



POLITÉCNICA

UNIVERSIDAD POLITÉCNICA DE MADRID

Escuela Universitaria de Ingeniería Técnica de
Telecomunicaciones

PROYECTO FIN DE CARRERA

Desarrollo de sistema de gestión de alumbrado
público mediante WSN

AUTOR

Jorge Gálvez Alfaro

TUTOR

Rubén De Diego Martínez

Septiembre de 2013



POLITÉCNICA

PROYECTO FIN DE CARRERA PLAN 2000



UNIVERSIDAD POLITÉCNICA DE MADRID
ESCUELA UNIVERSITARIA DE INGENIERÍA
TÉCNICA DE TELECOMUNICACIÓN
Campus Sur. Ctra. de Valencia km. 7. 28031 Madrid

TEMA: Gestión del conocimiento en redes de sensores inalámbricos.

TÍTULO: Desarrollo de un sistema de gestión de alumbrado público mediante WSN.

AUTOR: Jorge Gálvez Alfaro

TUTOR: Rubén de Diego Martínez

Vº Bº.

DEPARTAMENTO: Diatel

Miembros del tribunal calificador

PRESIDENTE: Ricardo Monedero Mateo

VOCAL: Rubén de Diego Martínez

VOCAL SECRETARIO: Ana Belén García Hernando

DIRECTOR:

Fecha de lectura:

Calificación:

El Secretario,

RESUMEN DEL PROYECTO

Se desarrollado un sistema de gestión de alumbrado público mediante el uso de Redes de Sensores Inalámbricos, WSN, a través del cual se buscará un funcionamiento eficiente de los elementos que la conforman.

En concreto:

- Se realiza un recorrido por las tecnologías existentes que cubren este campo, así como de las actuales soluciones comerciales.
- Se expone una propuesta real de despliegue realizada al Excmo. Ayuntamiento de Pedro Muñoz.
- Se describe el sistema propuesto, su teoría de funcionamiento, algoritmos utilizados y demás aspectos para su completo desarrollo.



Jorge Gálvez Alfaro

Desarrollo de sistema de gestión de alumbrado público mediante WSN



POLITÉCNICA



A mi familia, tanto los que están
como los que no, por construir
mi presente.



AGRADECIMIENTOS

Los agradecimientos que pudiera hacer podrían ser miles y ocupar cientos de página. Para bien o para mal, cada persona con la que me he cruzado en estos años ha aportado algo, por poco que sea, a la consecución de esta carrera.

Aunque a ellos está dedicado este Proyecto Fin de Carrera, agradecer a mi familia que me ha dado la educación necesaria y me ha facilitados los medios que han hecho posible finalizar los estudios superiores. Sin ellos no hubiera sido posible.

Mucha gente conocida durante estos años han aportado su granito de arena a través de sus ánimos, de sus risas, de su simpatía,... y quienes mejor encajan en esta descripción son sin duda mis queridos "Guzmanes", tanto compañeros como trabajadores, grandes amigos repartidos por toda la geografía española que se convertían en la razón para seguir adelante.

A "los amigos locales", es decir, a esos amigos pedroteños que siempre han estado ahí y que me daban las fuerzas para seguir adelante y no abandonar.

A cada uno de los profesores que han sembrado semillas llenas de conocimientos en mi cabeza.

A Roberto, por darme una oportunidad laboral, tan agradecida en estos tiempos que corren.

A Carlos y Vero, pilar fundamental en la aventura iniciada hace ahora un año por Asturias. Sin ellos esto sería imposible de llevar.

A Chema Molina y Daniel Herrera. Gracias por hacerme crecer y ser tan comprensivos.

A D. José Juan Fernández, alcalde de Pedro Muñoz, y a Fátima su secretaria, por su gran predisposición a ayudarme a través de toda la información facilitada.

Y por último, y no por ello menos importante, a mi tutor, Rubén de Diego, por no abandonar, por su gran ayuda y por los grandes esfuerzos realizados para intentar llegar lo más lejos posibles con este proyecto.

A todos y a todas,¡¡Gracias!!

Jorge Gálvez Alfaro



Resumen

Este proyecto surge de la búsqueda de un campo de aplicación de las Redes de Sensores Inalámbricos, WSN, aplicadas a la vida cotidiana. Dicha aplicación consistirá en un sistema de gestión de alumbrado público a través del cual se buscará una reducción del consumo energético y del gasto económico, así como una gestión en tiempo de real de la operativa del alumbrado.

Para ello se desarrollará un sistema basado en nodos instalados en farolas, los cuales se comunicarán entre sí para funcionar de la forma más optimizada posible, complementándose todo ello con un "nodo base", que se encargará de servir de nexo entre la red y los diferentes elementos necesarios para la configuración de los nodos y la recogida de información. Esta información servirá para que una página web pueda mostrar al usuario final toda la información necesaria para tener un control sobre el estado actual de funcionamiento de cada una de las farolas, control del consumo, así como detección de averías.

En este proyecto se describen las tecnologías actuales relacionadas con el campo de las WSN y los sensores, presentando aplicaciones que en la actualidad se encuentran desplegadas.

Se expone también una propuesta real de despliegue presentada al Ayuntamiento de una localidad, Pedro Muñoz, para implementar un proyecto piloto en varias de sus calles. Se describe el entorno, tanto hardware como software, explicando los algoritmos utilizados para las asociaciones entre nodos, diagramas de funcionamiento en las distintas fases de la que está compuesta la operativa de los nodos, la codificación de los programas que se necesitan ejecutar para el correcto funcionamiento del sistema.

