

¿Qué es un protocolo 6LoWPAN?, ¿Para qué se usa? Ejemplifique

Desde el año 2003, la IEEE liberó el estándar 802.15.4 para redes inalámbricas de área personal (WPAN), proporcionando el primer estándar mundial de radio de baja potencia. Poco después, el grupo denominado “*ZigBee Alliance*” desarrolló una solución para redes ad-hoc sobre IEEE 802.15.4. *ZigBee*, sin embargo, tiene problemas de escalabilidad e integración y aunque existen arquitecturas de clúster que permiten que la red *ZigBee* se conecte a Internet, los dispositivos *ZigBee* no pueden ser conectados directamente a la misma. Por lo tanto, se necesita un nuevo paradigma que permita la posibilidad de que los dispositivos inalámbricos IEEE 802.15.4 se puedan conectar directamente a Internet.

Para tal efecto, la IETF ha propuesto el protocolo **6LoWPAN**, el cual ofrece a los dispositivos IEEE 802.15.4 interoperabilidad con otro tipo de redes a través de un dispositivo intermedio que permite la conexión directa de los nodos mediante una dirección IP. Este estándar consiste en implementar una ligera pila IPv6 en los dispositivos inalámbricos de baja potencia.

El protocolo 6LoWPAN es un estándar abierto definido por la IETF (*Internet Engineering Task Force*), tal como lo indica su nombre “*IPv6 over Low-Power Wireless Personal Area Networks*” es una tecnología de red o capa de adaptación que permite el envío eficiente de paquetes IPv6 dentro de tramas pequeñas de la capa de enlace de datos como las definidas en el estándar IEEE 802.15.4.

En la figura 1 se puede observar que la red 6LoWPAN se conecta a Internet mediante un *router* frontera (*Edge Router*), el cual realiza tres funciones fundamentales:

- Intercambio de información entre los nodos 6LoWPAN e Internet (u otra red IPv6).
- Intercambio de información entre los nodos 6LoWPAN.

Generación y mantenimiento de la red **Pila de 6LoWPAN**

Como se dijo antes, hasta ahora, se han requerido de puertas de enlace a nivel de la capa de aplicación para hacer que dispositivos con tecnologías como *Zigbee* o *Bluetooth* se puedan conectar a Internet. 6LoWPAN resuelve este dilema introduciendo una capa de adaptación entre la capa de enlace de datos y la capa de red para habilitar la transmisión de tramas IPv6 sobre tecnologías como IEEE 802.15.4.

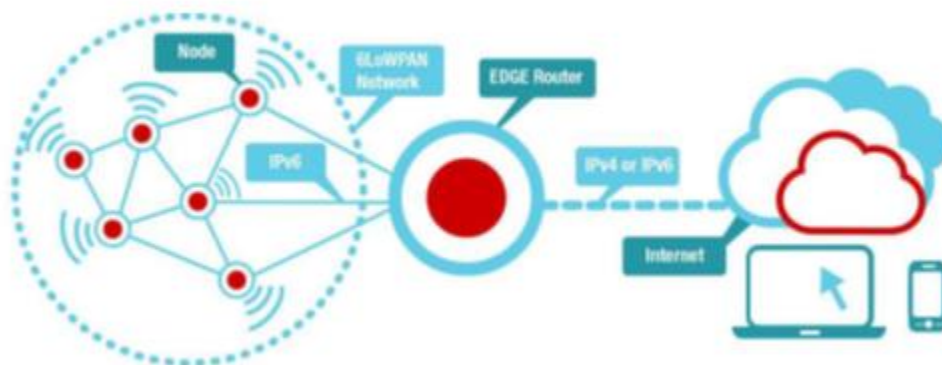


FIGURA 1 ARQUITECTURA DE LA RED 6LoWPAN

Los *router* frontera reenvían tramas a nivel de la capa de red, por lo cual no necesitan de ninguna capa de aplicación. Otras arquitecturas como *Zigbee*, *Z-wave*, *Bluetooth* o redes propietarias requieren de complejas puertas de enlace a nivel aplicación para conectarse a redes IP. Estas puertas de enlace a nivel aplicación deben entender cualquiera de los perfiles de aplicación que

son usados en la red, por tanto, cualquier cambio también viene acompañado de cambios en la puerta de enlace. En contraste, los *router* frontera, funcionan independientemente de los protocolos de aplicación usados en la red 6LoWPAN. Esto permite disminuir la carga en el *router* frontera en términos de procesamiento, haciendo posible el uso de dispositivos de menor potencia y menor costo, que ejecuten software más simple y dispongan de un hardware menos complejo.

