del canal de la vivienda de ThingSpeak.

Fuente: (Elaboración propia, 2016).

B. Conector

El conector es básicamente una clase que contiene los métodos de conexión con el servicio. Para el caso de la aplicación cliente se utilizará una única clase de tecnología, protocolo y formato para la petición y recepción de los datos que es API REST y JSON, tanto para conectarse con el Servidor IzoT como con ThingSpeak. Por lo tanto, se implementará una única clase con los métodos GET, PUT, POST y DELETE.

Tabla 10

Clases del conector del modelo de la aplicación cliente

No.	Clase	Descripción
1	Connection	Clase que implementara un cliente HTTP para la petición/respuesta con los servicios API REST. Él envió, recepción y devolución de los datos los realizara en un formato JSON por lo que utilizara clases que representen objetos JSON.

Fuente: (Elaboración propia, 2016).

C. Control de accesos

El control de accesos contiene clases que, a través de los métodos del conector, realiza la consulta de datos y envía las acciones propias de cada tarea de la aplicación (ver **Tabla 11**). Estas clases también realizan funciones de conversión y corrección de los datos recibidos o a ser enviados desde y hacia el conector. En otras palabras, el control de acceso especifica métodos para cada una de las peticiones que serán necesarias por la aplicación y transforma los VO en estructuras JSON y viceversa.

Para el caso del manejo de la red de control (Sistema Empresarial IIoT), el control de accesos implementa métodos para la lectura de datapoints de salida, como para el control de datapoint de entrada. En el caso de la gestión de usuarios, construye un modelo CRUD (create, read, update y delete) para la administración de los usuarios, y para el caso del IoT implementa métodos para la consulta de los campos (temperatura, humedad relativa, luminosidad, demanda de las fases y potencia eléctrica consumida) almacenados en el canal de la vivienda en ThingSpeak y los devuelve en arreglos para su cálculo y presentación.

Tabla 11

Clases del control de acceso del modelo de la aplicación cliente

No.	Clase	Descripción
1	NetworkAccessControl	Clase que contiene los métodos para las peticiones de lectura y actualización para la Red de Control.
2	UserAccessControl	Clase que contiene un modelo CRUD para la administración de la base de datos de usuarios.
3	IoTAccessControl	Clase que contiene métodos para solicitar arreglo de datos entre fechas determinadas a ThingSpeak, de cada campo o medición de estancia y energía que ha enviado la red de control.

D. Lógica de negocio

La lógica de negocio no es más que una envolvente del control de accesos que implementa métodos que ejecutan validación de los datos de entrada y salida de las funciones de los controles de accesos. Ante errores en los datos ingresados o devueltos, el negocio ejecuta excepciones que podrán ser mostradas en un panel de notificaciones. En la **Tabla 12** se describe las clases que serán implementadas dentro de esta subcapa.

Tabla 12

Clases de la lógica de negocio del modelo de la aplicación cliente

No.	Clase	Descripción
1	NetworkBusinessLogic	Clase que construye métodos para validar las peticiones y respuestas de los métodos de la clase NetworkAccessControl.
2	UserBusinessLogic	Clase que construye métodos para validar las peticiones y respuestas de los métodos de la clase UserAccessControl. Implementa también las funciones necesarias para el logueo o inicio de sesión.
3	IoTBusinessLogic	Clase que construye métodos para validar las peticiones y respuestas de los métodos de la clase IoTAccessControl.

Fuente: (Elaboración propia, 2016).

2.5.3.2. Definición de la vista

Las Vistas contienen el código de la aplicación que va a producir la visualización de las interfaces de usuario. JavaFX contiene una librería completa de controles, paneles, layouts, charts y handles, que permiten la construcción de atractivas y ligeras interfaces, que junto con las clases Property, Binding y

ObservableList permiten implementar una programación por eventos ante entradas y salidas de usuario.

La interfaz de la aplicación cliente estará constituida por un panel de bordes conformado por un menú de navegación, un árbol con la red de control, un panel de notificaciones, una barra de funciones y un panel de trabajo. Al ingreso o arranque de la aplicación, una pantalla de inicio de sesión permitirá la validación de un usuario a través del nombre y contraseña para dar paso al panel principal. En la **Figura 13** se muestra un diagrama con el flujo de la aplicación, como también una distribución del panel principal.

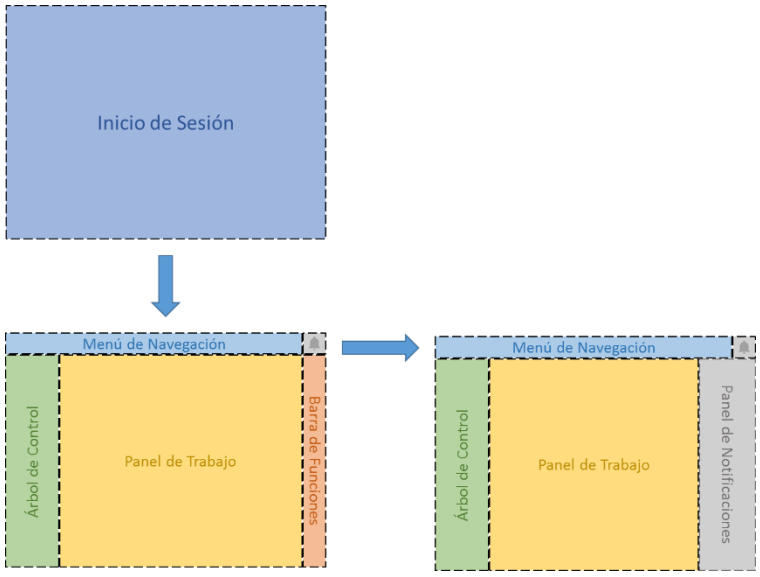


Figura 13 Flujo principal de la aplicación cliente

Fuente: (Elaboración propia en Microsoft Power Point 2013, 2016).

A continuación, se define cada una de las partes de la vista de la aplicación, como las acciones y tareas esperadas.

A. Pantalla de inicio de sesión

Esta pantalla permite la validación o logueo de un usuario. Dos cuadros de textos permitirán el ingreso del nombre de usuario y contraseña. En la **Tabla 13** se muestra una descripción de la clase que implementará esta pantalla.

Tabla 13

Clases de la pantalla de inicio de sesión de la aplicación cliente

No.	Clase	Descripción
1	LoginPane	Clase que proporcionara los controles y diseños necesarios para el inicio de sesión del usuario.

Fuente: (Elaboración propia, 2016).

B. Menú de navegación

El menú de navegación será una barra que estará conformado de iconos que permitirán el ingreso a las diferentes interfaces del panel de trabajo. Junto a este menú, en la misma barra, estará una etiqueta con el nombre completo del usuario logueado y un botón que permitirá abrir el panel de notificaciones. En la **Tabla 14** se describe la clase que implementará este menú.

Tabla 14

Clases del menú de navegación de la aplicación cliente

No.	Clase	Descripción
1	NavigationMenu	Clase que proporcionara los controles y diseños necesarios para construir el menú de navegación.

Fuente: (Elaboración propia, 2016).

C. Árbol de control

En el costado izquierdo de la pantalla principal estará un árbol con la jerarquía de la red de control. El árbol contendrá el total de datapoints de la red de control, agrupados en sus bloques y controladores. Desde el árbol de control se permitirá el control de los datapoints de entrada, de acuerdo a su tipo de dato, y la lectura de los valores de los datapoints de salida (ver **Tabla 15**).

Tabla 15

Clases del árbol de control de la aplicación cliente

No.	Clase	Descripción
1	NetworkTree	Clase que proporcionara los controles y diseños necesarios para construir el árbol de control.

Fuente: (Elaboración propia, 2016).

D. Barra de funciones

La barra de funciones estará al costado derecho de la pantalla principal y permitirá un acceso o control rápido de ciertas funciones, seguridades y automatismos de la red de control como armado/desarmado de la supervisión de incendio, vigilancia de intrusión, función de sueño, simulación de presencia, desbloqueo de la función de candado, desconexión de la red de potencia, apagado total de las luces o apertura de la puesta de acceso (ver **Tabla 16**).

E. Panel de notificaciones

Este es un panel retráctil, ubicado en el extremo derecho del panel principal. Contendrá una lista histórica con los diferentes mensajes de éxito o fallos enviados por los controladores y lógicas de negocio luego de una petición o

acción. Cada que una nueva notificación se genere, el botón de notificaciones en la barra del menú de navegación cambiara su estado a alerta (ver **Tabla 17**).

Tabla 16

Clases de la barra de funciones de la aplicación cliente

No.	Clase	Descripción
1	FunctionBar	Clase que proporcionara los controles y diseños necesarios para construir la barra de funciones.

Fuente: (Elaboración propia, 2016).

Tabla 17

Clases del panel de notificaciones de la aplicación cliente

No.	Clase	Descripción
1	NewsView	Clase que proporcionara los controles y diseños necesarios para construir el panel de notificaciones.

Fuente: (Elaboración propia, 2016).

F. Panel de trabajo

En este panel central se mostrarán las actividades y tareas de la aplicación. En total suman cuatro que son: supervisión y control en plano de todos los periféricos, reporte de histórico y estadístico de la supervisión de estancia, reporte de histórico y estadístico de la eficiencia energética y administración de usuarios. En la **Tabla 18** se describen las clases que implementarán los paneles de trabajo.

Tabla 18

Clases del panel de trabajo de la aplicación cliente

No.	Clase	Descripción
1	HomeMapPane	Clase que proporcionara los controles y diseños necesarios para construir el panel de supervisión y control en plano. Esta vista contendrá un plano de la vivienda sobre el cual podrá supervisarse y controlarse estados, alarmas y mediciones de los periféricos de la red de control a través de iconos.
2	RoomReportPane	Clase que proporcionara los controles y diseños necesarios para construir el panel de reporte de confort de estancia. Esta vista contendrá un gráfico lineal para histórico de mediciones de estancia (temperatura, humedad, luminosidad), un gráfico de barras para estadístico de promedios, y gráficos de pasteles para estadísticos porcentuales de promedios de cada medición. Además de una tabla de resultados (mínimos, máximos, promedios).
3	EnergyReportPane	Clase que proporcionara los controles y diseños necesarios para construir el panel de reporte de eficiencia energética. Esta vista contendrá un gráfico lineal para histórico de mediciones de energía (demanda de fases, potencia total), un gráfico de barras para estadístico de consumos (kWh), y gráficos de pasteles para estadísticos porcentuales de consumos. Además de una tabla de resultados (mínimos, máximos, promedios, consumo total).
4	UserAdminPane	Clase que proporcionara los controles y diseños necesarios para construir el panel de administración de usuario. Desde esta clase podrá ejecutarse el CRUD de los usuarios como visualizar los usuarios registrados en una lista.

Fuente: (Elaboración propia, 2016).

2.5.3.3. Definición de controladores

Los Controladores contienen el código necesario para responder a las acciones que se solicitan en la aplicación. En la aplicación cliente, la capa de Controladores implementará clases y eventos que realizaran las tareas de respuesta del usuario y presentación de la información.

Cada vista de la aplicación tiene asociado una clase controlador, donde es instanciado la clase de la vista y la clase de la lógica de negocio adecuada. A través de los métodos de estas clases controladoras se implementarán los eventos de las vistas, las consultas al modelo y las lógicas de aplicación necesarias. Una descripción de estas clases de presenta en la **Tabla 19**.

Para la consulta y peticiones a los servicios se implementarán clases hilos para no bloquear la aplicación. Una de estas clases ejecutara un servicio continuo para consultar el estado de los datapoints de la red de control con el fin de mantener la información en tiempo real.

Tabla 19

Clases controladoras de la aplicación cliente

No.	Clase	Descripción
1	Main	Clase principal de la aplicación. En esta clase se implementa las clases Scene y Stage de JavaFx y se ejecuta la aplicación. Instancia la clase Coordinator.
2	Coordinator	En esta clase se instancia y construye todos los controladores, vistas y lógicas de negocio. En esta clase también se construye el panel principal de la aplicación y se controla el menú de navegación.
3	LoginController	Clase que implementa las lógicas de la clase LoginPane.
5	NetworkController	Clase que implementa las lógicas de la clase NetworkTree
6	FunctionController	Clase que implementa las lógicas de la clase FunctionBar
7	NewsController	Clase que implementa las lógicas de la clase NewsView
8	HomeMapController	Clase que implementa las lógicas de la clase HomeMapPane.
9	RoomReportController	Clase que implementa las lógicas de la clase RoomReportPane.
10	EnergyReportController	Clase que implementa las lógicas de la clase EnergyReportPane.
11	UserAdminController	Clase que implementa las lógicas de la clase UserAdminPane.
12	QueryService	Esta clase es un hilo que permite la implementación de consultas y respuesta para el manejo de las lógicas de negocio. Gracias a esta clase la aplicación no se bloqueará ante una consulta a servicios.
13	RealTimeService	Esta clase es un hilo que continuamente permanece consultando y actualizando los objetos asociados a las variables de la red de control.

Fuente: (Elaboración propia, 2016).

CAPÍTULO III

IMPLEMENTACIÓN DEL SISTEMA DE CONTROL INTEGRADO

3. IMPLEMENTACIÓN DEL SISTEMA DE CONTROL INTEGRADO

3.1. INSTALACIÓN DE LA RED DE CONTROL DISTRIBUIDA

Para la instalación de la red de control distribuida se ha considerado el diseño de la *Sección 2.3 Diseño de la red de control distribuida,* en el cual se expone los planos, diagramas y elementos necesarios para la construcción e instalación tanto de los controladores como de la red misma. La red conformada de periféricos, controladores y elementos de red, se expande e interconecta a través de la vivienda tipo suite por medio de las líneas de potencia, control, comunicaciones y alimentación.

3.1.1. Construcción y conexionado de los controladores

La instalación de la red de control empezó con la construcción de los controladores, su definición, diseño y componentes se exponen en las *Secciones 2.3.2. Descripción y selección de componentes y 2.3.5. Conexionado de los controladores.* Partiendo de estos, se presenta a continuación el armado y conexionado de cada uno de los tres controladores que conforman la red de control.

3.1.1.1. Construcción del controlador CD01 (cocina)

El controlador CD01 de acuerdo a las necesidades y diseño está conformado por una carcasa plástica de protección, un SBC, un módulo de alimentación DC/DC, un módulo I2C de 8 entradas digitales, un módulo I2C de 4 entradas analógicas y un módulo TTL de 4 salidas por relé, como se aprecia en la **Figura 14**. En el **APÉNDICE E** de este documento se muestra un registro fotográfico de la construcción y cableado paso a paso del controlador CD01.

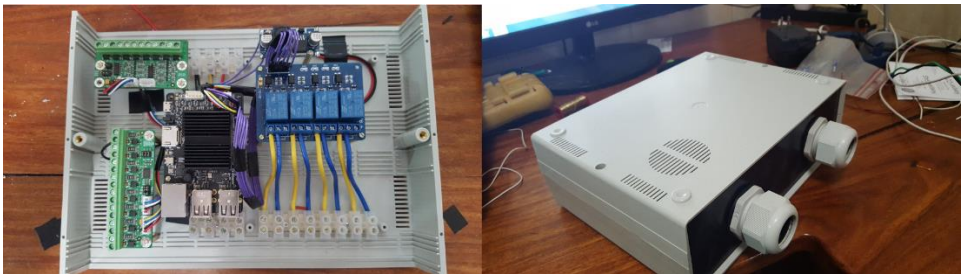


Figura 14 Muestra de la construcción y cableado del controlador CD01

Fuente: (Elaboración propia, 2017).