Por último, debido a que el campo de las WSN está en constante evolución, se presentarán diversas ideas para implementar diversas mejoras que pudieran ser desplegadas en un futuro, ampliando la oferta de aplicaciones a ofrecer al usuario final.

Palabras clave: nodo, sensor, inalámbrico, farola, red, comunicación, iluminación, presencia, gestión, base de datos, web.



Abstract

This project results from the development for an application field of wireless Sensor Networks (WSN), applied to daily life. That application will consist of a system of street lighting management, through which it will seek a reduction in energy consumption and economic cost, and a real-time management of the operative of the street lighting. To do this, a system based on nodes installed in streetlights will be developed. These nodes will communicate with each other to operate in the most optimized way possible, complementing all with a Base-station, which will act as a link between the network and the components required for configuring the nodes and collecting data from them. This information will help a website to show the end user all the information needed to have a control on the current operating status of each of the streetlights, consumption control and troubleshooting.

To this end, this project will describe the current technologies related to the field of WSN and sensors, presenting applications that are currently deployed. It will be also exposed a real proposal submitted to a city council to deploy a pilot project in many of its streets.

Will be described the environment, both hardware and software, explaining the algorithms used for the associations between nodes, operating diagrams in the different phases of the nodes operation, and the coding of programs that are needed for proper system performance.

Finally, because the field of WSN is in constant evolution, will be presented different ideas to implement various improvements which could be deployed in the future, extending the range of applications to provide to end-users.

Keywords: node, sensor, wireless, lighting, streetlight, network, communications, presence, management, database, web.



Índice

1. INTRODUCCIÓN.....	1
1.1. Introducción.....	3
1.2. Estructura del documento	4
2. ESTADO DEL ARTE	5
2.1. ¿Qué es una WSN?.....	7
2.2. Características de una WSN	7
2.3. Campos de aplicación de una WSN.....	8
2.3.1. Salud y deporte	9
2.3.2. Medio ambiente.....	11
2.3.3. Automoción.....	12
2.3.4. Domótica	13
2.3.5. Medidores de Lectura Remota	14
2.3.6. SmartGrids	15
2.3.7. Tráfico	15
2.3.8. Vía pública.....	16
3. WSN: Wireless Sensor network	19
3.1. Elementos de una WSN.....	21
3.1.1. Sensores	21
3.1.2. Nodos (motas).....	22
3.1.2.1. Mercado	23
3.1.3. Gateways y sumideros	32
3.2. Tecnologías de comunicación	32
3.2.1. ZigBee.....	33
3.2.1.1. Nivel Físico (PHY).....	34
3.2.1.2. Nivel de Control de Acceso al Medio (MAC).....	35
3.2.1.3. Nivel de red (NWK)	35
3.2.1.4. Nivel de Aplicación (APL)	37
3.2.1.5. Versiones de ZigBee	37
3.2.2. Bluetooth Low Energy.....	38
3.3. Gestión del conocimiento. Almacenamiento distribuido.	39
4. Propuesta real de despliegue	43



4.1. Introducción	45
4.2. Situación actual de la red de alumbrado público	45
4.2.1. Características técnicas de las lámparas según potencia	46
4.2.2. Potencia instalada	46
4.2.3. Horas de funcionamiento	47
4.2.4. Consumo eléctrico	48
4.3. Propuesta de Proyecto Piloto	49
4.3.1. Objetivo	50
4.3.2. Zona de actuación	50
4.3.3. Luminarias actuales	51
4.3.4. Sustitución de luminarias	52
4.3.4.1. Inducción Magnética	52
4.3.4.2. LED	54
4.3.5. Comparativa de consumo eléctrico	55
4.3.5.1. Estudio del consumo eléctrico en la totalidad de Pedro Muñoz	56
4.3.5.1.1. Inducción Magnética	56
4.3.5.1.2. LED	59
4.3.5.1.3. Conclusiones sobre la sustitución de bombillas	62
4.3.5.2. Estudio del consumo eléctrico en el Proyecto Piloto	62
4.3.5.2.1. Inducción Magnética	62
4.3.5.2.2. LED	63
4.3.5.2.3. Conclusiones	64
4.3.6. Red Inalámbrica de Nodos con Sensores	64
4.3.6.1. Sensores	65
4.3.6.1.1. Iluminación	65
4.3.6.1.2. Movimiento	66
4.3.6.2. Funcionamiento de la infraestructura	69
4.3.7. Presupuesto	73
4.3.7.1. Presupuesto de los elementos individualizados	73
4.3.7.1.1. Nodos	73
4.3.7.1.2. Luminarias	76
4.3.7.2. Presupuesto total implementación completa del alumbrado	76
4.3.7.4. Presupuesto total implementación Proyecto Piloto	78
4.3.7.5 Mano de obra	78