Problemas de la tecnología actual

La comunicación directa de nodos en redes IP requiere de muchos protocolos que son demandantes para dispositivos inalámbricos de baja potencia, por ejemplo, IPv6 incluye seguridad para autenticación y encriptación de datos, dichas técnicas pueden ser muy complejas y demandantes para este tipo de dispositivos. Los servicios ofrecidos en Internet generalmente se basan en servicios web, usando principalmente protocolos como TCP, HTTP, SOAP y XML los cuales utilizan complejos patrones de transacción. Además, los protocolos de Internet actuales requieren de enlaces que manejan tramas de gran tamaño (mínimo 1280 bytes para IPv6), y pesados protocolos de aplicación que requieren de un ancho de banda considerable, por lo que no están optimizados para redes inalámbricas de baja potencia, ya que en este tipo de redes hay menos fiabilidad debido a fallas en nodos, agotamiento de energía, y ciclos de inactividad. Estas limitaciones han restringido al IoT a dispositivos que cuenten con un procesador potente y con un sistema operativo que maneje la pila TCP/IP para tener la capacidad de un enlace directo mediante una dirección IP.

Pila de 6LoWPAN

Como se dijo antes, hasta ahora, se han requerido de puertas de enlace a nivel de la capa de aplicación para hacer que dispositivos con tecnologías como *Zigbee* o *Bluetooth* se puedan conectar a Internet. 6LoWPAN resuelve este dilema introduciendo una capa de adaptación entre la capa de enlace de datos y la capa de red para habilitar la transmisión de tramas IPv6 sobre tecnologías como IEEE 802.15.4.

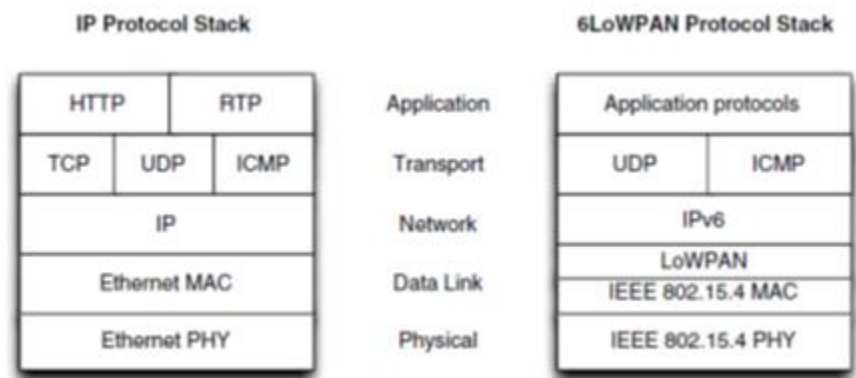


FIGURA 2 PILAS TCP/IP CONTRA 6LoWPAN

En la figura 2, se muestra el uso de IEEE 802.15.4 en 6LoWPAN. La capa de enlace de datos provee de un enlace confiable entre dos nodos conectados detectando errores en la conexión que pueden ocurrir en la capa física durante transmisión y recepción. La capa de enlace de datos incluye la capa de control de acceso al medio (MAC) que usa técnicas como CSMA-CA (*Carrier Sense Multiple Access - Collision Avoidance*) en donde se comprueba que ningún otro dispositivo está transmitiendo antes de enviar información, el entramado de la información, así como la capa 6LoWPAN que provee adaptación de IPv6 a IEEE 802.15.4.

La capa de red direcciona información a través de la red, incluso si se requieren de múltiples saltos. La capa de transporte genera sesiones de comunicación entre aplicaciones ejecutadas en los dispositivos. Esta capa permite que múltiples aplicaciones tengan su propio canal de comunicación. TCP es el protocolo de transporte dominante en Internet, sin embargo, este protocolo orientado a conexión tiene cabeceras demasiado grandes y por lo tanto no es siempre factible en dispositivos de bajo consumo de energía. Para ese tipo de dispositivos, UDP, es una mejor opción al ser un protocolo no orientado a conexión que maneja cabeceras pequeñas.