3.1.1.2. Construcción del controlador CD02 (sala-comedor)

El controlador CD02 de acuerdo a las necesidades y diseño está conformado por una carcasa plástica de protección, un SBC, un módulo de alimentación DC/DC, un módulo I2C de 8 entradas digitales, un módulo I2C de 4 entradas analógicas y un módulo I2C de 8 salidas por relé, como se puede apreciar en la **Figura 15**. En el **APÉNDICE E** de este documento se muestra un registro fotográfico de la construcción y cableado paso a paso del controlador CD02.

Figura 15 Muestra de la construcción y cableado del controlador CD02

Fuente: (Elaboración propia, 2017).

3.1.1.3. Construcción del controlador CD03 (dormitorio-baño)

El controlador CD03 de acuerdo a las necesidades y diseño está conformado por una carcasa plástica de protección, un SBC, un módulo de alimentación DC/DC, un módulo I2C de 8 entradas digitales y un módulo I2C de 8 salidas por relé, como se expone en la **Figura 16**. En el **APÉNDICE E** de este documento se muestra un registro fotográfico de la construcción y cableado del controlador CD03.

Figura 16 Muestra de la construcción y cableado del controlador CD03

Fuente: (Elaboración propia, 2017).

3.1.2. Cableado de líneas e instalación de periféricos

El siguiente paso dentro de la instalación de la red de control fue el cableado de las diferentes líneas y la instalación de los periféricos. Para el cableado se consideró la ubicación referencial y conexionado de los periféricos, controladores y elementos de red, expuestos en la *Sección 2.3.3. Distribución de la red de control* y *Sección 2.3.4. Composición de la red de control* respectivamente. Se realizaron los trabajos de desconexión y reubicación de los retornos de los circuitos de iluminación y timbre (línea de potencia), paso de cable UTP categoría 5e para las líneas de control, paso de cable UTP categoría 6a para la red Ethernet desde la ubicación de los controladores hasta la ubicación del switch repetidor (línea de comunicación), y paso de cable número 18 AWG para la línea de alimentación. Una vez, hecho el cableado se procedió con la conexión de los diferentes periféricos. En el **APÉNDICE E** de este documento se muestra un registro fotográfico de los trabajos antes mencionados.

3.1.3. Instalación de los controladores y elementos de red

Finalmente, se termina la instalación de la red de control con los trabajos de armado, conexionado e instalación de la caja de control, instalación del switch repetidor, ponchado de los cables de red, conexionado de los periféricos a los controladores y su posterior guardado sobre el “cielo falso” de la vivienda.

Para la parte de armado, conexionado e instalación de la caja de control se consideró el diseño desarrollado en la Sección *2.3.6. Conexionado de periféricos y elementos de red.* De igual forma, para el conexionado de los controladores se utilizó como referencia los diagramas expuestos en el **APÉNDICE B**. Un registro fotográfico de estos trabajos se muestra en el **APÉNDICE E** de este documento.

3.2. INSCRIPCIÓN Y CONFIGURACIÓN DE LOS SERVICIOS DE INTERNET

Los servicios de internet, dentro del presente proyecto, representan uno de los principales aplicativos, debido a la integración del sistema de control de la vivienda basada en el concepto IIoT con servicios informáticos que se prestan en la nube de forma abierta, aprovechando de una forma eficiente lo ya desarrollado hoy en día.

Para el presente proyecto, se requiere la utilización de servicios que permitan notificar incidencias y desarrollar una aplicación del Internet de las Cosas para almacenar datos de sensores en a la nube. En la *Sección 1.3. Servicios de internet* se describe de forma teórica cada uno de los servicios a ser empleados, con sus aplicativos y ventajas. En esta sección se pretende mostrar los trabajos que fueron realizados para su inscripción, configuración y utilización.

3.2.1. Configuración de la cuenta de correo electrónico Gmail

Gmail es un servicio de correo electrónico provisto por **Google** que posee la gran ventaja de ser aceptado por la mayoría de servidores de recepción de correo electrónico debido a la confiabilidad que **Google** provoca. Los servicios de correo electrónico utilizan el protocolo SMTP para su transmisión, por lo que para integrar la red de control IzoT únicamente será necesario la utilización de módulos Python que permitan el manejo de dicho protocolo.

Obtener una cuenta de *Gmail* es muy sencillo, basta con abrir una cuenta de *Google Accounts* y se podrá contar con el servicio de correo electrónico. *Gmail* utiliza el protocolo SMTP con comunicación cifrada STARTTLS, autentificación por usuario y contraseña, y para conectarse al servicio se debe apuntar a la dirección *smtp.gmail.com* por el puerto 587.

Para el presente proyecto se ha creado una cuenta de *Gmail* con la dirección de correo electrónico *iiot.home.notification@gmail.com* (ver **Figura 17**), y se está utilizando la librería **smtplib** de Python para la conexión del bloque funcional *Notificación e-Mail* (ver **APÉNDICE C**) con la cuenta de *Gmail* creada.

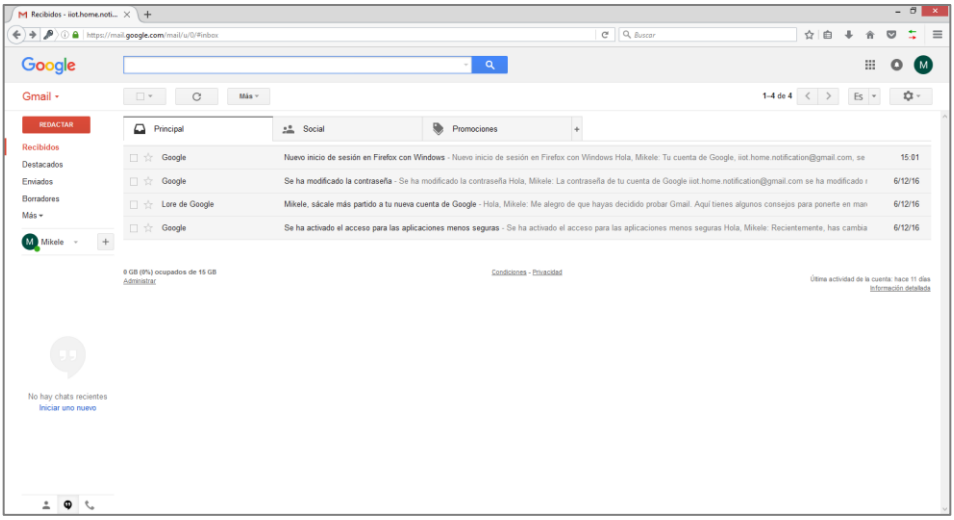


Figura 17 Sitio web de una cuenta de Gmail

Fuente: (Elaboración propia, 2017).

3.2.2. Configuración de la cuenta del servicio SMS Twilio

Twilio es una plataforma de internet que permite, entre varios servicios, enviar mensajes de texto SMS a cualquier operadora del mundo. Gracias a su API REST, es fácil y rápido de integrar a otras aplicaciones. Para el presente proyecto se lo empleara para él envió de notificaciones ante las alarmas más críticas de la red de control IzoT.

Suscribirse y obtener una cuenta en *Twilio* es fácil y gratuito. Solo basta con rellenar un formulario y autentificar una cuenta de correo electrónico accediendo

a la página *https://www.twilio.com*. Una vez dado de alta se podrá entrar a una consola dashboard vía web (ver **Figura 18**) desde donde es posible acceder a todos los servicios ofrecidos por *Twilio*.

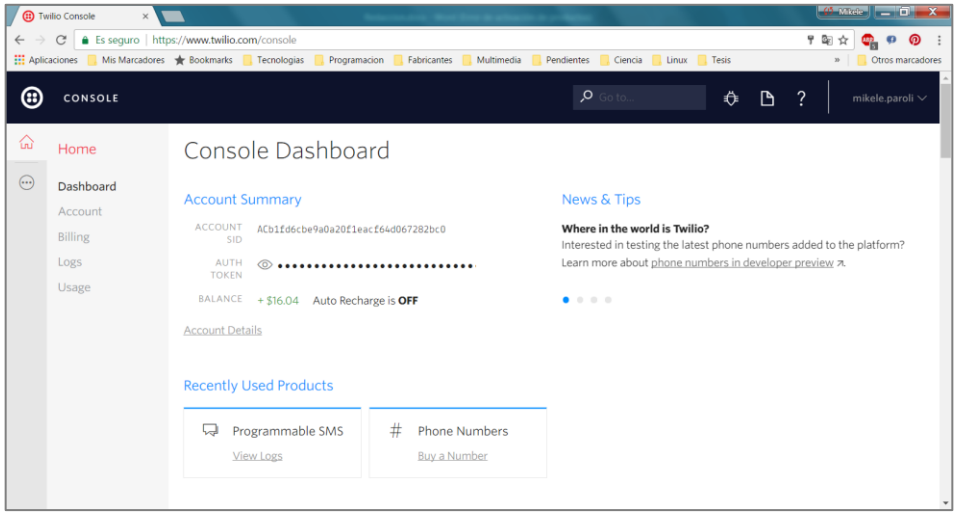


Figura 18 Consola Dashboard de Twilio

Fuente: (Elaboración propia, 2017).

Twilio permite el alquiler de una línea telefónica desde la misma consola web (ver **Figura 19**), dicho línea corresponde a una operadora de los Estados Unidos, y desde la misma consola es posible hacer recargas con una tarjeta de crédito o a través de **Paypal**.

Para más información es posible acceder a la documentación completa del API de *Twilio* desde *https://www.twilio.com/docs/api/*.

3.2.3. Configuración de la cuenta del servicio de datos ThingSpeak

ThingSpeak es una plataforma para el Internet de las Cosas, especialmente diseñado para almacenar datos en la nube de sensores o dispositivos. Contiene

un API REST que permite integrar la aplicación de los dispositivos de una forma fácil y rápida.

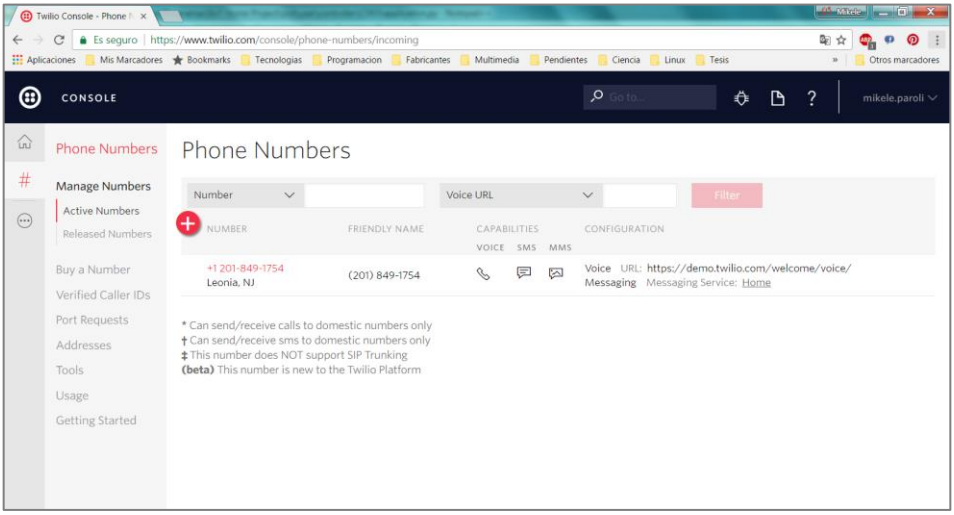


Figura 19 Compra de número telefónico en Twilio

Fuente: (Elaboración propia, 2017).

Obtener una cuenta en *ThingSpeak* es fácil y gratuito, solo con llenar un formulario desde *https://thingspeak.com/* ya se puede tener una cuenta y espacio ilimitado para adicionar dispositivos y almacenar datos.

El concepto de *ThingSpeak* es el de crear canales, “channel”, cada canal contiene campos, “field”, que son los asociados a los dispositivos o medidas particularmente. Un campo almacena la data de un dispositivo. Para el presente proyecto se creó el canal *Home IoT Application* dentro de la cuenta personal del autor de este proyecto y se crearon los campos que se exponen en la **Tabla 20**. El sitio web del canal de *ThingSpeak* proporciona de dashboard que permiten monitorear en tiempo real los datos de cada campo (ver **Figura 20**).

Tabla 20

Campos creados en canal Home IoT Application

No.	Campo (Field)	Descripción
1	RoomTemperature	Temperatura de estancia.
2	RoomRH	Humedad relativa de estancia.
3	RoomLighting	Luminosidad de estancia.
4	ElecCurrentPhaseA	Demanda de la fase A.
5	ElecCurrentPhaseB	Demanda de la fase B.
6	ElecPowerConsumed	Potencia total consumida.

Fuente: (Elaboración propia, 2017).

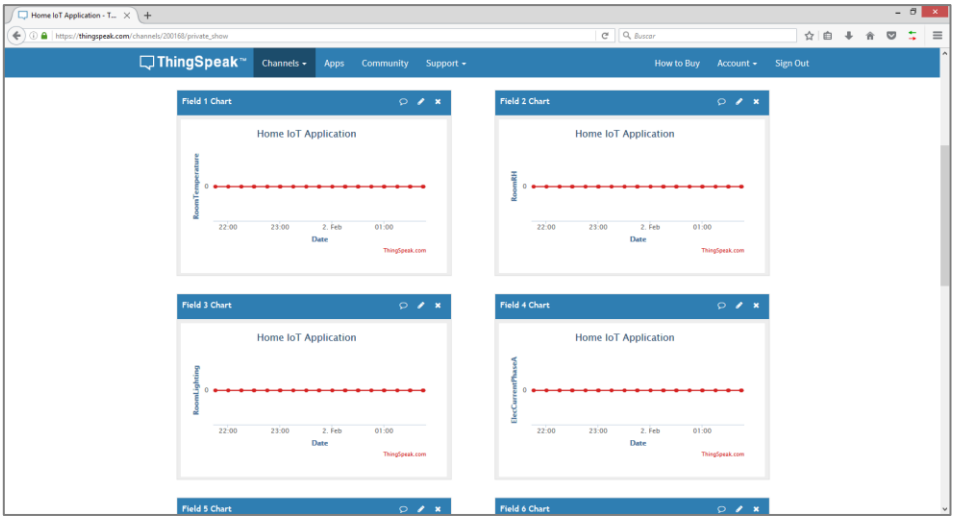


Figura 20 Sitio de monitoreo del canal Home IoT Application

Fuente: (Elaboración propia, 2017).

Para poder realizar peticiones a los campos de un canal de *ThingSpeak* es necesario que el canal haya sido configurado como público y generar un API Key.

Para poder enviar o extraer datos en un canal es necesario utilizar métodos HTTP como GET y POST teniendo en cuenta el “API Key”, el “Channel ID” y el número del campo (field).