5.3.7.4 Comentarios sobre el presupuesto	79
5. Desarrollo Experimental	81
5.1. Introducción	83
5.2. Elementos	84
5.2.1. Máquina Virtual	86
5.2.2. Software JAVA.....	86
5.2.3. Base de Datos.....	91
5.2.4. Gestor Web	92
5.2.5. APIs de Google	92
5.3. Escenario y Objetivos	93
5.3.1. Objetivos	93
5.3.2. Escenario	94
5.4. Funcionamiento	95
5.4.1. Visión General	95
5.4.2. Puesta en marcha.....	96
5.4.2.1. Asignación de identificadores	109
5.4.2.2. Estructura de la Base de Datos	112
5.4.2.3. Configuración de la Basestation para el acceso a la Base de Datos	114
5.4.2.4. Algoritmo de selección de vecinos.....	116
5.4.3. Operativa de las SunSPOT	138
5.4.3.1. Detección de presencia.....	139
5.4.3.2. Envío de actualizaciones a la Basestation	143
5.4.3.2.1. Proceso seguido en el nodo	143
5.4.3.2.2. Proceso seguido en la Basestation.....	145
5.4.3.3. Recepción de avisos de presencia.....	146
5.4.4. Herramienta de monitorización.....	152
5.5. Pruebas realizadas	154
6. Conclusiones y desarrollos futuros	161
6.1 Trabajos futuros	164
7. Referencias Bibliográficas	165
8. Anexos.....	171



Índice de Figuras

Figura 1. Wireless ECG Sensor [3]	9
Figura 2. Sensor Nike +.....	10
Figura 3. miCoach Pacer de Adidas	10
Figura 4. miCoach Zone de Adidas	10
Figura 5. Captura de la aplicación miCoach Mobile.....	10
Figura 6. Sensor de Humedad [7].....	11
Figura 7. Sensor de Temperatura [8]	11
Figura 8. Sensor de Intensidad de luz [9].....	12
Figura 9. Sensor UV [10].....	12
Figura 10. Sensor de Gases [11].....	12
Figura 11. Esquema de la estructura general del sistema. Imagen extraída de [12].....	13
Figura 12. Foto del Hogar Digital de la EUITT.	14
Figura 13. Medidor de Lectura Remota de Agua [14].....	15
Figura 14. Esquema de infraestructura inteligente en carretera. Imagen extraída de [17]	16
Figura 15. Sensor incrustado en el asfalto [19].....	17
Figura 16. Sensor a nivel de calle [19].....	17
Figura 17. Esquema de Monitorización de humedad del suelo IMMS. Imagen extraída de [20].	18
Figura 18. Sustainable City Light de Philips. Figuras extraídas de [22]	18
Figura 19. Logo de ARDUINO	23
Figura 20. Arduino UNO	23
Figura 21. Arduino FIO	24
Figura 22. Arduino Ethernet Shield. Imagen extraída de [30]	25
Figura 23. SunSpot	26
Figura 24. Logo de Libelium	27
Figura 25. Placa base de WaspMote.....	27
Figura 26. Módulo GSM/GPRS para WaspMote de Libelium.	30
Figura 27. Módulo GPS para WaspMote de Libelium	30
Figura 28. Logo de Sensinode	31
Figura 29. Pila de Protocolos de Zigbee (extraída de [35])	33
Figura 30. Características de transmisión según la banda de frecuencia (Tabla extraída de [36])	34
Figura 31. Topologías de Red en Zigbee. Imagen extraída de [37].....	36
Figura 32. Zona de actuación del proyecto piloto presentado.....	51
Figura 33. Luminaria tipo “Fernandino”	51
Figura 34. Sensor de Iluminación.....	66
Figura 35. Módulo con sensor PIR	67
Figura 36. Simulación en 3D de la zona de detección de los sensores PIR.....	68
Figura 37. Vista aérea de la zona de detección de los sensores PIR.....	68
Figura 38. Recreación en 3D del funcionamiento de la instalación propuesta.	70
Figura 39. Recreación en 3D del funcionamiento de la instalación propuesta	71
Figura 40. Recreación en 3D del funcionamiento de la instalación propuesta.	72
Figura 41. Esquema general del sistema.	84



Figura 42. Esquema general del sistema implementado para la realización de las pruebas experimentales.	85
Figura 43. Captura de Sun SPOT Manager Tool.....	87
Figura 44. Captura programa Solarium	88
Figura 45. Estructura de una Sun SPOT (Imagen extraída de	89
Figura 46. Comparativa de la máquina virtual Squawk de JAVA, con una máquina virtual estándar de JAVA (Imagen extraída de)	90
Figura 47. Relaciones de los elementos JAVA con la Base de Datos	91
Figura 48. Localización sobre las que se han realizado las simulaciones	95
Figura 49. Diagrama de funcionamiento de la lectura de los sensores	100
Figura 50. Formato del datagram de Solitud de Información	104
Figura 51. Formato del datagram de envío de Información.....	104
Figura 52. Formato datagram recepción de datos	105
Figura 53. Diagrama del proceso de Puesta en Marcha de la Sun SPOT.....	109
Figura 54. Zona de la localidad de Pedro Muñoz.	110
Figura 55. Proceso de asignación de IDs	110
Figura 56. Proceso de asignación de IDs	111
Figura 57. Proceso de asignación de IDs	111
Figura 58. Proceso de asignación de IDs	112
Figura 59. Estructura de la Base de Datos.....	113
Figura 60. Algoritmo de selección de vecinos.	117
Figura 61. Algoritmo de selección de vecinos.	118
Figura 62. Algoritmo de selección de vecinos.	121
Figura 63. Algoritmo de selección de vecinos.	123
Figura 64. Algoritmo de selección de vecinos	124
Figura 65. Algoritmo de selección de vecinos	126
Figura 66. Algoritmo de selección de vecinos	128
Figura 67. Algoritmo de selección de vecinos	130
Figura 68. Algoritmo de selección de vecinos	131
Figura 69. Algoritmo de selección de vecinos	134
Figura 70. Algoritmo de selección de vecinos	136
Figura 71. Diagrama del proceso operativo del nodo	140
Figura 72. Formato del Datagram enviado a la Basestation	143
Figura 73. Diagrama del proceso de recepción de avisos de los nodos vecinos	148
Figura 74. Captura de la página web principal	153
Figura 75. Captura de la página web específica de cada farola	154
Figura 76. Despliegue de Virtual SPOTs en Solarium	155
Figura 77. Resultados de la Prueba 1	156
Figura 78. Resultados de la Prueba 2	157
Figura 79. Resultados de la Prueba 3	157
Figura 80. Resultados de la prueba 4	158
Figura 81. Resultados de la prueba 5	159