Finalmente, la capa de aplicación es responsable del formato de la información. Una capa de aplicación ampliamente usada en Internet es HTTP sobre TCP. HTTP usa XML que es un lenguaje basado en texto con una cabecera grande, por lo tanto, no es óptimo su uso en redes 6LoWPAN. Por esta razón, se han desarrollado protocolos de aplicación alternativos, como CoAP (*Constrained Application Protocol*), un protocolo sobre UDP con un mecanismo REST muy similar a HTTP. CoAP está definido por IETF en el RFC7252 y define retransmisión, mensajes confirmados y no confirmados, suscripción y descubrimiento de recursos.

El protocolo 6LoWPAN puede considerarse como una gran alternativa para la implementación del IoT ya que al utilizar IPv6 se tiene conectividad e interoperabilidad con otras redes IP, se dispone de gran número de direcciones IP (2¹²⁸) lo cual permitirá conectar millones de objetos/ cosas de manera inalámbrica a Internet utilizando la infraestructura existente y finalmente ofrece una integración transparente con la misma.

EJEMPLO DE USO

En este estudio se implementa un prototipo de gestión de luminarias que cumple los siguientes requerimientos:

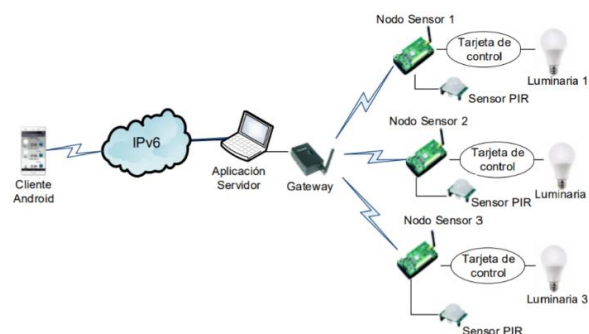
- Permite la gestión remota del encendido y apagado de las luminarias, control de la intensidad lumínica, temporización, detección de movimiento y simulación de presencia.
- La gestión remota se hace con dispositivos basados en Arduino, conectados a la red IPv6.
- El direccionamiento de los nodos sensores debe ser compatible con IPv6 y permitir una conectividad de extremo a extremo.
- Usa herramientas estándares para las comunicaciones en la capa aplicación de la arquitectura de la red inalámbrica de sensores.

Componentes del prototipo

El prototipo implementado (Figura 1) está formado por un dispositivo Android llamado cliente, una aplicación alojada en un computador llamada servidor, un nodo gateway, tres nodos sensores que forman parte de una red sensor inalámbrica que opera con IEEE 802.15.4 y la red IPv6, que se describen a continuación:

Figura 1.

Diagrama del ambiente de prueba

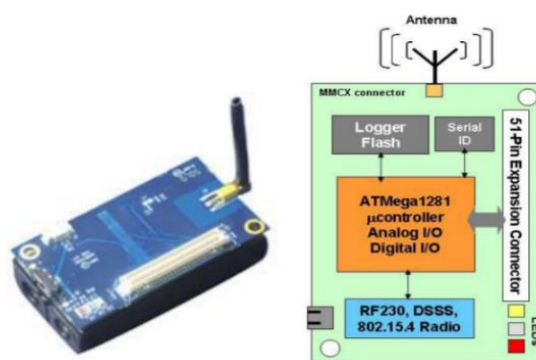


Nodo sensor

Es un nodo IRIS XM2110 (MEMSIC, 2012) (Figura 2) basado en un microcontrolador de baja potencia ATmega1281, de 8 bits, con un transmisor y receptor que operan en la banda 2.4 GHz. Este nodo gestiona el sensor de movimiento y la intensidad de la luminaria por medio de una tarjeta de control diseñada para tal efecto.

Figura 2.