3.3. INSTALACIÓN Y CONFIGURACIÓN DEL IZOT SDK EN LOS CONTROLADORES

Para la instalación del IzoT SDK debemos tener en claro diferentes condiciones y herramientas para el trabajo con el dispositivo ODROID C2. En primer lugar, se debe instalar la imagen del sistema operativo en la tarjeta de memoria eMMC que funcionará como la memoria de programas del ODROID C2. Para esto descargamos la imagen y la instalamos con *Win32 Disk Imager* (ver **Figura 21**), una útil, sencilla y popular herramienta para *Windows*. Para el presente proyecto se va a utilizar como sistema operativo de los controladores, *Debian Wheezy* por ser el sistema operativo más estable sobre el que trabaja el IzoT SDK.

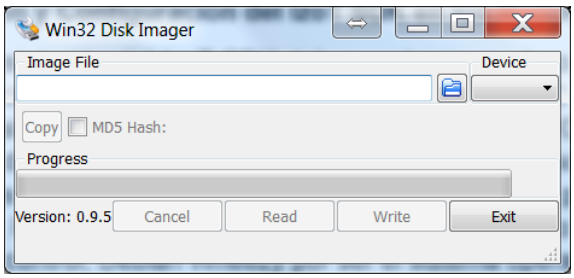


Figura 21 Herramienta Win32 Disk Imager para el montaje de imágenes

Fuente: (Elaboración propia, 2017).

La primera vez que se ejecuta un controlador se debe verificar su dirección IPv4 desde el router ya que la interfaz de comunicación por defecto está trabajando con direccionamiento dinámico DHCP.

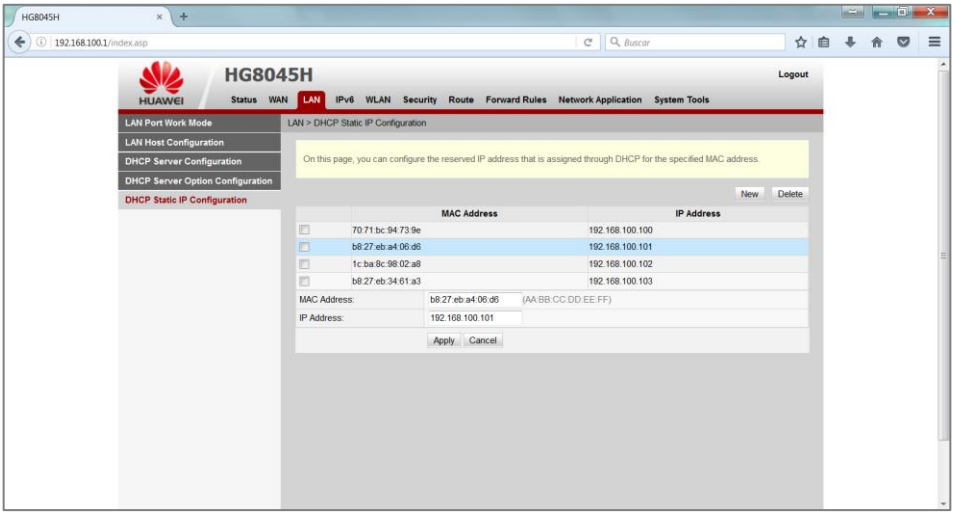


Figura 22 Interfaz Web de configuración del router, lista de clientes DHCP

Fuente: (Elaboración propia, 2017).

En la lista de clientes DHCP del router podemos encontrar un controlador (ver **Figura 22**), para este caso se va a configurar el dispositivo *iot-device-1x* el cual corresponde al controlador de la cocina (*iot-devive-1x* es el nombre de equipo o hostname del SBC, mismo que fue pre-configurado). Es posible desde el router reservar la dirección IP del dispositivo a través de la dirección física. La **Tabla 21** muestra los diferentes controladores a ser configurados, junto a su nombre de equipo y dirección de red designados.

Para la conexión y administración de los controladores se utilizaron dos herramientas clientes SSH y SFTP para la conexión remota por consola y gestor de archivos respectivamente. Por defecto la imagen del sistema operativo

instalado viene con servicios SSH y SFTP levantados. Para la conexión por SSH se utilizó la herramienta libre *PuTTy* (ver **Figura 23**) y para la conexión por SFTP se utilizó *WinSCP* (ver **Figura 24**). En ambos casos únicamente es requerido la dirección de red, el puerto, el usuario y la contraseña del nodo para su conexión. En todos los dispositivos el usuario y contraseña por defecto es “debian” y “admin” respectivamente.

Tabla 21

Asignaciones de hostname y direcciones

No.	Controlador	Hostname	Dirección de Red
1	CD01 (Cocina)	iot-device-1x	192.168.200.101
2	CD02 (Sala-Comedor)	iot-device-2x	192.168.200.102
3	CD03 (Dormitorio-Baño)	iot-device-3x	192.168.200.103

Fuente: (Elaboración propia, 2017).

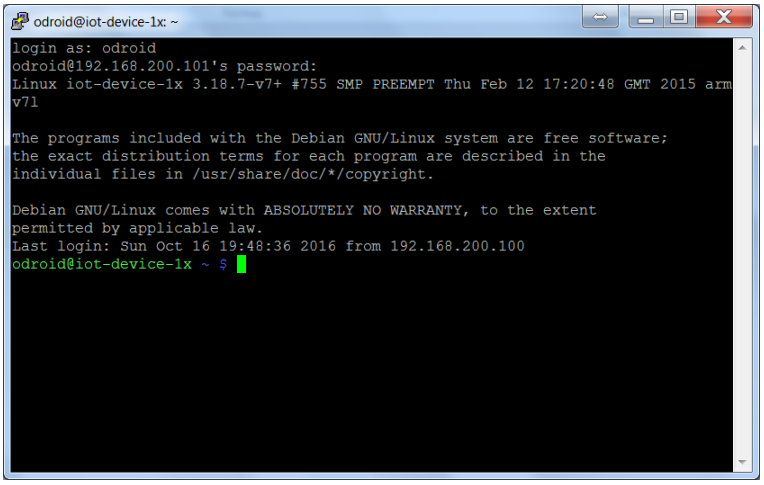


Figura 23 Consola de PuTTy, conexión SSH con un Controlador

Fuente: (Elaboración propia, 2017).

El primer paso para la instalación del IzoT SDK es actualizar el sistema operativo de los SBC, para lo cual se debe ejecutar los comandos **sudo apt-get update** y **sudo apt-get upgrade**. El siguiente paso es transferir el fichero comprimido del IzoT SDK al directorio de usuario. El fichero comprimido del IzoT SDK se lo puede descargar desde el sitio web de **Echelon** de forma gratuita y se lo puede transferir al ODROID C2 a través de *WinSCP* (ver **Figura 24**).

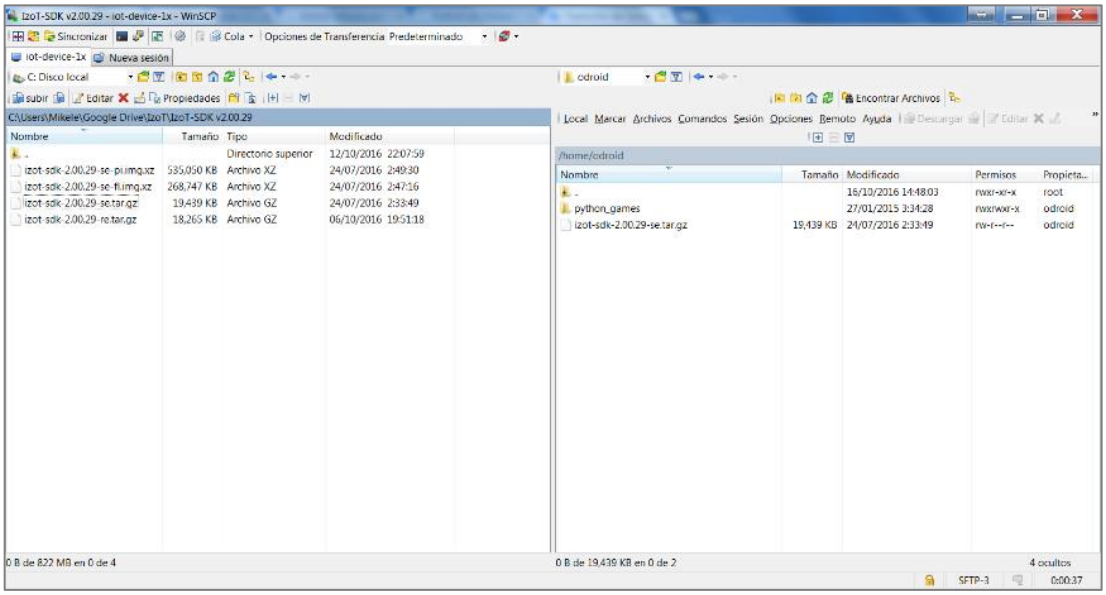


Figura 24 Interfaz de WinSCP para transferir archivos por SFTP

Fuente: (Elaboración propia, 2017).

Una vez transferido el IzoT SDK, se lo puede instalar a través de la consola remota. Se descomprime el fichero a través del comando **tar –xf izot-sdk-2.00.29-se.tar.gz** y se lo instala con **izot-sdk-2.00.29-se/install**. La instalación puede tardar varios minutos y durante el proceso se descargan dependencias, se copian los directorios y ficheros del Entorno Virtual e IzoT, como también se instalan el Supervisor y el Servidor IzoT con sus diferentes servicios.

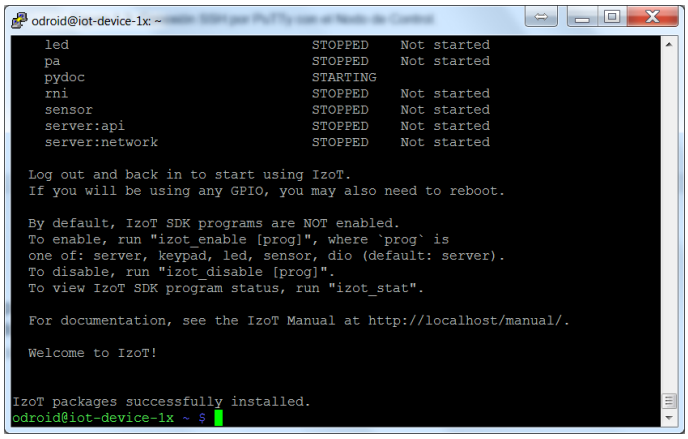


Figura 25 Finalización de la instalación del IzoT SDK

Fuente: (Elaboración propia, 2017).

Tras la finalización de la instalación (ver **Figura 25**) es necesario reiniciar la sesión para que el Entorno Virtual de IzoT empiece a ejecutarse. Esto se puede comprobar si una vez que se inicie la sesión, debe aparecer al inicio del nombre de usuario de la sesión (en este caso **debian@iot-device-1x**) el nombre del Entorno Virtual entre paréntesis **(izot-sdk)** como se aprecia en la **Figura 26.**

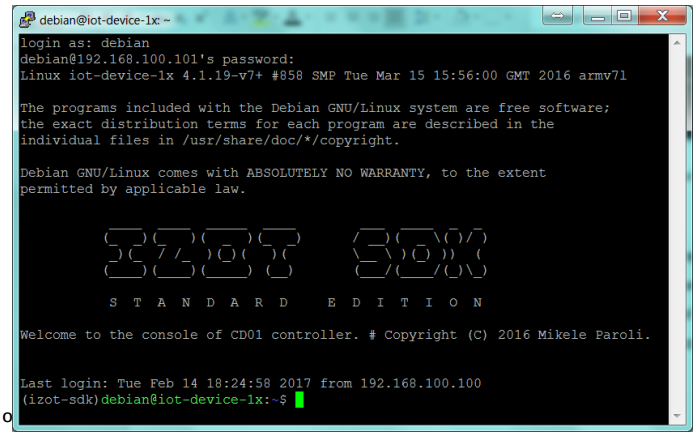


Figura 26 Inicio de sesión con el Entorno Virtual de IzoT

Fuente: (Elaboración propia, 2017).

Por último, es posible comprobar que aplicaciones se encuentran instaladas en el IzoT SDK por medio del comando **izot_stat** (ver **Figura 27**)**.**

El procedimiento descrito en esta sección es aplicable de igual forma al resto de controladores.

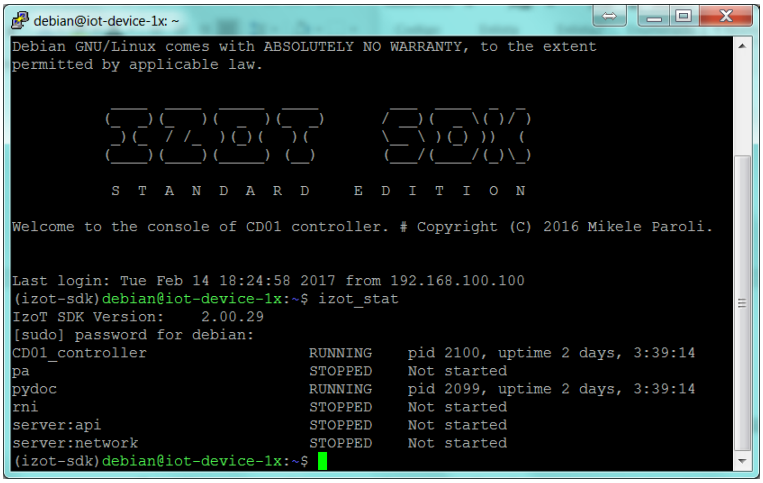


Figura 27 Ejecución del comando izot_stat

Fuente: (Elaboración propia, 2017).

3.4. PROGRAMACIÓN E INSTALACIÓN DEL FIRMWARE DE LOS CONTROLADORES

El firmware de un controlador IzoT está conformado por clases y módulos desarrollados en Python, que construyen el modelo de los datos y las funciones de la aplicación del controlador. Básicamente el firmware está constituido de los perfiles funcionales (que dan forma a los bloques funcionales), los programas funcionales y la aplicación o programa principal del controlador (donde se instancian los bloques funcionales y los programas funcionales). Los firmwares de los controladores IzoT están desarrollados en Python y se instalan dentro de

los directorios del SDK de IzoT y se ejecutan en segundo plano desde el Entorno Virtual de Python como un servicio o demonio de Linux. Una vez en ejecución el firmware de un controlador, este podrá ser encontrado por una herramienta de instalación de redes LonTalk/IP y comisionado dentro de una red de control IzoT.

En el **ANEXO B** se puede apreciar los directorios de programas que comprenden el IzoT SDK. Se ha dispuesto del directorio o ruta */izot/user/* para la instalación de los diferentes componentes que conforman el firmware del controlador, y desde el cual se configura, copila y ejecuta. En la **Tabla 22** se presenta una descripción del modelo de directorios construido para la instalación del firmware del controlador

3.4.1. Desarrollo de los perfiles funcionales

En la *Sección 2.4.1. Modelado de los bloques funcionales* se describo un modelado de los diferentes perfiles funcionales relacionados con las funciones y automatismos esperados en la implementación del sistema de control integrado con tecnología IzoT.