Índice de Tablas

Tabla 1. Características Técnicas de Arduino UNO	24
Tabla 2. Características Técnicas de Arduino FIO.	25
Tabla 3. Características Técnicas de SunSpot de Oracle.....	26
Tabla 4. Características técnicas de WaspMote de Libelium.....	28
Tabla 5. Modelos del módulo ZigBee.....	29
Tabla 6. Características técnicas módulo ZigBee	29
Tabla 7. Características del módulo GSM/GRPS	30
Tabla 8. Características del Módulo GPS para Wasmote de Libelium	31
Tabla 9. Características técnicas de NanoStack 2.0 de Sensinode.....	32
Tabla 10. Comparativa entre distintas versiones de ZigBee.....	38
Tabla 11. Distribución de lámparas en el municipio de Pedro Muñoz.	46
Tabla 12. Potencia total instalada en el municipio de Pedro Muñoz.	47
Tabla 13. Horas de funcionamiento del alumbrado público.....	48
Tabla 14. Tarifaciones.....	49
Tabla 15. Comparativa entre bombillas de INDUCCIÓN y HALOGENURO METÁLICO	53
Tabla 16. Características de una lámpara de Inducción	54
Tabla 17. Características de una lámpara LED de 70 W.....	55
Tabla 18 Estudio de Consumo Eléctrico y Coste Económico en la instalación eléctrica y comparativa con INDUCCIÓN	57
Tabla 19. Estudio de Consumo Eléctrico y Coste Económico en la instalación eléctrica y comparativa con LED	60
Tabla 20. Ahorro anual en el consumo energético según el tipo de bombilla	62
Tabla 21. Estudio de Consumo Eléctrico y Coste Económico en el Proyecto Piloto y comparativa con INDUCCIÓN.....	63
Tabla 22. Estudio de consumo eléctrico y coste económico en el Proyecto Piloto y comparativa con LED.....	63
Tabla 23. Comparativa de ahorro entre bombillas.	64
Tabla 24. Presupuesto por nodo con sensores	73
Tabla 25. Presupuesto por nodo sin sensores	74
Tabla 26. Presupuesto del material complementario necesario para la instalación.....	75
Tabla 27. Presupuesto estimado del cambio a lámparas de Inducción.....	76
Tabla 28. Presupuesto estimado del cambio a lámparas LED	76
Tabla 29 Presupuesto Total para la implementación del proyecto en la red de alumbrado completa	77
Tabla 30. Presupuesto Total para la implementación del Proyecto Piloto.....	78
Tabla 31. Presupuesto mano de obra	79



1

INTRODUCCIÓN



Jorge Gálvez Alfaro

Desarrollo de sistema de gestión de alumbrado público mediante WSN



POLITÉCNICA



1.1. Introducción

Durante los últimos años, los avances tecnológicos se han producido de forma vertiginosa. Gracias a ello, actualmente se puede tener un control prácticamente total de todo, desde los elementos del hogar, hasta la humedad de la tierra de una cosecha.

Tales avances han permitido desarrollar y evolucionar las Redes de Sensores Inalámbricos (WSN), gracias a las cuales, es posible monitorizar una gran cantidad de variables, además de tener un control a distancia del funcionamiento de toda clase de artilugios. Gracias a su tamaño, o a su alimentación por batería, entre otras características, dichas redes se pueden desplegar en cualquier lugar, por recóndito que se encuentre.

¿Por qué no aprovechar las posibilidades que ofrece para sacarle provecho?

En este Proyecto Fin de Carrera se crea y desarrolla una aplicación para estas redes de Sensores Inalámbricos: poder gestionar el funcionamiento de una red de alumbrado público, consiguiendo con ello, además de tener una visión actualizada del funcionamiento del sistema, un ahorro energético gracias a la modificación, según diferentes valores, de la intensidad de trabajo de una farola, o un ahorro económico. Gracias al control de la intensidad de funcionamiento, se conseguirá que el consumo sea el idóneo en cada momento, es decir, si hay personas o vehículos que necesitan luz, el funcionamiento será normal, y el resto de tiempo, la farola trabajará a una intensidad mínima.



1.2. Estructura del documento

Para comprender en qué consiste una WSN, de qué elementos está compuesta, cómo funciona, y qué usos se están dando actualmente, se realizará un estudio en los **capítulos 2. ESTADO DEL ARTE y 3. WSN: Wireless Sensor network.**

Debido a que esta solución podría tener un despliegue real relativamente sencillo, en el **capítulo 4. Propuesta real de despliegue,** se expone el proyecto piloto que se redactó y presentó al Excelentísimo Ayuntamiento de Pedro Muñoz (Ciudad Real) para su despliegue en dicha localidad.