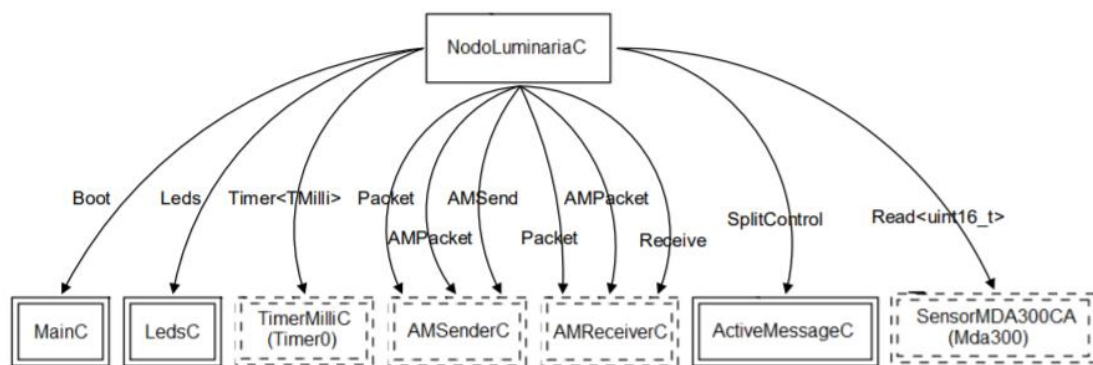
Nodo sensor IRISxm2110



La información recolectada por cada nodo sensor se envía por medio del gateway hacia el servidor. El código de la aplicación fue creado con NesC y sus componentes se presentan en la Figura 3.

Figura 3.

Componentes de la aplicación ejecutada en nodo sensor Iris XM21



En la Figura 4 se presenta el componente de código nodoluminaria.h, en el cual se define la estructura del paquete, que consiste en la cabecera de 6LoWPAN con compresión HC1-HC2. Este paquete forma parte de la carga útil de la trama IEEE 802.15.4.

Figura 4.

Definición del direccionamiento para 6LowPAN

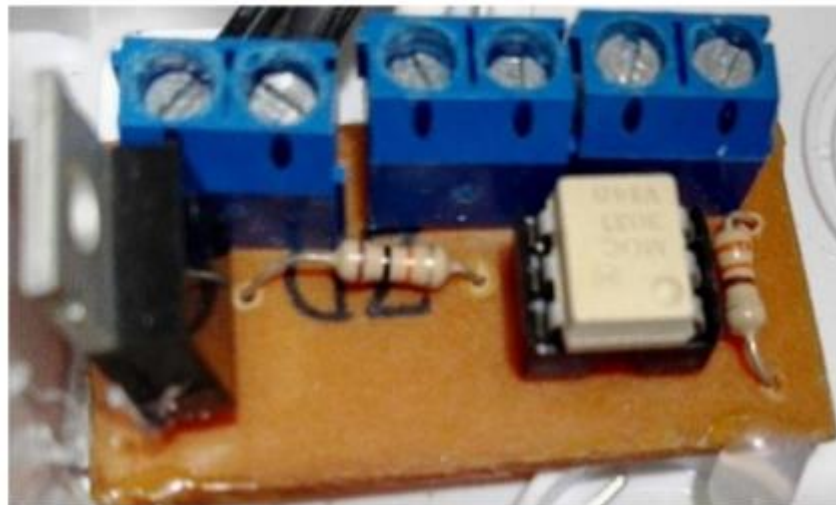
```
enum {
//Tipo mensaje Active Message
AM_BLINKTORADIO = 6,
//Intervalos para tomar medidas
FRECUENCIA_MUESTREO = 10000
};
typedef nx_struct BlinkToRadioMsg {
//Campos 6LoWPAN encapsulados
//en la trama IEEE 802.15.4
nx_uint64_t direccion_origen;
nx_uint64_t direccion_destino;
nx_uint8_t dispatch;
nx_uint8_t HCl;
nx_uint8_t limite_salto;
nx_uint16_t datos;
} BlinkToRadioMsg;
```

Tarjeta de control

En la Figura 5 se presenta el dispositivo, implementado en este trabajo, acoplado al nodo sensor, el cual le permite al nodo sensor controlar el funcionamiento de la luminaria para encenderla, apagarla y controlar el nivel de iluminación.

Figura 5.