Cada uno de estos perfiles funcionales fueron desarrollados como clases heredadas de la clase **izot.resources.base.Profile** que se encuentra dentro del SDK de IzoT. La clase **izot.resources.base.Profile** es una estructura base para la construcción de los bloques funcionales dentro del protocolo LonTalk/IP. Para la estructuración de un perfil funcional es necesario insertar en la lista datapoints, instancias de la clase **izot.resources.base.Profile.DatapointMember** con una declaración de cada datapoint o variable del perfil funcional.

Dentro del **APÉNDICE F** se puede encontrar el código de cada perfil funcional que se ha instalado en el directorio */izot/user/resources/profile/.*

Tabla 22

Directorios del firmware de los controladores

Directorios		Descripción	
/izot/user	**/common**	Contiene las clases que construyen los métodos que facilitan la conexión del programa principal del controlador con el resto de capas del protocolo LonTalk/IP.	
		/framework	Contiene los módulos framework.py e isi_framework.py que construyen el framework para la implementación de la aplicación IzoT.
		/util	Contiene un módulo util.py con clases y métodos utilizados por el framework.
	/controller	Contiene el los programas principales de cada uno de los controladores.	
		/CD01	Contiene el módulo application.py con el programa principal del controlador CD01.
		/CD02	Contiene el módulo application.py con el programa principal del controlador CD02.
		/CD03	Contiene el módulo application.py con el programa principal del controlador CD03.
	/resources	Contiene los recursos y módulos que conforman el firmware de los controladores.	
		/profile	Contiene las clases con la declaración de los perfiles funcionales modelados y desarrollados para el presente proyecto.
		/program	Contiene las clases con los programas funcionales desarrollados para los bloques funcionales del presente proyecto.
		/util	Contiene módulos con funciones utilizados por los programas funcionales.

Fuente: (Elaboración propia, 2017).

3.4.2. Desarrollo de los programas funcionales

En la *Sección 2.4.2. Diseño de los programas funcionales* se describe el diseño de cada uno de los programas funcionales que acompañan a cada bloque funcional modelado. El diseño de los programas funcionales se lo realizo a través de diagramas de flujos, los cuales fueron desarrollados como clases en Python, para más tardes ser instanciados y ejecutados en el programa principal de cada controlador.

Todas las clases poseen atributos de inicialización que son instanciados en el constructor de la clase. Los atributos de inicialización están constituidos por: el bloque funcional asociado, una clase interfaz I/O (si el programa funcional corresponde a un periférico), y constantes como umbrales o tiempos. Cada clase posee un único método llamado **run()** donde se construye el programa funcional, tal cual se especifica en los diagramas de flujo presentados en el **APÉNDICE D** de este documento. En el **APÉNDICE F** se presenta el código de cada clase que conforma cada uno de los programas funcionales.

Un parte muy importante del desarrollo de los programas funcionales son las utilidades que se encuentran instaladas en el directorio */izot/user/resources/util/*. Estas utilidades contienen clases y métodos que permiten: la lectura y mando de las interfaces de I/O a través del bus I2C o los GPIOs de los SBCs, la conexión con las APIs de los Servicios de Internet como *Gmail*, *Twilio* y *ThingSpeak*, y la función especial de temporizador no bloqueante. Los códigos de estas utilidades pueden ser encontrados de igual forma en el **APÉNDICE F**.

3.4.3. Desarrollo de los programas principales de los controladores

El programa o aplicación principal de un controlador es aquel que se ejecuta continuamente dentro del entorno virtual del SDK. Cuando este inicializa

construye la estructura del dispositivo IzoT como datapoints, bloques funcionales y programas funcionales. Dentro de la aplicación principal se realiza una instancia del Framework común que contiene todo los métodos y lógicas para copilar al controlador como un dispositivo LonTalk/IP. De esta manera únicamente es necesario, desde la aplicación, llamar al método **app.block()** para construir y suscribir un bloque funcional, **app_start()** para arrancar la aplicación, y **app.service()** para refrescar la aplicación dentro del lazo principal.

El código completo de cada uno de las aplicaciones principales de cada controlador y el código del Framework se puede encontrar en el **APÉNDICE F** de este documento. Cada programa principal está desarrollado con el número de bloques funcionales que se describe en la *Sección 2.4.3. Distribución de los servicios de control.*

3.4.4. Puesta en servicio de los Firmwares

Para poner en servicio un firmware de un controlador es necesario elaborar un fichero de configuración, el cual es utilizado por el supervisor del entorno virtual para encontrar las rutas de las dependencias del dispositivo, como la ruta de la aplicación principal. El fichero de configuración contiene, además, el nombre de presentación del dispositivo en el Entorno Virtual, variables de configuración de la ejecución de la aplicación y el directorio de almacenamiento del fichero **log** (registros) de eventos de la aplicación. El fichero de configuración debe ser guardado o instalado en el directorio */etc/supervisor/* y deben contener la extensión **conf**. En el **APÉNDICE F** es posible encontrar los ficheros de configuración de cada uno de los controladores.

Una vez que los ficheros de configuración se encuentren instalados es posible poner en servicio el programa del dispositivo únicamente utilizando los comandos **izot_start {nombre del dispositivo}** o **izot_enable {nombre del dispositivo}**.

En la **Figura 28** se muestra la puesta en servicio del controlador CD01, y su verificación con el comando **izot_stat**:

Luego de que el programa del controlador se haya ejecutado y se mantenga corriendo es posible revisar el **log** del supervisor. Este log puede ser muy util para detectar errores o excepciones de programa cuando la ejecución y puesta en servicio hayan fallado.

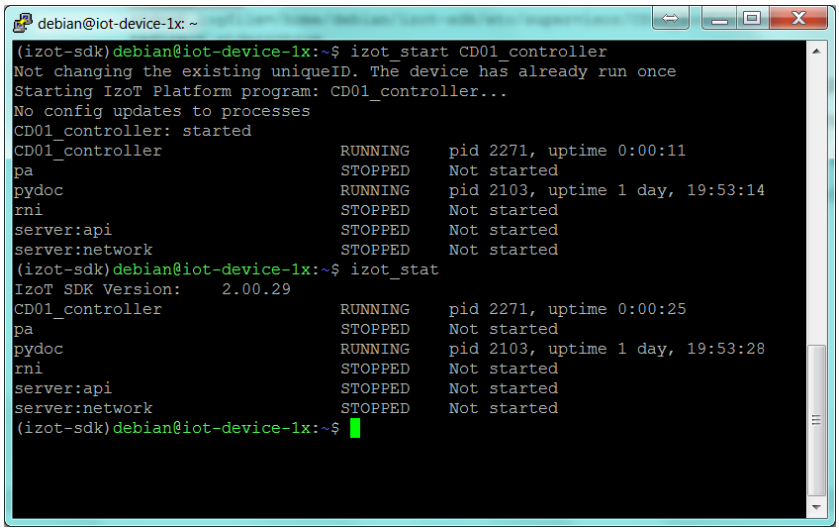


Figura 28 Puesta en servicio del controlador CD01

Fuente: (Elaboración propia, 2017).

Una parte importante de este registro de inicialización del dispositivo, es el "Unique ID" que se le ha asignado al controlador. Este identificador es utilizado más adelante para comisionar el controlador en la red de control IzoT. El "Unique ID" permite que la herramienta de comisionado encuentre el controlador en la red o canal TCP/IP. Finalmente, es posible detener el programa del controlador en cualquier momento con los comandos **izot_stop {nombre del dispositivo}** o **izot_disable {nombre del dispositivo}**.

3.5. CONFIGURACIÓN Y PUESTA EN SERVICIO DEL SERVIDOR IZOT

En la *Sección 1.2.4. Servidor IzoT* se hace una descripción detallada del Servidor IzoT provisto por el SDK de IzoT. El objetivo principal del servidor IzoT es el de proveer de un API REST al cual se puede conectar una aplicación empresarial o interfaz de usuario para el control, gestión y administración de una red de control IzoT. En el **ANEXO C** de este documento se adjunta una guía completa del Servidor IzoT y de su API REST.

3.5.1. Procedimiento para la configuración y puesta en servicio

Para configurar y poner en servicio un Servidor IzoT lo primero que hay que hacer es la elaboración de las clases de los dispositivos a ser controlado. Las clases son modelos de los datapoints de un dispositivo que proporcionan al Servidor IzoT la información necesaria para conocer que datapoints contiene un dispositivo y como debe manejar sus datos. Una clase es un fichero XML y debe ser guardado o instalado en el directorio */etc/lonbridge/clases/* del SDK. Cada una de las clases de cada uno de los controladores se puede apreciar en el **APÉNDICE F** de este documento.

Una vez elaboradas las clases se debe proceder con la declaración del fichero *device.xml* que se registra en el directorio */etc/lonbridge/instances/* del SDK. Este fichero contiene la declaración de los dispositivos que el Servidor IzoT debe inscribir dentro del API REST. A continuación, el código del fichero *device.xml* para el presente proyecto:

```
<devices>
	<device nid="FE375ADCF59D" id="0" fblock="-1" type="unknown" name="Unknown
	Device" brand="IzoT" active="false" />
	<device nid="FE04184D5EC2" id="1" fblock="-1" type="unknown" name="Unknown
	Device" brand="IzoT" active="false" />
	<device nid="FECB2F2E2AA3" id="2" fblock="-1" type="unknown" name="Unknown
	Device" brand="IzoT" active="false" />
</devices>
```

La parte principal de esta declaración es el atributo “nid” de cada dispositivo. El “nid” corresponde al “Unique ID” del controlador que se puede extraer del **log** del supervisor de la aplicación del mismo como de explica en la *Sección 3.4.4. Puesta en servicio de los Firmwares*. Este identificador será el utilizado por el Servidor IzoT para encontrar el dispositivo en la red.

Una vez que el Servidor IzoT encuentra un dispositivo en una red IzoT, consulta su PID, y con este busca una clase que coincida con dicho PID, procesa su modelo de datos, y lo inscribe y construye en el API REST del SDK. Todo esto, cuando se levanta el Servidor IzoT a través del comando **izot_start** (ver **Figura 29**).

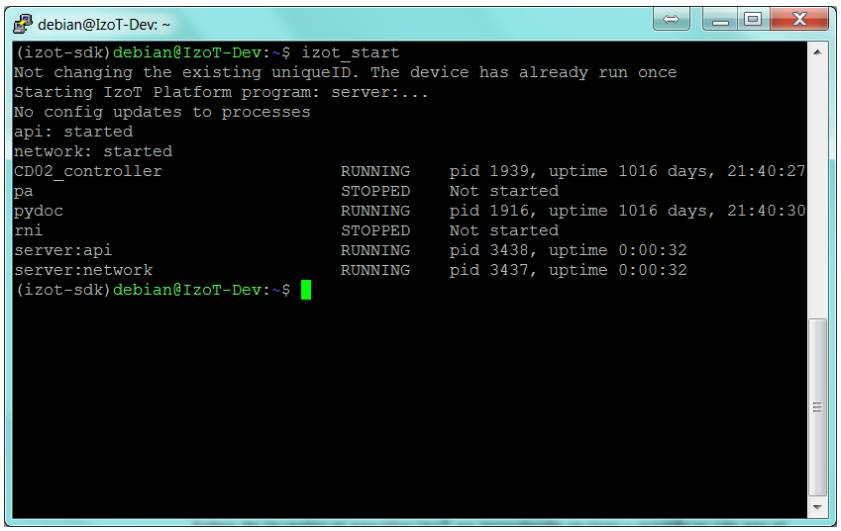


Figura 29 Puesta en servicio del Servidor IzoT en el controlador CD02

Fuente: (Elaboración propia, 2017).

Es posible detener o reiniciar el Servidor IzoT con los comandos **izot_stop** e **izot_restart** respectivamente.

Antes de levantar el Servidor IzoT es importante revisar y modificar (de ser el caso) el fichero *config.xml* en el directorio */etc/lonbridge/*. A continuación, el código de este fichero:

```
<bridge>
    <id value="lon1" />
    <interface value="default" />
    <ip value="127.0.0.1" />
    <port value="3050" />
    <pollrate value="0.1" />
    <logging value="general" />

    <comment value="The default to be that the server is managed" />
    <comment value="by the IzoT Commissioning Tool." />
    <comment value="The domain Id is 3 hex (length 1) and the subnet/node Id" />
    <comment value="is 1/50 for the Wi-Fi/Ethernet channel which default to use" />
    <comment value="IP address 10.3.1.1" />
    <comment value="Refer to the IzoT SDK User's Guide for detailed information." />
    <isi value="disable" />
    <subnetnodeid value="1/50" />
    <domainid value="1, 3 0 0" />
</bridge>
```

Los atributos a tener en cuenta del fichero *config.xml* son “subnetnodeid” y “domainid”. Estos atributos están relacionados con la red de control a ser programada, el “subnetnodeid” será el identificador de nodo del Servidor IzoT (para este proyecto, 50) en la red LonTalk/IP y no lo puede contener otro dispositivo en la red, mientras que el “domainid” contiene el identificador de dominio de la red LonTalk/IP a ser programada y donde serán comisionados los controladores (para este proyecto el ID de dominio será 03). Esto será ampliado en la *Sección 3.6. Puesta en servicio de la red de control IzoT.*

A pesar que el SDK de IzoT instala en todos los dispositivos el Servidor IzoT, es únicamente suficiente configurar y levantar el Servidor IzoT en un solo dispositivo o controlador (en este caso). Para el presente proyecto se ha designado al controlador CD02 de la sala-comedor como el portador del Servidor IzoT y al cual se conectará la Aplicación Cliente a ser desarrollada dentro de este capítulo.

3.5.2. Recursos Del API REST Del Servidor IzoT

Una vez puesto en servicio al Servidor IzoT, es posible conectarse al API REST, esto a través de sus recursos a los cuales es posible acceder a través de las direcciones o URL del controlador CD02 (ver **Tabla 23**).

Una guía completa de los recursos del API REST se los puede encontrar en el **ANEXO C** de este documento.

Tabla 23

Recursos del API REST del Servidor IzoT

No.	Dirección/Recurso	Descripción
1	http://192.168.100.102/api/users/{id}/	A través de este recurso es posible administrar los usuarios del API REST. Puede administrarse todos los usuarios o uno en específico ingresando el id del respectivo usuario en un nivel más del recurso.
2	http://192.168.100.102/api/devices/{id}/	A través de este recurso es posible gestionar los dispositivos inscritos en el API REST. Puede gestionarse todos los dispositivos o uno en específico ingresando el id del respectivo dispositivo en un nivel más del recurso.
3	http://192.168.100.102/api/datapoints/{id}/	A través de este recurso es posible controlar los datapoints inscritos en el API REST. Puede controlarse todos los datapoints o uno en específico ingresando el id del respectivo datapoint en un nivel más del recurso.

Fuente: (Elaboración propia, 2017).

Es posible consultar o enviar datos al API REST de tres formas distintas: WEB, JSON y XML. Siendo WEB la forma por defecto (ver ejemplo de la **Figura 30**).

De esta manera cuando se accede a un recurso del API REST, el servidor devolverá una página HTML con la información solicitada.