Ya en el **capítulo 5. Desarrollo Experimental,** se mostrará todo el despliegue técnico para montar el sistema: medios técnicos utilizados, diagramas de funcionamiento, codificaciones, así como las pruebas realizadas.

Para finalizar, en el **capítulo 6. Conclusiones y desarrollos futuros** se expondrán las conclusiones a las que se ha llegado tras la realización de este PFC, así como posibles mejoras que se podrían implementar.



2

ESTADO DEL ARTE



Jorge Gálvez Alfaro

Desarrollo de sistema de gestión de alumbrado público mediante WSN



POLITÉCNICA



2.1. ¿Qué es una WSN?

Según Bang Wang [1], una Red de Sensores Inalámbricos se define como “[...] cientos o miles de nodos capaces de recoger, procesar y enviar muestras de un entorno, para así monitorizar ciertos fenómenos físicos o controlar ciertos objetos en un área determinada.”

De esta forma, una WSN será capaz de recoger diferentes variables, tales como luz, temperatura, humedad, presencia... Para conseguirlo, se necesitará que exista una colaboración entre ellos.

Así, los elementos que formarán una WSN serán los **sensores**, los cuales serán los encargados de recoger los datos, datos que provendrán del análisis de diferentes magnitudes físicas como la presión, temperatura, humedad,...

Si bien por regla general los sensores sólo se encargarán de recoger datos y enviarlos para su análisis, se puede dar el caso que sean los propios nodos los que analicen esa información y actuar según los resultados obtenidos.

2.2. Características de una WSN

Si se hace un repaso a las características de una WSN [1], es obligatorio hablar en primer lugar del tamaño. Si bien una WSN podrá estar formada desde por una cantidad muy pequeña de nodos hasta una cantidad de decenas de millones de ellos, el tamaño de la WSN será algo específico de la aplicación final para la que esté destinada la WSN a implementar y el despliegue que se vaya a realizar. Y es precisamente el hecho de que estas redes puedan estar formadas por una gran cantidad de nodos la que lleve a hablar de que esos nodos no puedan volver a ser revisados físicamente para, por ejemplo, cambiarles la batería, en caso de que la fuente de alimentación tenga un tiempo de



vida determinado y cuando éste finalice, el nodo deje de ser útil. Es por ello por lo que será necesario un completo y preciso análisis de la red a desplegar y sus características para evitar problemas como el citado con la fuente de energía. Más adelante se tratará el problema del consumo energético en las WSN.

Otra de las características primordiales en una WSN son los propios nodos y sus características técnicas. Estos nodos, por lo general, suelen ser dispositivos con unas características técnicas muy bajas, lo que hace que el procesamiento que realicen de los datos obtenidos no pueda ser muy excesivo, ya que de no ser así y poder realizar un gran procesamiento de datos, aparecería la necesidad de tener una fuente de alimentación mucho mayor, además de un tamaño mayor del nodo, lo que provoca la necesidad de unos dispositivos más grandes, algo que podría interferir en un despliegue donde se necesite un mínimo tamaño de los nodos.

Además, dependiendo del despliegue, la situación puede hacer que los nodos se encuentren en constante movimiento (por ejemplo, si se quisieran monitorizar los animales de una manada). Es por ello por lo que surge la necesidad de que la red sea capaz de adaptarse a los cambios que se den en la topología y que busquen la mejor solución para conseguir obtener los resultados de forma satisfactoria [1].

2.3. Campos de aplicación de una WSN

Una de las grandes ventajas de las Redes de Sensores Inalámbricos es la **gran cantidad de campos de aplicación** en las que pueden ser desplegadas este tipo de redes. Los campos de aplicación pueden ir desde lo más sencillo hasta lo más complejo: desde la monitorización de pacientes médicos, hasta el control de tráfico en una ciudad pasando por el control de animales a la domotización de un edificio.

2.3.1. Salud y deporte

La monitorización de las constantes vitales en pacientes de un hospital o personas que necesiten en su hogar seguir un seguimiento especial es un campo que está creciendo en los últimos años. Con las Redes Inalámbricas de Sensores de Área Corporal [2], se puede conseguir una monitorización completa de pacientes que así lo requieran (Figura 1) pudiéndose ampliar al campo del deporte, ya que permitiría un exhaustivo control de las constantes vitales de un deportista tanto durante competiciones como durante el día a día en el desarrollo de su actividad profesional, permitiendo así llevar un control exhaustivo de las estadísticas de los entrenamientos para su posterior análisis, o incluso para un análisis en el momento a través de un “Entrenador Virtual” que pueda ir guiando al deportista sobre sus prácticas diarias.

Algunos ejemplos actuales son el Sensor Nike+ [3] de Nike que colocado en un lugar especial debajo de la plantilla de la zapatilla y conectado al Ipod permite ir controlando la evolución del ejercicio (Figura 2), o la gama MiCoach que ofrece Adidas [4] y el cual ofrece una gama de productos para el control de actividades deportivas formada por el “miCoach PACER”(Figura 3), o “miCoach Zone”(Figura 4) productos que miden nuestras constantes vitales y la velocidad de carrera, o “miCoach Mobile”(Figura 5) que hace las veces de Entrenador Virtual y que guía al deportista a lo largo del entrenamiento.