Tarjeta de control

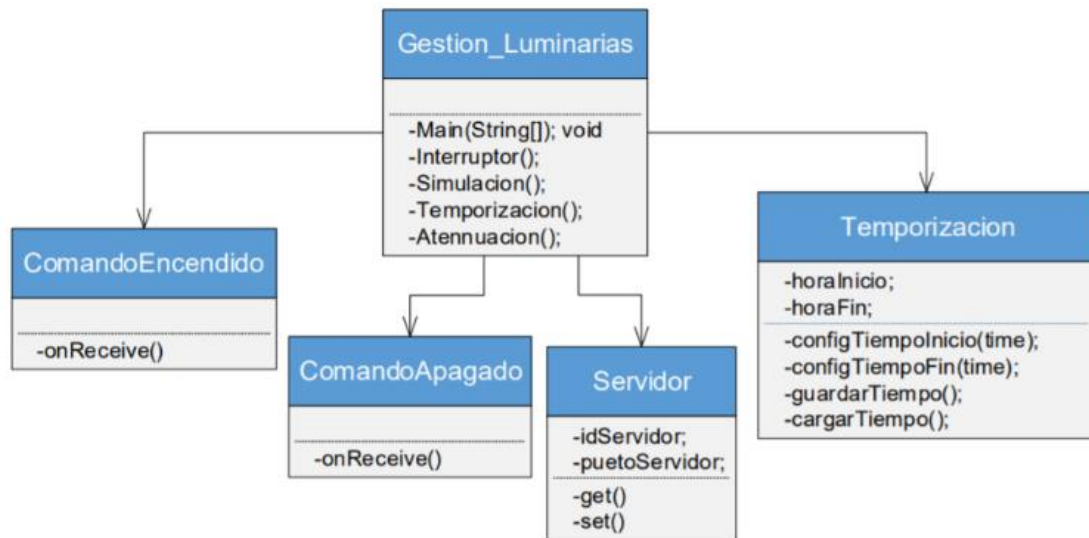


Cliente Android

Debido a que se necesita hacer la gestión remota utilizando una red IPv6, el cliente debe ser ejecutado en un dispositivo con el sistema operativo Android, con la versión 5.0 o superior, habilitado para tener conectividad con IPv6. Para desarrollar el aplicativo se utiliza el lenguaje de programación Java para implementar el sistema de gestión remota de las luminarias. En la Figura 6, se observa el diagrama de clases de la aplicación ejecutado en el cliente Android.

Figura 6.