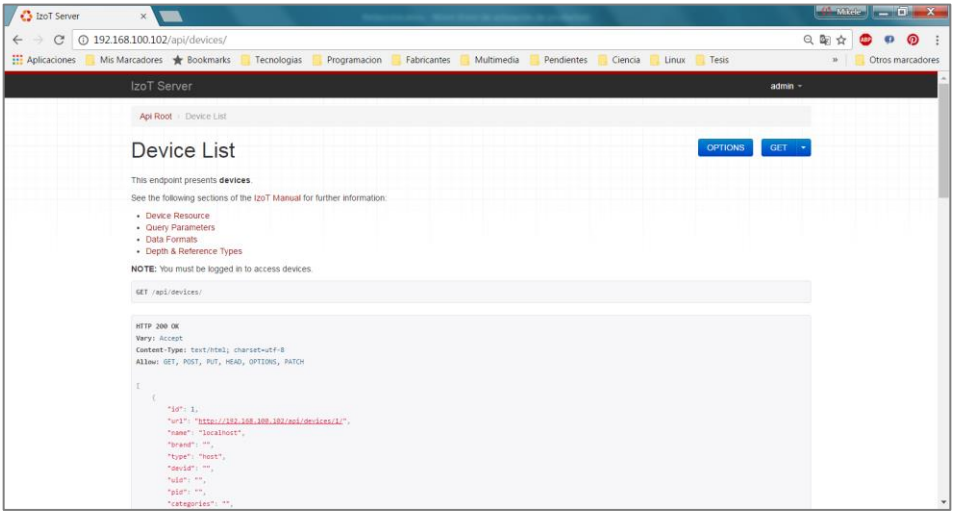


Figura 30 Acceso al recurso /api/devices/ del API REST por WEB

Fuente: (Elaboración propia, 2017).

Para recibir o enviar datos en otros formatos debe ser especificado en el atributo format (Ej. *http//:192.168.100.102/api/device/?format=json*).

3.5.2.1. Administración de usuarios del API REST

Para poder manejar cualquier recurso del API REST es necesario estar inicializado como un usuario valido anteriormente registrado. Esto como medida de seguridad. Cada que se solicita o envía un dato, este debe estar acompañado del nombre de usuario (username) y contraseña (password) de un usuario registrado.

El API REST del Servidor IzoT proporciona un servicio de gestión de usuarios completo desde el cual es posible crear, leer, modificar y eliminar usuarios a

través de los métodos POST, GET, PUT y DELETE respectivamente. Para poder administrar la base de datos se debe acceder como un usuario administrador. Por defecto el API REST proporciona un usuario “Administrator” cuyo **username** es “admin” y **password** es “admin.

3.5.2.2. Control de datapoints desde el API REST

Cada datapoint de la red de control puede ser leído (datapoint de salida) o escrito (datapoint de entrada) con el API REST del Servidor IzoT. Esto a través de los métodos GET y PUT.

3.5.2.3. Conexión desde internet con el Servidor IzoT

Como parte fundamental de este proyecto, es el de acceder desde el exterior al API REST del Servidor IzoT para el control de la red desde una aplicación cliente, para esto es requerido un dominio único de Internet que apunte a la IP publica del router de frontera y permita una conexión constante.

Lamentablemente los proveedores de internet manejan IPs dinámicas, haciendo que se vuelva complicado que un dominio apunte a la IP designada por el proveedor de internet. Una solución a este problema son los servicios de dominio dinámico como **DynDNS** (utilizado para este proyecto). Por una cuota mínima, **DynDNS** provee del servicio de rastreo de IP y dominio único.

Hoy en día, la mayoría de routers dan soporte para servicio de dominio dinámico, este consiste en insertar las credenciales correspondientes del dominio dinámico contratado en la página de administración del router (ver **Figura 31**), y a su vez, este se encarga de transmitir al servidor cada cierto periodo de tiempo la IP publica con la que cuenta. Para este proyecto se utilizará el dominio *iiot.dyndns-home.com*, contratado a **DynDNS**.

Por último, es importante redireccionar el puerto 80 a la dirección del controlador que ejecuta el Servidor IzoT (ver **Figura 31**).

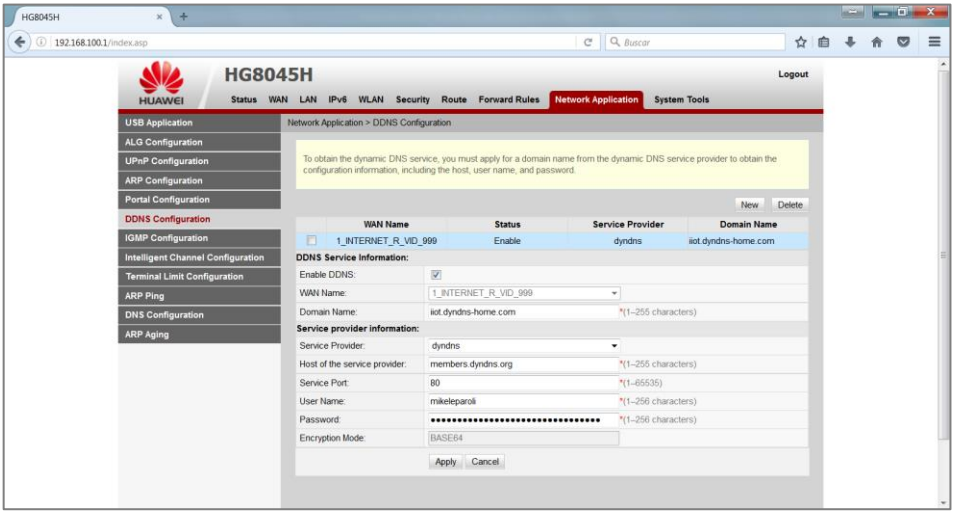


Figura 31 Configuración del dominio dinámico en el router

Fuente: (Elaboración propia, 2017).

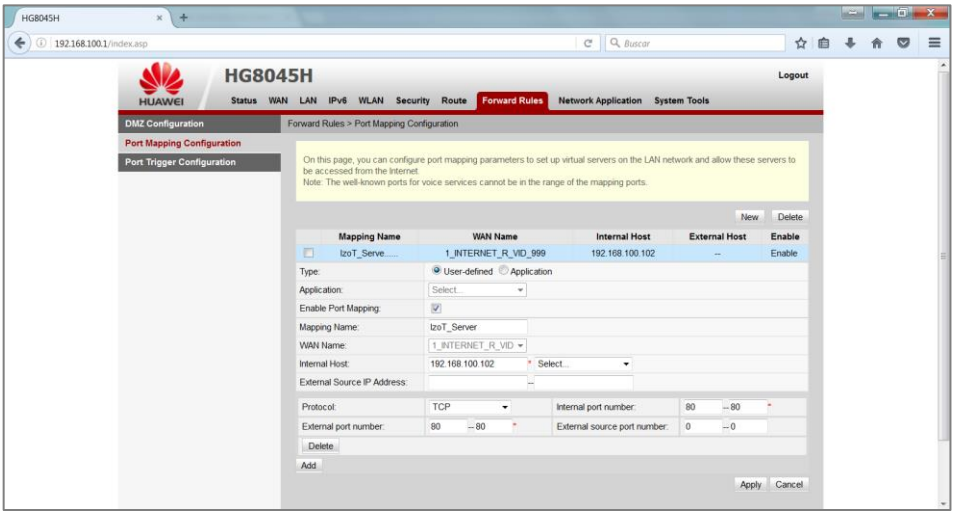


Figura 32 Redireccionamiento del puerto 80 a la IP 192.168.100.102

Fuente: (Elaboración propia, 2017).

3.6. PUESTA EN SERVICIO DE LA RED DE CONTROL IZOT

Hasta este punto se tiene la instalación física de la Red de Control Distribuida, como también la instalación del Firmware de cada controlador. Como parte final, para la instalación de la Red de Control IzoT y su puesta en servicio, fue necesario crear una red virtual LonTalk/IP, comisionar cada controlador dentro de dicha red virtual, y realizar el conexionado virtual de los diferentes datapoints para conseguir todos los automatismos y funcionalidades, objetos de este proyecto.

3.6.1. Procedimiento para la creación de una red LonTalk/IP

Con el motivo de realizar los trabajos de programación de la red LonTalk/IP, se utilizó la herramienta provista por **Echelon**, *OpenLNS Commissioning Tool*. Este trabaja como un plug-in de *Microsoft Visio 2010*, que permite programar la red de forma gráfica, aprovechando la potencia de *Visio* para realizar diagramas.

Para que el *OpenLNS Commissioning Tool* se pueda conectar con la red de control, requiere de una interfaz virtual LonTalk/IP. Esta interfaz se ejecuta como un servicio desde cualquier computador y se puede crear con la herramienta *IzoT Network Services LonTalk-IP Interfaces* (ver **Figura 33**).

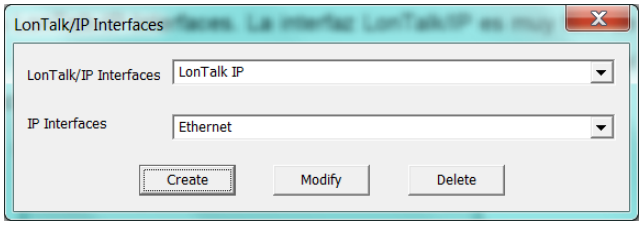


Figura 33 Panel del IzoT Network Services LonTalk-IP Interfaces

Fuente: (Elaboración propia, 2017).

La interfaz LonTalk/IP es muy fácil de crear solo basta con asignarle un nombre y escoger la tarjeta de red del computador conectada a la misma red TCP/IP de la red de control IzoT.

Cuando se ejecuta el *OpenLNS Commissioning Tool*, se abre un panel que permite crear, abrir o respaldar una red LonTalk (ver **Figura 34**). Para crear una nueva red, solo basta con asignar un nombre y presionar sobre “Create Network”. Para el presente proyecto se ha designado el nombre **IzotNetwork**.

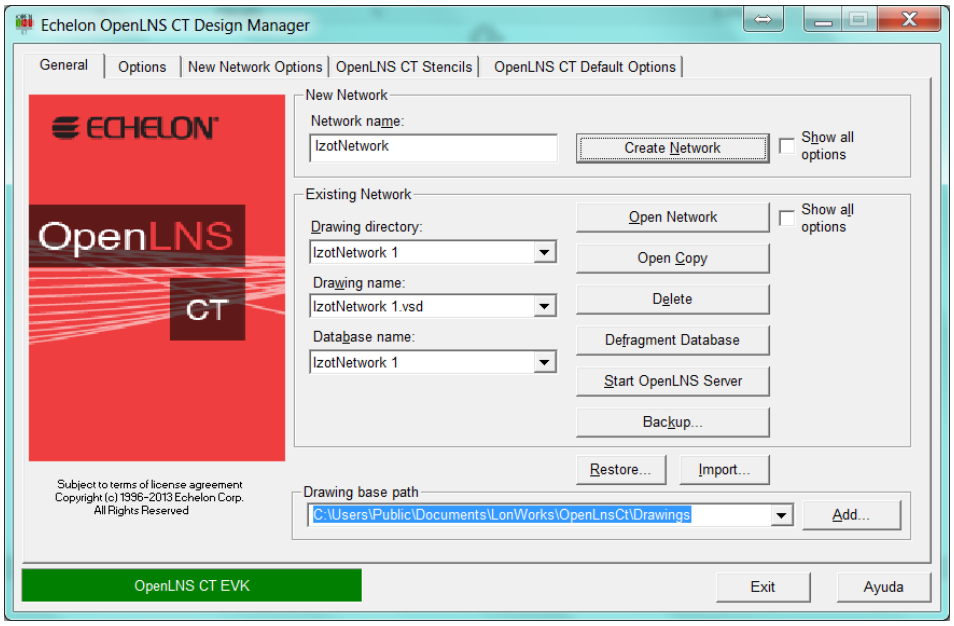


Figura 34 Panel del OpenLNS Commissioning Tool

Fuente: (Elaboración propia, 2017).

Mientras se crea la red, se abre un proyecto de *Microsoft Visio 2010* y se solicita escoger la interfaz de red, para este caso, se seleccionó **LonTalk IP** creado en el *IzoT Network Services LonTalk-IP Interfaces*. Una vez hecho esto y luego de un proceso de construcción, se muestra una hoja de *Visio* con la interfaz virtual y el canal de comunicaciones LonTalk/IP (ver **Figura 35**). Sobre el cual es posible

agregar los diferentes dispositivos de la red simplemente arrastrando desde la barra izquierda de contenedores, y comisionarlos siguiendo un *Wizard* donde paso-a-paso se puede ir asignado parámetros como del nombre al dispositivo, el canal, el firmware y el NID (ver **Figura 36**). El *OpenLNS Commissioning Tool* automáticamente busca el dispositivo en la red TCP/IP y lo comisiona con los parámetros adecuados dentro de la red LonTalk/IP.

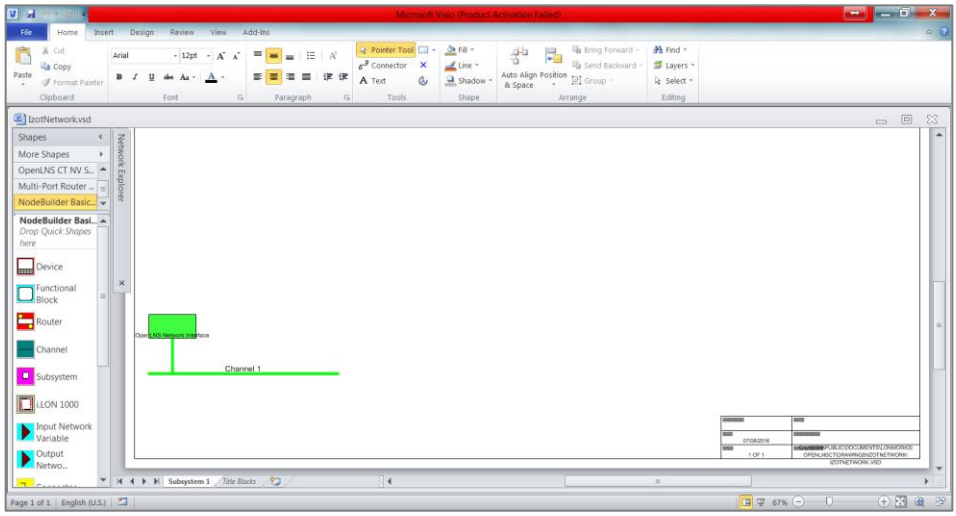


Figura 35 Red LonTalk/IP recién creada en Microsoft Visio 2010

Fuente: (Elaboración propia, 2017).

OpenLNS Commissioning Tool permite crear también subsisemas, bloques funcionales y conexiones de red de igual forma que con los dispositivos, arrastrándolos de la barra de contenedores.

3.6.2. Programación y configuración de la red LonTalk/IP

Una red LonTalk/IP, al igual que una red LonTalk clásica, permite una jerarquía de direccionamiento para simplificar el enrutamiento de paquetes, cada red

LonTalk/IP incluye una dirección o identificador de dominio, subred y nodo. Una única dirección de dominio está vinculada a toda la red LonTalk/IP, a su vez una red LonTalk/IP puede segmentarse en subredes o subsistemas cada uno con su dirección y, por último, a cada dispositivo comisionado en la red se le asigna una dirección de nodo (InterAct, 2002).