Figura 1. Wireless ECG Sensor [3]



Figura 2. Sensor Nike +



Figura 3. miCoach Pacer de Adidas



Figura 4. miCoach Zone de Adidas

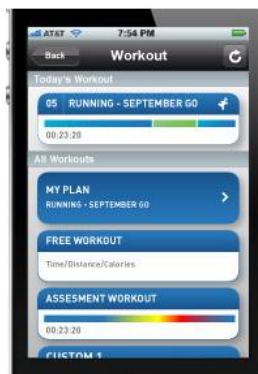


Figura 5. Captura de la aplicación miCoach Mobile

2.3.2. Medio ambiente

La monitorización de entornos medioambientales es sin duda el campo de aplicación clásico de las WSN. Desplegando una red de sensores inalámbricos en una determinada zona, se conseguirá monitorizar diversos parámetros, tales como temperatura, humedad, concentración de CO₂,... y obtener resultados que permitan realizar un análisis de cómo afectan al entorno (animales, flora,...) [5].

A continuación se citan algunos de los parámetros que se podrían analizar con una WSN:

- Presión Atmosférica [6].
- Humedad.



Figura 6. Sensor de Humedad [7]

- Temperatura.



Figura 7. Sensor de Temperatura [8]

- Intensidad de la luz.



Figura 8. Sensor de Intensidad de luz [9]

- Radiación Ultravioleta.



Figura 9. Sensor UV [10]

- Gases (CO_2 , NO_2).



Figura 10. Sensor de Gases [11]

2.3.3. Automoción

Múltiples son las aplicaciones en el mundo de la automoción sobre la que se pueden implantar redes de sensores inalámbricos, pudiendo conseguir un análisis exhaustivo del vehículo y mantener controlado en todo momento la situación de las diversas partes, como por ejemplo, la presión del aire en los neumáticos, el nivel de gasolina, detección de

presencia de los ocupantes del vehículo o automatización de los limpiaparabrisas entre otros muchos.

Un ejemplo sobre esta tecnología es el expuesto por Jiaming Zhang, Quan Liu, Yi Zhong en la International Conference on MultiMedia and Information Technology [12] en el cual proponen un sistema basado en WSN por el cual es posible tener una supervisión constante de la presión de los vehículos y así poder avisar al conductor en caso de que se produzca alguna anomalía que le pueda poner en peligro.

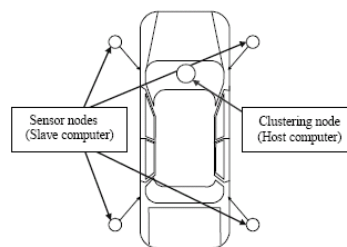


Figura 11. Esquema de la estructura general del sistema. Imagen extraída de [12]

2.3.4. Domótica

Gracias a las redes de sensores inalámbricos, será posible una automatización de los diferentes elementos que conforman una casa o un edificio y permitirá un control integral de los mismos.

A través de los diferentes sensores se podrán monitorizar diferentes factores para que, a raíz de los resultados o de la propia necesidad del usuario, se pueda modificar el estado de los elementos controlados, como por ejemplo:

- Persianas.
- Temperatura de la caldera.
- Cámaras de seguridad.
- Bombillas.

- Un ejemplo muy completo de casa domotizada se puede encontrar en la Escuela Universitaria de Ingeniería Técnica de Telecomunicación, situada en el Campus Sur de la Universidad Politécnica de Madrid [13], donde se puede encontrar una construcción de un hogar completamente domotizado y adaptado a necesidad especiales.



Figura 12. Foto del Hogar Digital de la EUITT.

2.3.5. Medidores de Lectura Remota

Una de las acciones que se está llevando a cabo en los últimos años en cuanto a la toma de lecturas de los contadores de agua es su sustitución por dispositivos de medida automática, los cuales, haciendo uso de tecnología inalámbrica, son capaces de transmitir la información de lectura sin necesidad de que una persona pase in-situ por el contador. Una red de estos sensores permitiría un control sobre el consumo de agua en determinadas zonas residenciales de forma exhaustiva, tanto a la hora del cobro correspondiente, como de realizar una gestión eficiente en caso de gasto excesivo.



Figura 13. Medidor de Lectura Remota de Agua [14]

2.3.6. SmartGrids

SmartGrids European Technology Platform for Electricity Networks of the Future [15] es un proyecto de la Unión Europea que comenzó en 2005 y que tiene como objetivo dotar de inteligencia a las diferentes redes eléctricas para tener así un control del suministro y de la entrega de energía en todas sus fases, generación, transmisión, distribución y consumo). De esta forma se consigue llevar un control para evitar abusos en el consumo, o poder detectar averías en caso de detectarse datos incoherentes.

2.3.7. Tráfico

Las WSN pueden permitir un control del tráfico en determinadas zonas. A través de la instalación de sensores en carreteras o en calles, se podrá obtener un informe de la situación del tráfico en una determinada zona y que así los conductores puedan consultar dicha información y por tanto, poder tomar una ruta alternativa. En [16] se propone un modelo de un Sistema que desarrolla esta idea.

Además del control del tráfico, sería posible una integración mayor a través de sensores que detectaran señales de tráfico y así avisar al conductor del vehículo de la velocidad permitida, así como la posibilidad de limitar la velocidad del coche de forma automática a esa velocidad, poder activar medidas de seguridad del vehículo como el control ESP en caso de que la humedad del asfalto supere un cierto límite, o el control de excesos de velocidad en los turismos [17].