Diagrama de clases de la aplicación Cliente

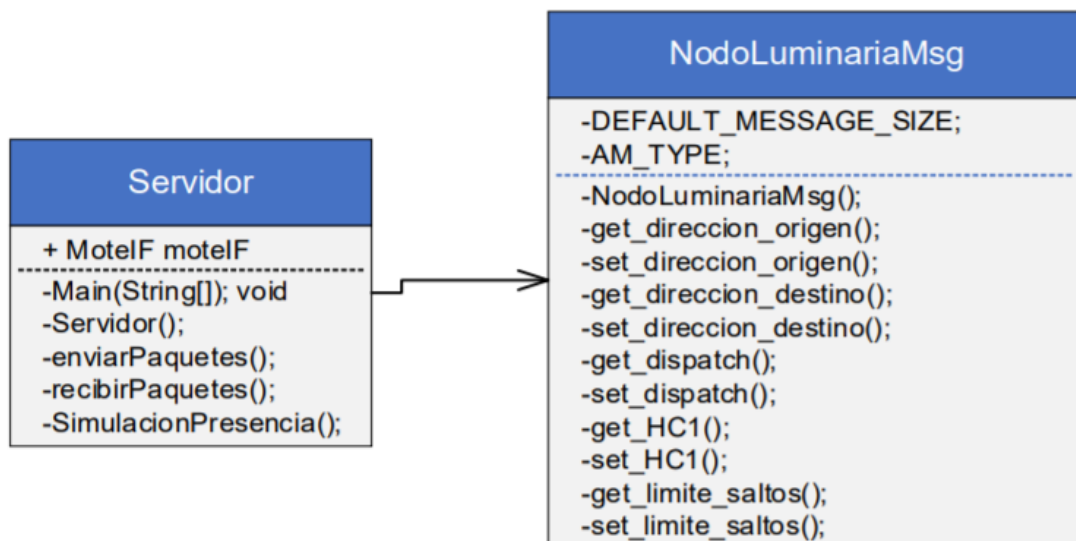


Servidor

Es el computador en el cual se ejecuta la aplicación Servidor para convertir los protocolos de la red sensor inalámbrica con 6LoWPAN al protocolo IPv6 de la red internet. Para hacer la conversión de protocolos, el servidor trabaja conjuntamente con un nodo sensor definido como Gateway, el cual se conecta al puerto serial del servidor. El servidor envía los datos a la aplicación Cliente usando el protocolo IPv6. La aplicación Servidor se implementa empleando lenguaje Java y el IDE Eclipse. El diagrama de clases se muestra en la Figura 7.

Figura 7

Diagrama de clases de la aplicación Servidor



El servidor envía los paquetes hacia el nodo Gateway y al dispositivo cliente. Los paquetes se capturan en el terminal del servidor usando las herramientas MsgReader (Sicos1977, 2013). Para visualizar los paquetes en el servidor se utilizó la herramienta Java Listen, que funciona como un monitor de paquetes (sniffer, en inglés) y facilita la presentación de los paquetes binarios en la pantalla.

Nodo gateway

Implementado con el dispositivo MIB520 (MEMSIC, 2010), este nodo recolecta información de la red sensor inalámbrica que opera con el protocolo IEEE 802.15.4 y la transmite al servidor; además, recibe los datos desde el servidor y los retransmite hacia los nodos sensores para el control de las luminarias. Para las comunicaciones entre el servidor y los nodos sensores, el nodo Gateway utiliza la aplicación BaseStation (TinyOS, 2009), herramienta de TinyOS, que permite al Gateway operar como un puente entre el puerto serie del servidor y la red inalámbrica de sensores. Cuando el nodo Gateway recibe un paquete del puerto serie del servidor, lo transmite por radio; cuando recibe un paquete por la radio, lo transmite al puerto serie. De esta manera, la herramienta BaseStation establece la comunicación entre el servidor y los nodos sensores. La herramienta MIG (TinyOS, 2011) es empleada para encapsular el paquete en la carga útil de la trama IEEE 802.15.4; se pueden leer e imprimir los campos de la carga útil de la trama 802.15.4. Utilizamos la herramienta MsgReader (TinyOS, 2010) para extraer únicamente la carga útil de la trama omitiendo la cabecera IEEE 802.15.4.

Red IPv6

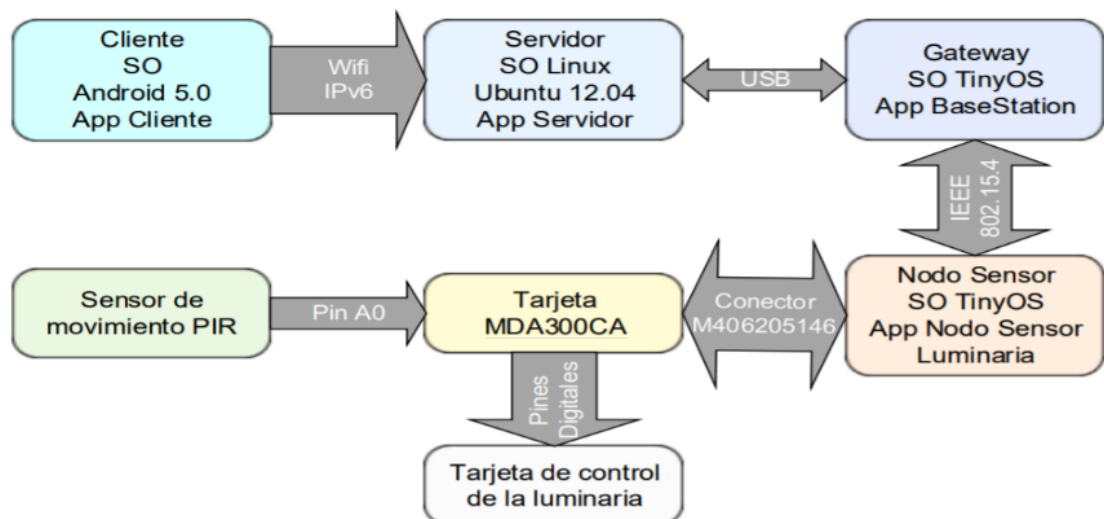
La red IPv6 se implementa con 3 ruteadores Cisco 1841 conectados en topología anillo y configurados con IPv6.