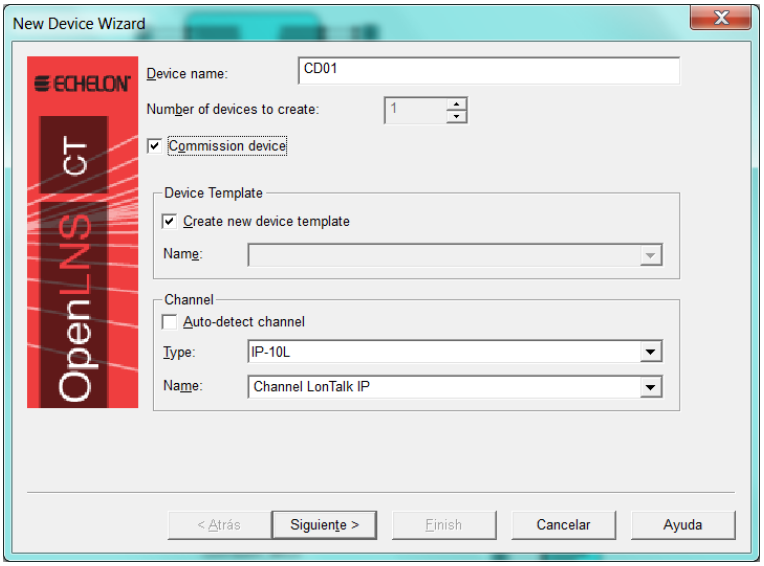


Figura 36 Wizard para la creación de un nuevo dispositivo

Fuente: (Elaboración propia, 2017).

El *OpenLNS Commissioning Tool* genera de forma automática todas estas direcciones durante la construcción de la red LonTalk/IP, aunque es posible asignar de forma manual la dirección de dominio (muy importante para que coincida con la del Servidor IzoT) desde la ventana “Network Properties” accesible desde el menú “Add-Ins”, ítem “LonWorks Network” (ver **Figura 37**).

Para el presente proyecto se ha dispuesto que la red LonTalk/IP este segmentada en tres subsistemas, cada uno asociado a un área de la vivienda (cocina, sala-comedor, dormitorio). Para esto, desde el *Visio* se puede arrastrar

un contenedor tipo **Subsystem** en el subsistema principal (denominado **Main**) y una nueva hoja de trabajo se abre con el nombre del subsistema asignado. Dentro de cada una de estas hojas, se creó y comisiono cada uno de los controladores, como también se instancio todos los bloques funcionales desarrollados en la *Sección 3.4. Programación e instalación del Firmware de los controladores*.

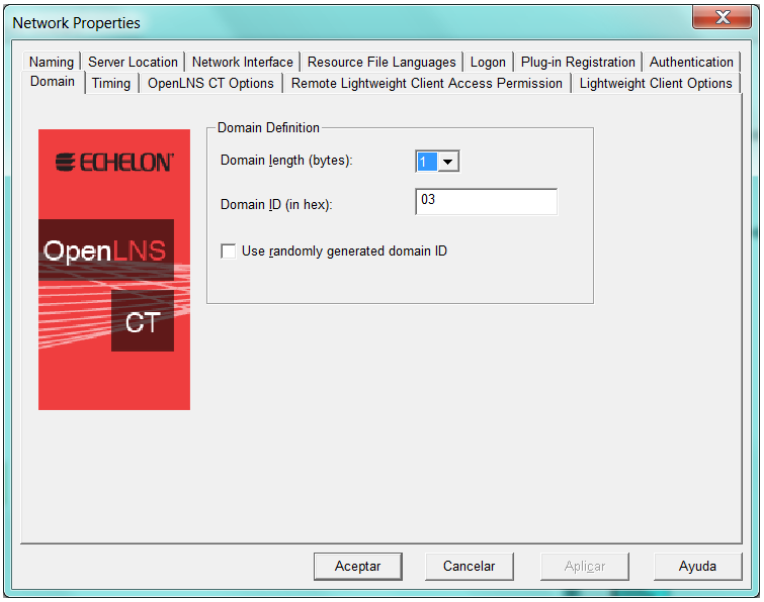


Figura 37 Panel Network Properties de la red LonTalk/IP

Fuente: (Elaboración propia, 2017).

Con los bloques funcionales en cada hoja, se procedió con la conexión de sus datapoints con el objetivo de cumplir con los trabajos descritos en la *Sección 2.2. Definición del sistema de control integrado.* La conexión de los datapoints se la puede realizar a través de la ventana *Network Variable Connection* (ver **Figura 38**), accesible desde el menú contextual, haciendo click derecho sobre un bloque funcional, opción “connect…”.

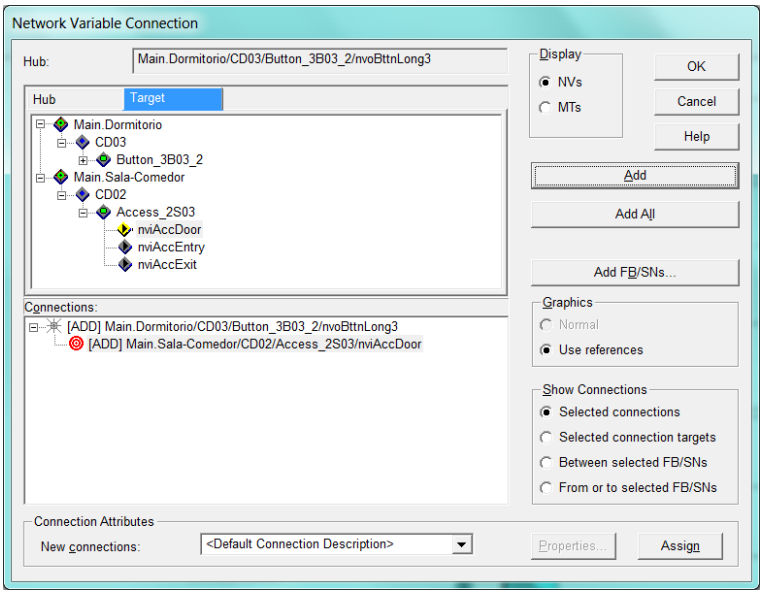


Figura 38 Panel Network Variable Connection

Fuente: (Elaboración propia, 2017).

En el **APÉNDICE G** de este documento, se muestra cada uno de los subsistemas programados para la red **IzotNetwork**.

3.7. PROGRAMACIÓN DE LA APLICACIÓN CLIENTE MULTIPLATAFORMA

Para culminar con la implementación del Sistema de Control Integrado de la vivienda, y una vez que se tiene completado la instalación y puesta en marcha de la red de control distribuida, se procedió con el desarrollo y programación de la Aplicación Cliente Multiplataforma IIoT. En la *Sección 2.5. Diseño de la aplicación cliente multiplataforma* se describe los principios, arquitectura y estructura de la Aplicación Cliente IIoT a ser desarrollada para la red de control IzoT y el servicio de internet IoT. Dentro de la arquitectura se especifica y define el modelo de capas MVC, arquitectura con la que fue desarrollada la aplicación y que permite la separación en desarrollo de los datos, la presentación y la lógica.

La aplicación cliente fue desarrollada con la tecnología de software JavaFX, para lo cual se utilizó el JDK de Java versión 8 y el entorno de desarrollo *IDE IntelliJ IDEA versión 2016.3.1* distribuido por **JetBrains** (ver **Figura 39**).

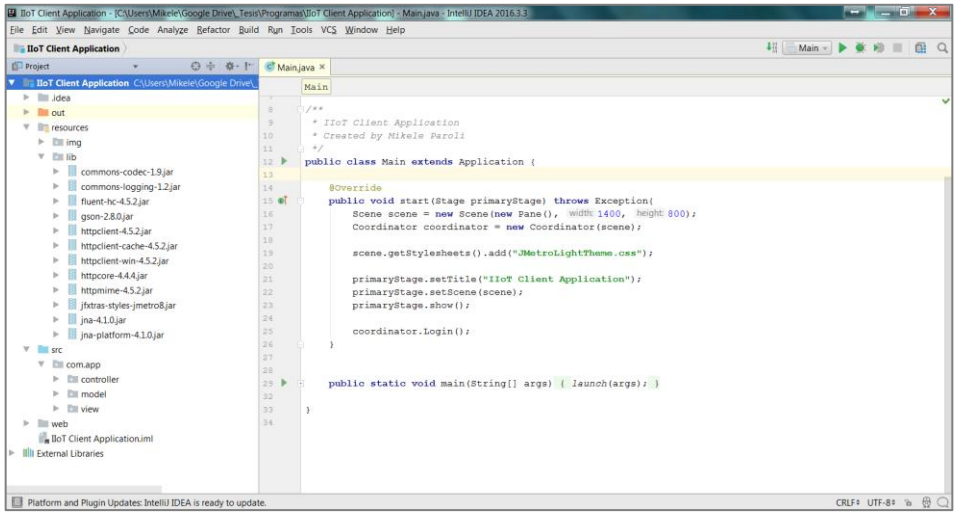


Figura 39 Interfaz del entorno de desarrollo IntelliJ IDEA

Fuente: (Elaboración propia, 2017).

IntelliJ IDEA es un potente e inteligente entorno de desarrollo para el lenguaje Java desde el cual es posible estructurar, programar y copilar un proyecto de JavaFX. *IntelliJ IDEA* es muy util a la hora de desplegar la aplicación de JavaFX, tanto en escritorio como web.

3.7.1. Desarrollo de la aplicación cliente IIoT

La aplicación fue desarrollada por capas, empezando por el modelo, seguido de la vista y terminando con el controlador. En la **Tabla 24** se puede apreciar la estructura empleada para la construcción de la aplicación.

Tabla 24

Estructura del proyecto de la aplicación cliente IIoT

Directorio raíz de la aplicación				
img	Contiene todas las imágenes e iconos utilizados dentro de la aplicación.			
lib	Contiene las dependencias y librerías de la aplicación en ficheros JAR. A continuación, una lista de dichas dependencias: • commons-codec-1.9.jar • commons-logging-1.2.jar • fluent-hc-4.5.2.jar • gson-2.8.0.jar • httpclient-4.5.2.jar • httpclient-cache-4.5.2.jar • httpclient-win-4.5.2.jar • httpcore-4.4.4.jar • httpmime-4.5.2.jar • jfxtras-styles-jmetro8.jar • jna-4.1.0.jar • jna-platform-4.1.0.jar			
com	app	Contiene todos los paquetes y clases de la aplicación. Paquete principal.		
		model	Contiene la capa modelo de la aplicación.	
			object	Contiene los Value-Object (VO).
				var: Contiene las variables de flujo de la aplicación. Las siguientes clases se implementan en este paquete: • Alarm.java • Amp.java • LevPercent.java • Lux.java • Measure.java • Occupancy.java • Power.java • Setting.java • Switch.java • Temp.java
				iiot: Contiene las variables que representan la red de control. Las siguientes clases se implementan en este paquete:

Continua

				• Block.java • Controller.java • Datapoint.java • Network.java • Node.java
			util	Contiene clases de utilidad para las variables de la red de control. Clases implementadas: • Direction.java • Profile.java • Type.java
			user	Contiene una variable que representa a un usuario de la aplicación: • User.java
			iot	Contiene variables que representan la gestión de datos del servicio IoT de ThingSpeak. Clases implementadas: • Feed.java • Field.java • Thresholds.java
		exp		Contiene las excepciones de aplicación utilizadas por el modelo. Las siguientes clases se implementan en este paquete: • AccessControlException.java • ResultException.java • StatusCodeException.java • ValidationException.java
		conn		Contiene la clase que permite conectar con un API REST a través de métodos http. • Connection.java
		access		Incluye las clases que implementan el control de acceso para el manejo de la red de control y el administrador de usuarios del Servidor IzoT, como también con el servicio de datos de ThingSpeak. Contiene las siguientes clases: • IoTAccessControl.java • NetworkAccessControl.java • UserAccessControl.java

Continua

		business	Incluye las clases que contienen la “lógica de negocio”. Validan cada uno de los métodos de las clases de control de acceso. Se tiene las siguientes clases: • IoTBusinessLogic.java • NetworkBusinessLogic.java • UserBusinessLogic.java
	view	Contiene la capa vista de la aplicación.	
		base	Encierra controles personalizados que son utilizados en distintas pantallas.
			control: Contiene controles tipo iconos animados que pueden funcionar como botones, links o etiquetas. Algunos representan clases tipo Switch, Setting y Measure. Una lista de lo que contiene: • LabelIconBase.java • LabelIconLink.java • LabelIconNews.java • LabelIconSetting.java • LabelIconSwitch.java • LabelViewAlarm.java • LabelViewAlarm2.java • LabelViewMeasure.java • LabelViewSwitch.java
			report: Contiene controles tipo chart personalizados para las pantallas de reportes de estancia y eficiencia energética. Las siguientes clases se implementan en este paquete: • HistoryChart.java • ReportFilterBar.java • StatisticChart.java • StatisticPane.java • ToggleGroupBar.java • TrendChart.java
		login	Contiene una clase que implementa la pantalla de “inicio de sección”: • LoginPane.java
		control	Incluye clases que implementan las áreas de control de la aplicación como la barra de navegación, el árbol

Continua

		de la red de control, el panel de notificaciones. Una lista de las clases contenidas: • FunctionBar.java • NavigationMenu.java • NetworkTree.java • NewsView.java
	task	Contiene las clases que implementan el área de trabajo de la aplicación. A continuación, una lista de las clases: • DeniedPane.java • EnergyReportPane.java • HomeMapPane.java • RoomReportPane.java • UserAdminPane.java
controller	Contiene la capa controlador de la aplicación. En este paquete se encuentra la clase Coordinator.java que integra todas las lógicas de negocio, las vistas y los controladores, y la clase principal de la aplicación Main.java.	
	base	Incluye la clase IIoT.java que contiene una declaración de toda red de control instalada y sirve como constructor de una clase Network.java.
	query	Contiene clases para la implementación de las peticiones a los servicios externos. Implementan hilos para ejecutar las peticiones en segundo plano y no bloquear la aplicación durante una petición. Clases encontradas: • QueryService.java • RealTimeService.java
	login	Contiene el controlador de la clase LoginPane.java. Clase: • LoginController.java
	control	Contiene las clases que controlan las áreas de control de la aplicación. Cada clase de este paquete está relacionado con una clase de view.control. Clases encontradas: • FunctionController.java • NetworkController.java • NewsController.java

Continua

		task	Contiene las clases que controlan las áreas de trabajo de la aplicación. Cada clase de este paquete está relacionado con una clase de view.task. A continuación, una lista de las clases contenidas: • EnergyReportController.java • HomeMapController.java • RoomReportController.java • UserAdminController.java

Fuente: (Elaboración propia, 2017).

Todas las clases indicadas en la **Tabla 24** fueron descritas durante la etapa de diseño en la *Sección 2.5.3. Definición de la aplicación cliente.* Los códigos de las clases mencionadas se encuentran en el **APÉNDICE H** de este documento.