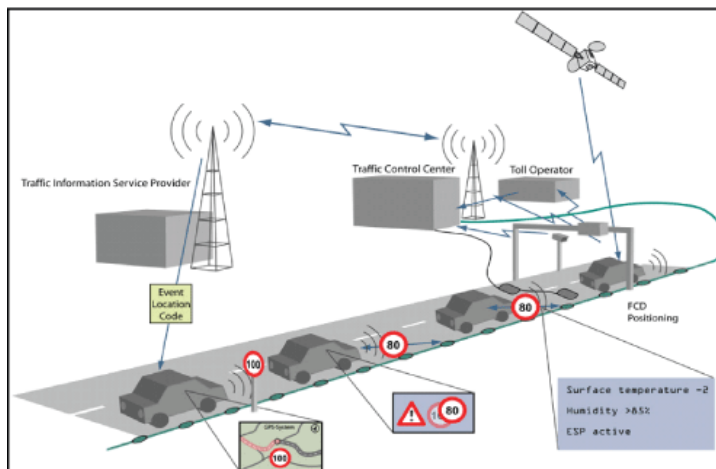


Figura 14. Esquema de infraestructura inteligente en carretera. Imagen extraída de [17]

2.3.8. Vía pública

A través de WSN es posible un control de zonas urbanas:

- **Aparcamiento.**
 - **Aparcamientos de pago:** por un lado, a través de sensores inalámbricos situados en las plazas de aparcamiento, se puede llevar un control de las plazas libres y ocupadas de un recinto. De esta forma se conseguiría un control de las plazas libres y una búsqueda más rápida de una plaza libre. Si en los vehículos existen también sensores inalámbricos, se podría realizar una comunicación que permitiera el pago automático de esa plaza de aparcamiento, ya que

estableciendo una comunicación con el sensor que está instalado en cada plaza de aparcamiento, se podría calcular el tiempo que ha estado ocupando dicha plaza y realizar el pago correspondiente. Se puede ver un ejemplo de esta aplicación en [18].

- **Aparcamientos al aire libre.** A través de sensores como los mostrados en la Figura 15 y en la Figura 16 , se podrá llevar un control de las plazas de aparcamiento libres en una determinada zona, pudiendo por un lado permitir a las autoridades pertinentes actuar en caso de aparcamiento indebido por zona prohibida o por incumplimiento de horario, o informar al usuario a través de una aplicación de las plazas de aparcamiento libre en su zona.



Figura 15. Sensor incrustado en el asfalto [19]



Figura 16. Sensor a nivel de calle [19]

- **Mantenimiento de zonas verdes.** A través de sensores de humedad del suelo instalados en zonas verdes como jardines o

parques, se podrá controlar la humedad del terreno y poder actuar al descender de un determinado nivel, activando el sistema de riego instalado. En la Figura 17 se puede observar el esquema que seguiría uno de estos sistemas, en concreto el Isaacs Moisture Monitoring System (IMMST[™]) [20]

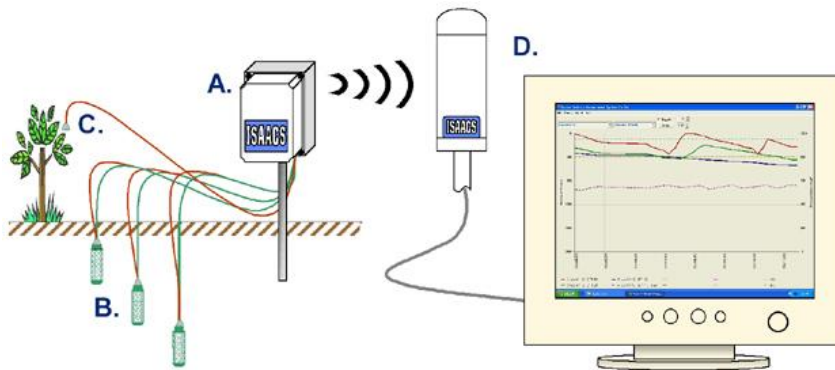


Figura 17. Esquema de Monitorización de humedad del suelo IMMS. Imagen extraída de [20].

- **Alumbrado Público.** Otra de las aplicaciones para zonas urbanas puede ser la gestión del alumbrado público, yendo más allá del encendido y apagado a una cierta hora. A través de sensores, se podría detectar la presencia de transeúntes en una determinada zona y por tanto actuar sobre el alumbrado consiguiendo un ahorro energético [21].

En la Figura 18 se puede observar otro sistema de este tipo ideado por Philips por el cual las farolas instaladas en la calle se adaptan a diversos factores para recoger energía o emitir luz con una intensidad variante.

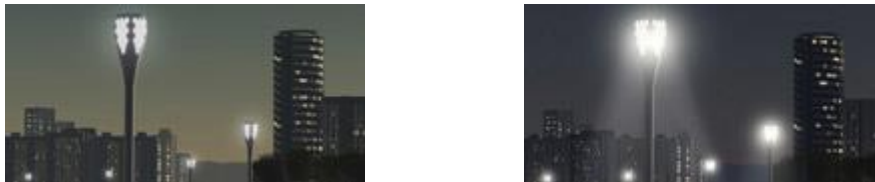


Figura 18. Sustainable City Light de Philips.
Figuras extraídas de [22]



3

WSN: Wireless Sensor network



Jorge Gálvez Alfaro

Desarrollo de sistema de gestión de alumbrado público mediante WSN



POLITÉCNICA



3.1. Elementos de una WSN

Los elementos por los cuales están formada una WSN se pueden clasificar en 4 [23]:

- **Sensores:** Recogerán del medio diferente información (luz, humedad, temperatura, sonido,...) y la transformarán en señales eléctricas.
- **Nodos (Motas):** Son los elementos de la red que se encargan de recoger la información proporcionada por los sensores, tratarla según las necesidades de la red y transmitirlas a la estación base.
- **Estación Base:** Será el encargado de recoger toda la información que le proporcionen los diferentes nodos por los que estará formado la red.
- **Gateway:** Servirá de interconexión entre la red de sensores y una red basada en TCP/IP a través de la cual se podrá operar sobre las estaciones base y así analizar la información que han recibido.