Componentes del sistema

Como se muestra en la Figura 8, inicialmente la aplicación Cliente Android, que se ejecuta en un teléfono celular, enviará una petición a la aplicación Servidor que opera con Ubuntu; estos dos componentes se comunicarán usando IPv6. Luego, el servidor envía un mensaje a la red inalámbrica IEEE 802.15.4 utilizando el Gateway. El Gateway y los nodos sensores operan con el sistema operativo TinyOS. En la misma figura, se aprecian las aplicaciones que se ejecutan en cada elemento que forma parte del prototipo. Con respecto a la manera en que se comunicarán los sistemas, las flechas indican la tecnología de comunicación empleada.

Figura 8.

Componentes del sistema



La luminaria prototipo que se implementó para controlarla ser controlada en forma remota se observa en la Figura 9.

Figura 9.

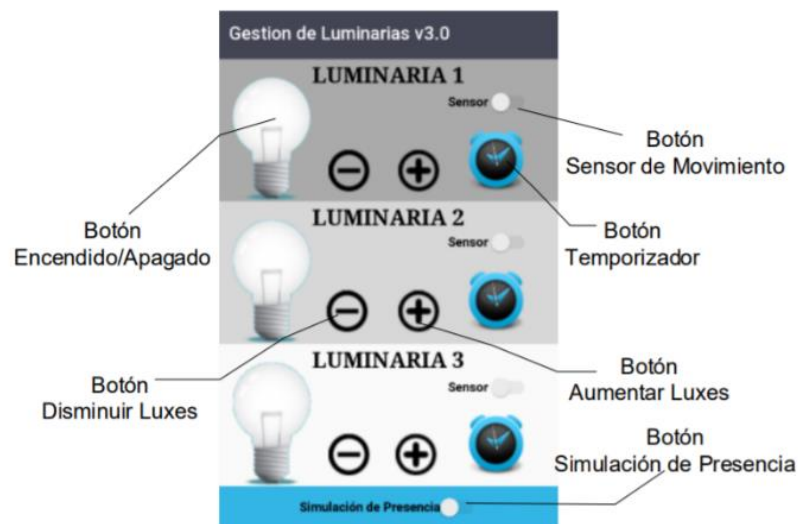
Luminaria implementada



La aplicación desarrollada para el dispositivo con el sistema operativo Android (teléfono celular) tiene una interfaz (Figura 10) que permite activar y desactivar la temporización, elegir la hora que se enciende o se apaga la luminaria y presentar los datos de configuración del temporizador.

Figura 10.

Interfaz para el control de la luminaria



Configurados todos los componentes del prototipo, se validó el funcionamiento del sistema propuesto. Las funciones de encendido, apagado, temporización de la luminaria y control de la intensidad de iluminación funcionaron de acuerdo con los requerimientos de diseño.

En la Tabla 1, se presentan las direcciones IPv6 configuradas en los nodos sensores para hacer las pruebas de conectividad con el teléfono celular configurado con IPv6. El resultado de la función ping, que verifica la conectividad entre el nodo sensor y el teléfono celular, fue exitoso.

Tabla 1.