3.7.2. presentación de la aplicación cliente IIoT

En el **APÉNDICE I** de este documento se presenta una *Guía de Usuario* de la Aplicación Cliente IIoT. La aplicación fue implementada para funcionar como una aplicación web y poder ser usada desde cualquier parte del mundo. Para lo cual fue adquirido un dominio, alojamiento y servicio web en la nube a la empresa de internet **Infomaniak.** El nombre de dominio adquirido para este proyecto e investigación fue *iiot-home.com*.

Para poder ejecutar la aplicación desde un explorador web se generó el fichero **iiot_app.html** donde se embebió una empaquetado JAR de la aplicación. El código del fichero **iiot_app.html** se adjunta en el **APÉNDICE H** de este documento. En las **Figuras 40, 41 y 42** se muestran algunas capturas de ejemplo de la Aplicación Cliente ejecutándose como una aplicación web.

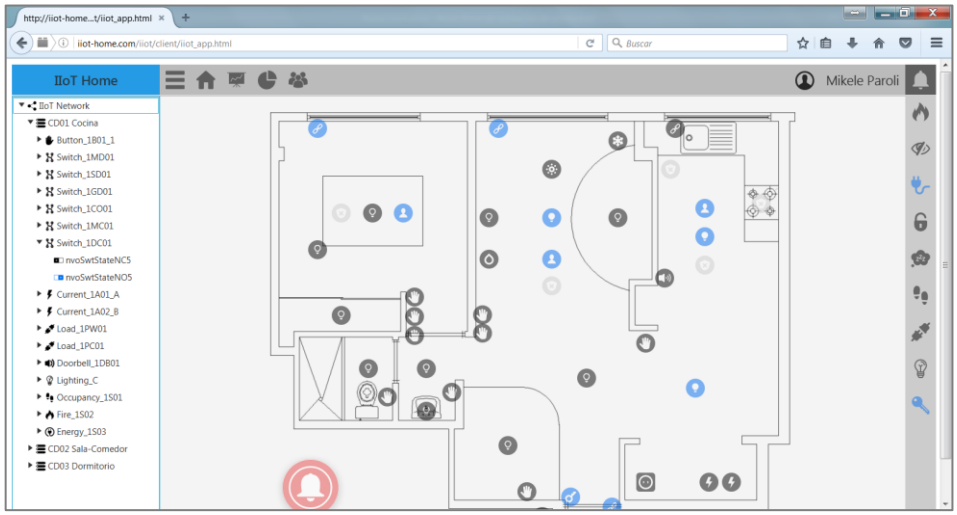


Figura 40 Aplicación Cliente IIoT desde un sitio web [1]

Fuente: (Elaboración propia, 2017).

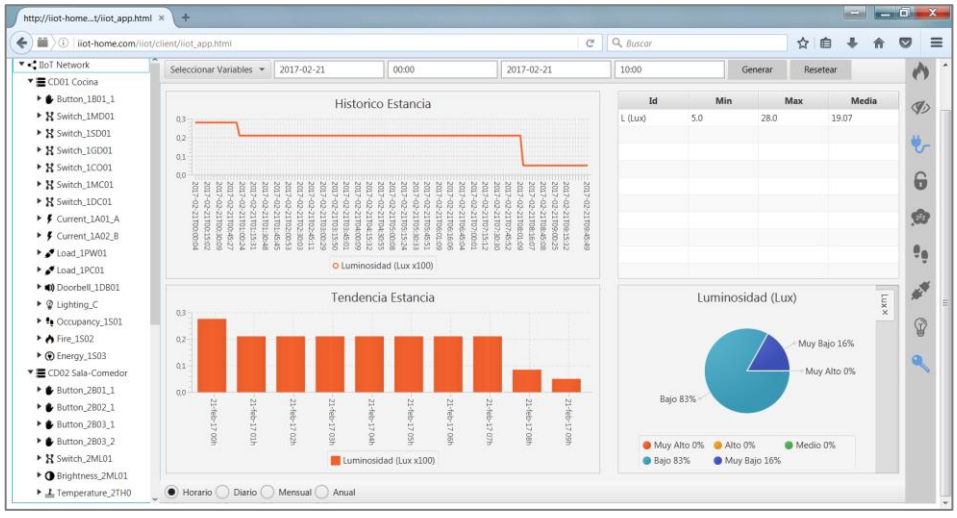


Figura 41 Aplicación Cliente IIoT desde un sitio web [2]

Fuente: (Elaboración propia, 2017).

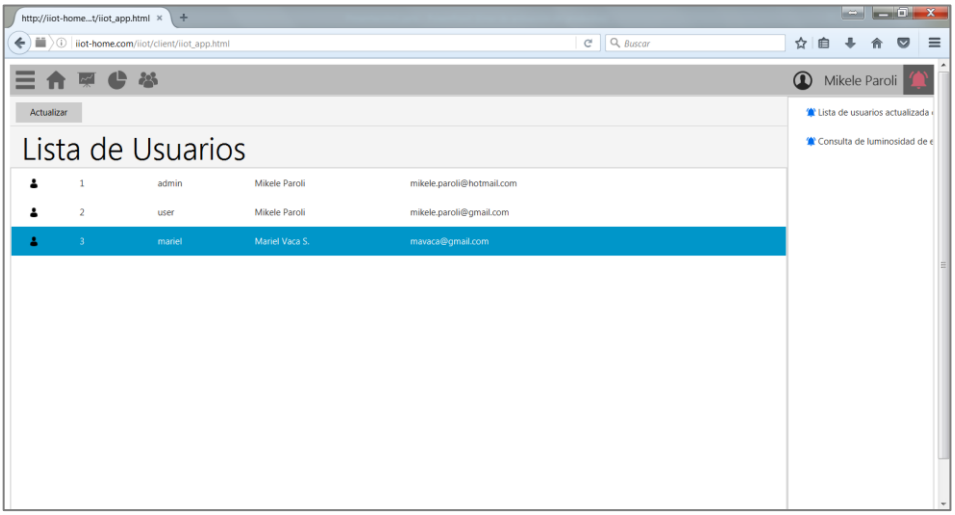


Figura 42 Aplicación Cliente IIoT desde un sitio web [3]

Fuente: (Elaboración propia, 2017).

3.8. PRUEBAS Y RESULTADOS DEL SISTEMA DE CONTROL INTEGRADO EN EL HOGAR

Una vez culminado todos los trabajos de implementación de los diferentes componentes eléctricos, electrónicos e informáticos que conforman la propuesta del Sistema de Control Integrado en el Hogar basándose en la filosofía del Internet de las Cosas Industriales, se procedió a realizar pruebas de funcionamiento con el fin de verificar lo acordado en la *Sección 2.2. Definición del sistema de control integrado.* En el **APÉNDICE J** se presenta un registro fotográfico de las pruebas realizadas con cada sistema implementado para el presente proyecto.

CAPITULO IV

CONCLUSIONES Y RECOMENDACIONES

4. CONCLUSIONES Y RECOMENDACIONES

4.1. CONCLUSIONES

Una vez terminado toda la implementación y pruebas del Sistema de Control Integrado en el Hogar es posible recopilar las siguientes conclusiones:

La tecnología IzoT demostró ser confiable y robusta, una vez puesta en servicio la red de control LonTalk/IP no se han registrado novedades o incidentes como ordenes no ejecutadas, incumplidas o retrasadas, incluso en periodos de alto consumo de datos de Internet. Se puede concluir que la red a funcionado, desde un punto de vista perceptible por el ser humano, en “tiempo real”.

Por otro lado, los SBCs ODROID C2 han demostrado un alto desempeño, no se han registrado fallos como colgamientos del sistema operativo, reinicios inesperados o faltas en la comunicación. Al igual de los módulos de interfaces de I/O, los cuales han funcionado correctamente, especialmente los módulos de salidas por relés, los cuales podrían ser los más críticos por tener componentes mecánicos.

Otra tecnología aplicada a este proyecto fue el uso de API REST para la conexión con la red de control distribuida IzoT desde el exterior o como un servicio. Esto a través de la implementación del Servidor IzoT. Dicha tecnología demostró ser bastante liviana y eficiente, como relativamente rápida, para la actualización y envió de estados y ordenes respectivamente. No se tuvo demoras

más allá de 2 segundos para conseguir resultados en la red de control, como tampoco en la lectura de estados efectuado sobre sensores y mandos. Gracias al uso de JSON, se pudo comprobar lo rápido y flexible que es manejar datos con esta clase de formato.

Se cumplió con la implementación de un Sistema de Control de Iluminación al conectar los retornos de los respectivos circuitos de iluminación con las salidas por relé de los controladores e integrándolos a la red de control IzoT a través del desarrollo del bloque funcional de tipo UFPT_lighting, que permitieron controlar el estado del retorno, como conocer su posición, desde la red LonTalk/IP.

Por otro lado, se cumplió con el Sistema de Mandos con la instalación de placas de pulsadores, su correspondiente conexión a las entradas digitales de los controladores y desarrollo del bloque funcional para la red LonTalk/IP, UFPT_button, con funciones de corta y larga pulsación retroalimentadas. La conexión entre bloque UFPT_button y UFPT_lighting representaron la primera línea de integración, permitiendo el control manual de circuitos, como es en el caso de los circuitos “b”, “d”, “g”, “h”, “i”, “j”, “m” y “n”.

Gracias a la implementación de la función de larga y corta pulsación de los bloques UFPT_button, se logró ahorrar el número de mandos de la vivienda, reduciendo de 13 interruptores a 9 pulsadores. Los circuitos de iluminación que se operan desde una larga pulsación (mayor a un segundo) son el “b”, “d”, “g”, “h”, “j” y “k” (para este circuito perteneciente al baño se dejó la pulsación corta para realizar un encendido temporizado).

Se cumplió con la implementación de un Sistema de Control de Ocupaciones, para la automatización del encendido y apagado de circuitos de iluminación en función de detectores de movimiento y medición de luminosidad. Para cumplir con esto se desarrolló el bloque UFPT_occupancy_control el cual trata las

señales de detección de movimiento en la estancia, cantidad de luz para determinar día o noche y control manual, a través de los cuales determina ocupación temporal o indefinida. Adicional trata una señal de “sleep” para desactivar el encendido automático durante las noches. Se emplearon 5 bloques de control de ocupación para el encendido automático de los circuitos “c”, “e”, “f”, “k” y “l”. Para el caso del circuito “e” el encendido automático se lo hizo a través de la señal de “luces de bienvenida” del bloque de control de acceso, y para el circuito “k” se utilizó la señal de “pulsación corta” del pulsante del baño.

De acuerdo a lo planificado se contó con un doble-sensor de movimiento y luminosidad, cuya señal de cantidad de luz se conectó con los bloques de control de ocupación para la determinación de día o noche. El umbral establecido para considerar por parte de los bloques de control de ocupación si es de día o de noche fue de 100 luxes.

Se cumplió con la implementación de un Sistema de Control de Demanda Eléctrica como parte del Sistema de Eficiencia Energética. Para esto se instaló dos sensores transductores de corriente en cada fase de la acometida de la vivienda dentro de la caja de control de la red, y conectado sus señales a las entradas analógicas del controlador CD01. Se desarrolló un bloque funcional UFPT_energy_control el cual recibe y trata las señales de los sensores de corriente para determinar el apago de circuitos de iluminación en dos ciclos de 10 segundos antes de disparar una alarma de sobre-demanda a ser notificada. Para el primer ciclo se consideró los circuitos “h”, “j” y “n”, mientras que para el segundo ciclo los circuitos “b”, “e”, “g” y “m”. El umbral de sobre-demanda se estableció en 10 A, un valor que se consigue con el encendido de todos los circuitos de iluminación más el uso del horno microondas (máquina que más consume en la vivienda).

Se cumplió con la elaboración de un Sistema de Control Contra-Incendios, para la detección de focos de fuego a través de detectores de humo, como también de fugas de gas y contaminación de monóxido de carbono. Para esto se instalaron tres detectores de humo en las partes centrales de cada área de la vivienda (cocina, sala-comedor y dormitorio), como también detectores de GLP y CO en la cocina. Estos de cablearon hasta las entradas digitales de los controladores y se virtualizaron a través de bloques tipo UFPT_switch. Se desarrolló un bloque funcional UFPT_fire_control con las funcionalidades propuestas para el control contra-incendios y se conectaron los sensores respectivos. Dicho bloque tiene como salidas alarmas para cada supervisión como la activación de una sirena, estrobo y apertura de la puerta de acceso. Se verifico las funciones de sirena y estrobo implementados en el timbre y circuitos de iluminación respectivamente con éxito.

Parte del bloque UFPT_fire_control es el control de ventilación de la cocina a través de la supervisión de ventana abierta. Para esto se dispuso de una señal para el contacto magnético instalado en la ventana de la cocina. Este sistema se pudo comprobar con éxito ya que cuando se encontraba la ventana cerrada se constató que el detector de CO se activaba cuando se encendía en calefón por un tiempo no mayor a 20 segundos. Un caso más que preocupante que llevo a la conclusión de una falta de mantenimiento en el calefón.

Se cumplió con la implementación de un Sistema de Control de Acceso para la puerta principal de la vivienda. Se instaló una cerradura electromagnética y un contacto magnético en la puerta de ingreso, cuyas señales fueron conectadas al bloque funcional UFPT_access_control. Se verifico el algoritmo de apertura y cierre propuesto para la puesta, como el control desde el interior (salida) y el exterior (entrada) con la función de pulsación larga del pulsador del timbre.

Se cumplió con la elaboración del Sistema de Control Contra-Intrusión, a través del desarrollo del bloque UFPT_vigilance_control que permite la vigilancia del estado de las ventanas, puerta de acceso y detectores de movimiento para disparar una alarma de intrusión. Se verifico el funcionamiento de este bloque, como la aplicación de la función de simulación de presencia desarrollado en cada bloque UFPT_occupancy_control.

Una parte fundamental del presente proyecto fue la conexión de la red de control con Servicios de Internet como correo electrónico, envió de mensajes de texto SMS y registro de datos IoT. Se verifico él envió de notificaciones de alarma a través de los bloques UFPT_email_notification (correo electrónico a través de Gmail) y UFPT_sms_notification (mensajes de texto a través de Twilio) con éxito. Como también se verifico él envió de datos o mediciones de estancia y energía a la aplicación de ThingSpeak a través del bloque UFPT_iot_application con un intervalo de 15 minutos.

Se cumplió con el desarrollo de una Aplicación Cliente Multiplataforma con la tecnología JavaFX y la arquitectura MVC. Se pudo corroborar la potencialidad que tiene JavaFX para la construcción de aplicaciones gráficas y su versatilidad de poder ser copilada tanto para escritorio como web.

El proyecto demostró ser un sistema integral y escalable, gracias al concepto de IzoT, el manejo de una estructura horizontal a través de sus servicios de control (bloques funcionales, datapoints y conexiones), permite rápidos cambios y fáciles adiciones de nuevos sistemas, funcionalidades y automatismos, sin tener que reestructurar o reprogramar lo ya implementado. Solo bastara en el futuro con crear nuevos dispositivos o bloques funcionales y comisionarlos a la red ya existente. Es importante recalcar también que gracias al API REST del Servidor IzoT el sistema también está preparado para integrarse, bajo

arquitectura cliente-servidor, con otras tecnologías de control no IIoT, como también con servicios e interfaces.

Como conclusión final se puede decir que el concepto del Internet de las Cosas Industriales demostró ser una solución para el intercambio de información entre objetos sin la necesidad de pasar por un servidor centralizado y solo bajo demanda (comunicación punto-a-punto), en contra posición a la arquitectura cliente-servidor, que es como se ofrece hoy en día, las aplicaciones IoT. Esto como ya se mencionó anteriormente, efectiviza el consumo del ancho de banda, no solo en redes locales, sino que también en redes de áreas amplias. Esto será fundamental en el futuro para que las nuevas aplicaciones IoT puedan convivir con las actuales redes de comunicación de datos bajo el modelo TCP/IP, especialmente si se considera que en los próximos años la cantidad de objetos o cosas que se conectaran a Internet aumentaran de forma exponencial.

4.2. RECOMENDACIONES

Se debe considerar el mantenimiento periódico de la red de control para una certeza y confianza del correcto funcionamiento del sistema, como también el incremento de vida util de los diferentes componentes. Para esto se recomienda la limpieza de los diferentes controladores como el reajuste de sus bornas, verificación del buen estado del cableado y revisión de la temperatura de las placas electrónicas. Es ideal que este mantenimiento se haga en lapsos de 6 meses a un año.

Como parte del mantenimiento, se debe verificar el buen funcionamiento de sensores y detectores, especialmente los detectores de humo, gas y CO, ya que su actuación es poco usual y critica.

Es importante mantener actualizado tanto el sistema operativo como el SDK del IzoT en los diferentes SBCs de los controladores. Para esto se debe comprobar la publicación de nuevas versiones de forma periódica o junto a los mantenimientos.

Para la implementación de la Aplicación Cliente IIoT en una página web, se necesitó realizar una excepción al dominio de la página en el *Panel de Control de Java* como modificaciones en el fichero de *Políticas de Seguridad* del JRE. Esto puede convertirse en un inconveniente pues debe ser realizado en cada máquina cliente donde se requiera abrir la aplicación. Se debe considerar la adquisición de un certificado SSL para el servidor web de la aplicación y así evitar estos procedimientos adicionales.

Lamentablemente, una desventaja detectada durante el desarrollo de la Aplicación Cliente IIoT, fue la no posibilidad de ser ejecutada o abierta desde dispositivos móviles, tanto con Android como iOS. Esto debido a que JavaFX no posee soporte para aplicaciones móviles por parte de **Oracle Corporation**. Se debe considerar en el futuro el desarrollo de una aplicación móvil que se conecte a la red de control a través de su API REST con el fin de generar un mayor confort en los ocupantes de la vivienda.

4.3. BIBLIOGRAFÍA

Alex Velasquez, Echelon Corporation. (4 de Abril de 2014). *IzoT Platform Hands-On Lab*. Obtenido de echelon.com: http://echelon.com/assets/bltf5a65a4326ff24c3/Hands-on-lab-Instruction-Manual-Final%20(1).pdf

AREA TECNOLOGIA. (s.f.). *INTERNET DE LAS COSAS*. Obtenido de areatecnologia.com: http://www.areatecnologia.com/nuevas-tecnologias/internet-de-las-cosas.html

Asier Marqués. (2013). *Conceptos sobre APIs REST.* Obtenido de asiermarques.com: http://asiermarques.com/2013/conceptos-sobre-apis-rest/

BBVA. (23 de Marzo de 2016). *API REST: qué es y cuáles son sus ventajas en el desarrollo de proyectos.* Obtenido de bbvaopen4u.com: https://bbvaopen4u.com/es/actualidad/api-rest-que-es-y-cuales-son-sus-ventajas-en-el-desarrollo-de-proyectos

CCM Benchmark Group. (Enero de 2017). *¿Qué significa TCP/IP?* Obtenido de es.ccm.net: http://es.ccm.net/contents/282-tcp-ip

CEDOM. (s.f.). *Qué es Domótica*. Obtenido de www.cedom.es: http://www.cedom.es/sobre-domotica/que-es-domotica

Cesar Calderón, Henry Sánchez. (2016). *Domótica una mejor calidad de vida*. Obtenido de www.monografias.com: http://www.monografias.com/trabajos107/domotica-mejor-calidad-vida/domotica-mejor-calidad-vida.shtml

Cristian Henao. (17 de Junio de 2013). *Ejemplo Modelo Vista Controlador*. Obtenido de codejavu.blogspot.com: http://codejavu.blogspot.com/2013/06/ejemplo-modelo-vista-controlador.html

Echelon Corporation. (2013). *Requirements for the Industrial Internet of Things.* Obtenido de echelon.com: http://echelon.com/assets/bltd38400cf0493d39a/IIoT_WP.pdf

Echelon Corporation. (2014). *Creating & Controlling Industrial IoT Devices with Echelon IzoT Platform* . Obtenido de echelon.com: https://player.vimeo.com/video/86437790

Echelon Corporation. (2014). *IzoT Device Stack EX.* Obtenido de echelon.com: http://www.echelon.com/assets/bltb4957b41c76e4938/IzoT-Device-Stack-EX-datasheet.pdf

Echelon Corporation. (2014). *IzoT Server Stack.* Obtenido de echelon.com: http://www.echelon.com/assets/blt3ec8596112a29305/IzoT-Server-stack-datasheet.pdf

Echelon Corporation. (4 de Agosto de 2014). *The Industrial Internet of Things is Really Control Networking 2.0*. Obtenido de echelon.com: http://echelon.com/assets/blt39358b5bf160dc36/M2M-IIoT-Control-networking-2.0-whitepaper.pdf

Echelon Corporation. (2015). *Open LDV Programmer's Guide.* Obtenido de echelon.com: http://downloads.echelon.com/support/documentation/manuals/devtools/078-0275-01E_OpenLDV_Programmer's_Guide.pdf

Echelon Corporation. (2016). *IzoT Manual.* Obtenido de echelon.com: http://downloads.echelon.com/support/documentation/izot/index.htm

Elena Sanz. (s.f.). *¿Qué es el "Internet de las cosas"?* Obtenido de muyinteresante.es: http://www.muyinteresante.es/curiosidades/preguntas-respuestas/ique-es-el-qinternet-de-las-cosasq

ENDESA. (2014). *Smart Buildings*. Obtenido de www.endesaeduca.com: http://www.endesaeduca.com/Endesa_educa/recursos-interactivos/smart-city/smart-building

ERESHOP. (2017). *Especificaciones I2C-AI418ML*. Obtenido de www.ereshop.com: https://www.ereshop.com/shop/index.php?main_page=product_info&cPath=143_179&products_id=826&zenid=6792993df484ade64655834256aecebf

ERESHOP. (2017). *Especificaciones I2C-IN830S*. Obtenido de www.ereshop.com: https://www.ereshop.com/shop/index.php?main_page=product_info&cPath=143_181&products_id=777&zenid=6792993df484ade64655834256aecebf

ERESHOP. (2017). *Especificaciones I2C-RL824M*. Obtenido de www.ereshop.com: https://www.ereshop.com/shop/index.php?main_page=product_info&cPath=143_178&products_id=767&zenid=6792993df484ade64655834256aecebf

Google. (2016). *Ayuda de Gmail.* Obtenido de google.com: https://support.google.com/mail/?hl=es#topic=7065107

Hardkernel. (2013). *Products ODROID-C2*. Obtenido de hardkernel.com: http://www.hardkernel.com/main/products/prdt_info.php?g_code=G145457216438&tab_idx=2

Hardkernel. (2017). *Odroid Wiki, odroid-c2*. Obtenido de odroid.com: http://odroid.com/dokuwiki/doku.php?id=en:odroid-c2

InterAct. (28 de Junio de 2002). *LONWORK*. Obtenido de odisea.ii.uam.es: http://odisea.ii.uam.es/esp/recursos/Lonwork.htm

Jacinto Fung. (11 de Noviembre de 2011). *Sistema Distribuido vs. Sistema Centralizado, o Sistema Distribuido en un Sistema Centralizado*. Obtenido de iscbunkerramo.blogspot.com: http://iscbunkerramo.blogspot.com/2011/11/sistema-distribuido-vs-sistema.html

Jack 2.0. (22 de Junio de 2014). *Patrón de Diseño MVC (Modelo Vista Controlador) y DAO (Data Access Object)*. Obtenido de jossjack.wordpress.com: https://jossjack.wordpress.com/2014/06/22/patron-de-diseno-mvc-modelo-vista-controlador-y-dao-data-access-object/

JFXtras. (s.f.). *Sitio Web de JFxtras*. Obtenido de jfxtras.org: http://jfxtras.org/

jjtorres. (2014). *¿Qué es y cómo funciona el Internet de las cosas?* Obtenido de hipertextual.com: https://hipertextual.com/archivo/2014/10/internet-cosas/

John Martens. (2014). *What is JavaFX?* Obtenido de javafxtutorials.com: http://www.javafxtutorials.com/whatisjavafx/

LonMark. (Diciembre de 2012). *SCPT Master List Version 14 Revision 00* . Obtenido de lonmark.org: http://www.lonmark.org/technical_resources/resource_files/scpt.pdf

LonMark. (Diciembre de 2012). *SNVT Master List Version 14 Revision 00* . Obtenido de lonmark.org: http://www.lonmark.org/technical_resources/resource_files/snvt.pdf

María Jesús Lamarca Lapuente. (2013). *Servicios de Internet.* Obtenido de www.hipertexto.info: http://www.hipertexto.info/documentos/serv_internet.htm

MathWorks. (2017). *ThingSpeak Documentation*. Obtenido de es.mathworks.com: https://es.mathworks.com/help/thingspeak/

McRock Capital. (30 de Marzo de 2014). *McRock's Industrial Internet of Things Report 2014* . Obtenido de issuu.com: https://issuu.com/mcrock/docs/mcrock_industrial_internet_of_thing

Miguel Angel Alvarez. (2 de Enero de 2014). *Qué es MVC*. Obtenido de desarrolloweb.com: https://desarrolloweb.com/articulos/que-es-mvc.html

Moor Insights & Strategy. (23 de Octubre de 2013). *Behaviorally Segmenting the Internet of Things (IoT)*. Obtenido de echelon.com: http://echelon.com/assets/bltf62408d68b065892/M2M-IIoT-Behaviorally-Segmenting-the-IoT-by-Moor-Insights-Strategy-whitepaper.pdf

Moor Insights & Strategy. (29 de Octubre de 2013). *Connecting with the Industrial Internet of Things (IIoT)*. Obtenido de echelon.com: http://echelon.com/assets/blt7227c218cd3869cb/IIOT-Connecting-with-the-Industrial-Internet-of-Things-IIoT-by-Moor-Insights-and-Strategy-whitepaper.pdf

Oracle Corporation. (s.f.). *¿Qué es JavaFX?* Obtenido de java.com: https://www.java.com/es/download/faq/javafx.xml

Oracle Corporation. (2010). *Introducción al conjunto de protocolos TCP/IP*. Obtenido de oracle.com: https://docs.oracle.com/cd/E19957-01/820-2981/6nei0r0r9/index.html

Oracle Corporation. (s.f.). *Información general sobre JavaFX.* Obtenido de java.com: https://www.java.com/es/download/faq/javafx.xml

Oracle Corporation. (s.f.). *Java Platform, Standard Edition (Java SE) 8*. Obtenido de docs.oracle.com: http://docs.oracle.com/javase/8/javase-clienttechnologies.htm

Pablo Turmero. (s.f.). *Control de demanda y factor de potencia*. Obtenido de www.monografias.com: http://www.monografias.com/trabajos104/control-demanda-y-factor-potencia/control-demanda-y-factor-potencia.shtml

Perez Saucedo Jose Ramon, Reyes Padilla Karenh Gyssel. (s.f.). *Domótica*. Obtenido de www.monografias.com: http://www.monografias.com/trabajos35/domotica/domotica.shtml

PyAr, Python Argentina. Python Software Foundation. (2014). *Tutorial de Python (y Django!) en Español.* Obtenido de python.org.ar: http://docs.python.org.ar/tutorial/

Python Software Foundation. (2016). *Lenguaje de Programación Python.* Obtenido de es.wikipedia.org: https://es.wikipedia.org/wiki/Python

Python Software Foundation. (2017). *Client library for the thingspeak.com API - thingspeak 0.4.1*. Obtenido de pypi.python.org: https://pypi.python.org/pypi/thingspeak/0.4.1

Python Software Foundation. (9 de Febrero de 2017). *The Python Standard Library, 21.17. smtplib — SMTP protocol client*. Obtenido de docs.python.org: https://docs.python.org/3/library/smtplib.html

Scott Hommel, Oracle Corporation. (Junio de 2013). *Using JavaFX Properties and Binding*. Obtenido de docs.oracle.com: http://docs.oracle.com/javafx/2/binding/jfxpub-binding.htm

Scott Hommel, Technical Writer, Oracle. (2013). *Using JavaFX Properties and Binding.* Obtenido de oracle.com: http://docs.oracle.com/javafx/2/binding/jfxpub-binding.htm

The Apache Software Foundation. (2017). *Apache HttpComponents*. Obtenido de hc.apache.org: https://hc.apache.org/

Twilio. (2017). *Twilio Docs.* Obtenido de www.twilio.com: https://www.twilio.com/docs/

Twilio. (2017). *TWILIO REST API*. Obtenido de www.twilio.com: https://www.twilio.com/docs/api/rest

APÉNDICE A

REGISTRO FOTOGRÁFICO – ESTADO INICIAL DE LA VIVIENDA

APÉNDICE B

PLANOS ARQUITECTÓNICOS Y ELÉCTRICOS

PLANOS Y DIAGRAMAS DE DISEÑO

APÉNDICE C

MANUAL FUNCIONAL - MODELADO DE BLOQUES FUNCIONALES

APÉNDICE D

DISEÑO DE PROGRAMAS FUNCIONALES - DIAGRAMAS DE FLUJO

APÉNDICE E

REGISTRO FOTOGRÁFICO – INSTALACIÓN DE LA RED DE CONTROL DISTRIBUIDA

APÉNDICE F

CÓDIGO DE PROGRAMAS – FIRMWARE Y FICHEROS DE LOS CONTROLADORES

APÉNDICE G

PROGRAMACIÓN DE LA RED DE CONTROL LONTALK/IP

APÉNDICE H

CÓDIGO DE PROGRAMA – APLICACIÓN CLIENTE IIOT

APÉNDICE I

GUÍA DE USUARIO – APLICACIÓN CLIENTE IIOT

APÉNDICE J

REGISTRO FOTOGRÁFICO – PRUEBAS Y RESULTADOS

ANEXO A

MANUAL TÉCNICO – TECNOLOGÍA IZOT

ANEXO B

MANUAL TÉCNICO – IZOT SDK

ANEXO C

MANUAL TÉCNICO – SERVIDOR IZOT API REST

ANEXO D

MANUALES DE USUARIOS – COMPONENTES DE LOS CONTROLADORES

ANEXO E

FICHAS TÉCNICAS – PERIFÉRICOS

ANEXO F

FICHAS TÉCNICAS – ELEMENTOS DE RED