Direcciones de los nodos sensores

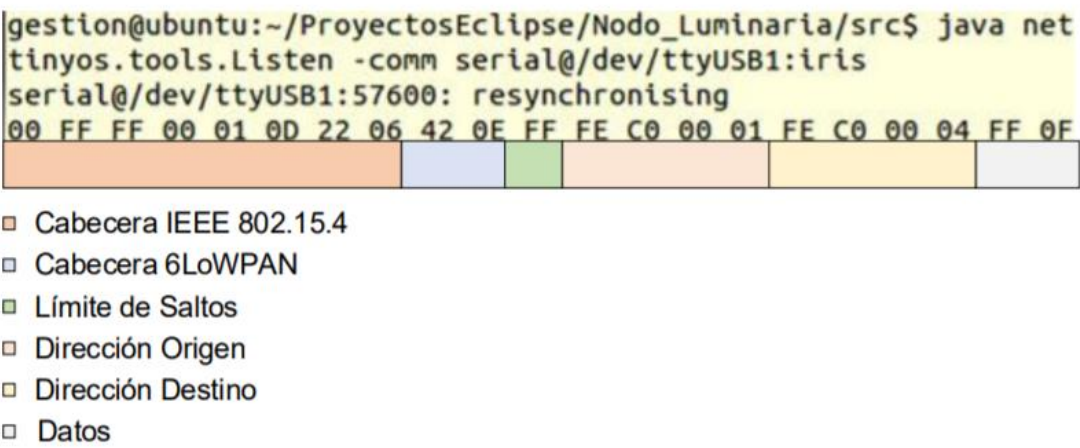
| Columna 1 | Columna 2 |
|-------------|-----------|
| Gateway | FECO::1B |
| Luminaria 1 | FECO::1 |
| Luminaria 2 | FECO::2 |
| Luminaria 3 | FECO::3 |

En la Figura 11, se presenta la información del paquete 6LoWPAN enviado por el nodo sensor, capturado en el servidor, para ser convertido en un paquete IPv6 y recibido por el teléfono celular.

Para medir la intensidad de iluminación, se usó la aplicación basada en Android denominada ‘luxómetro’; esta aplicación utiliza el sensor que está ubicado en la parte frontal del dispositivo celular.

Figura 11.

Paquete 6LoWPAN recibido en el servidor



Control de luminosidad

Se obtuvieron las medidas de intensidad de luz de la luminaria utilizando el luxómetro (ver Tabla 2); la disminución y el aumento de la intensidad de luz medida permitieron comprobar el control sobre la luminaria desde el teléfono celular. La intensidad de luz medida depende del voltaje suministrado a la luminaria y de los luxes del ambiente. La variación de lúmenes medidos al pasar de un nivel de intensidad a otro no es fija, debido al tipo de foco y a la variación del voltaje que tiene un valor promedio de 13 voltios.

Tabla 2.

Medidas obtenidas con el luxómetro

| Número de Medida | Ambiente (Luxes) | Luxómetro (Luxes) | Ambiente (Luxes) | Luxómetro (Luxes) |
|------------------|------------------|-------------------|------------------|-------------------|
| 1 | 146 | 4100 | 79 | 82 |
| 2 | 175 | 3595 | 104 | 466 |
| 3 | 116 | 2739 | 91 | 905 |
| 4 | 152 | 554 | 115 | 3278 |
| 5 | 110 | 110 | 146 | 3486 |

Temporización

En la Tabla 3, se presentan los resultados de la prueba de la temporización, en la cual se pueden apreciar los tiempos de respuesta del control de encendido de la luminaria en forma remota.

Tabla 3.

Medidas de tiempos de temporización

| Tiempo configurado | | Tiempo medido | |
|--------------------|--------------|----------------|--------------|
| Hora Encendido | Hora Apagado | Hora Encendido | Hora Apagado |
| 18:00:00 | 21:00:00 | 18:00:23 | 21:00:35 |
| 13:15:00 | 13:45:00 | 13:15:08 | 13:45:40 |
| 19:34:00 | 22:30:00 | 19:34:21 | 22:30:51 |
| 09:40:00 | 11:00:00 | 09:40:05 | 11:00:02 |

A pesar de que los resultados obtenidos con la función de temporización fueron satisfactorios, cabe mencionar que las acciones de prendido y apagado de la luminaria no fueron ejecutadas de manera inmediata y en el momento indicado, sino que se llevaron a cabo un tiempo después de la hora definida. Esto se debe a que en el celular se estaban ejecutando otras aplicaciones, y el servicio utilizado para el envío de mensajes no tiene una prioridad alta.

Simulación de presencia

Durante esta prueba se verificó que el encendido y apagado de la luminaria de manera aleatoria. Se hicieron pruebas con franjas horarias y se constató el funcionamiento correcto. Los resultados obtenidos cumplen con la función de simulación de presencia codificada.