3.1.1. Sensores

Los sensores serán los encargados de recoger, procedentes del medio en el que se encuentran, información de diversa índole para que los nodos (motas) en las que están integrados los analicen y actúen en consecuencia.

Numerosos son los tipos de nodos existentes:

- **Aceleración:** estos sensores miden la aceleración que tiene el objeto al que están unidos. De esta forma, este tipo de sensores puede actuar cuando se reciben un impacto superior a una determinada aceleración, medir vibraciones o conocer la inclinación.



- **Presencia:** estos sensores son capaces de detectar mediante tecnología de infrarrojos si algún cuerpo se interpone en el rango de acción del sensor.
- **Temperatura:** estos sensores son capaces de detectar la temperatura del ambiente en el que se encuentran.
- **Sonido:** sensores capaces de transformar una señal de audio en señal eléctrica para que el nodo la analice.
- Otros posibles sensores son: humedad del aire, fuerza, proximidad, gas,...

En [24] o [25] se pueden ver módulos de sensores para acoplar a las diferentes placas que ofrecen las empresas suministradoras de este tipo de elementos.

3.1.2. Nodos (motas)

Los nodos de una WSN tienen como objetivo el recolectar los datos que han tomado los sensores y enviar la información correspondiente a la estación base.

Los nodos se caracterizan por:

- Pequeño tamaño.
- Poca memoria (escasos Kilobytes).
- Poco procesados (unos cuantos MHz).
- Sistema de radio para envío de información de poco alcance.
- Fuente de alimentación que va desde una pila de botón hasta el aprovechamiento de la energía solar.

3.1.2.1. Mercado

A continuación se analizan diferentes nodos disponibles en el mercado.

- ARDUINO.

ARDUINO es, como definen en su página web, “una herramienta para hacer a los ordenadores capaces de sentir y controlar más cosas del mundo físico que tu ordenador de sobremesa” [26]. ARDUINO se caracteriza por ofrecer plataformas hardware abiertas con múltiples diseños.



Figura 19. Logo de ARDUINO

Las placas que a continuación se exponen, se completan con diferentes módulos que permiten funcionalidades como medición de las diferentes magnitudes o la comunicación entre nodos:

- **ARDUINO UNO** [27].

Arduino UNO es una placa con conexión USB, a través de la cual no solo se puede extraer información, sino que se puede alimentar.

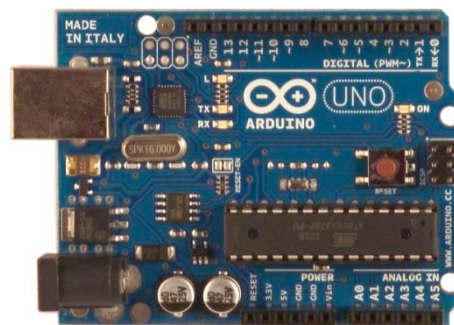


Figura 20. Arduino UNO

A continuación se muestran sus características técnicas más importantes:

Característica	Valor
Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

Tabla 1. Características Técnicas de Arduino UNO

- **ARDUINO FIO** [28]

Este modelo que ofrece ARDUINO, es el modelo más idóneo para las comunicaciones inalámbricas que incluye un Socket para una XBee radio, un conector para una batería LiPo y un circuito integrado para la carga de la batería.

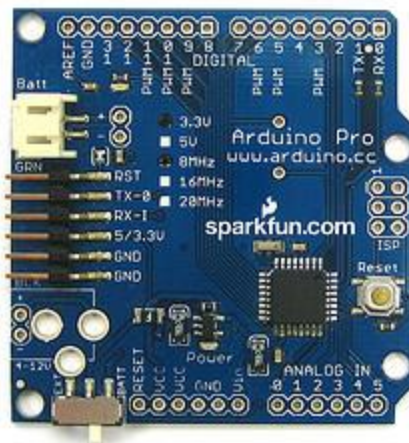


Figura 21. Arduino FIO

A continuación se muestran sus características técnicas más importantes:

Característica	Valor
Microcontroller	ATmega328P
Operating Voltage	3.3V
Input Voltage	3.35 -12 V
Input Voltage for Charge	3.7 - 7 V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	8
DC Current per I/O Pin	40 mA
Flash Memory	32 KB (of which 2 KB used by bootloader)
SRAM	2 KB
EEPROM	1 KB
Clock Speed	8 MHz

Tabla 2. Características Técnicas de Arduino FIO.

Estos son los principales productos que ofrece la marca Arduino. Existen muchos más productos de esta marca que, añadidos a las placas bases anteriormente expuestas o a otras de la misma marca, permiten añadir funcionalidades extra como por ejemplo la conexión entre nodos mediante ZigBee a una distancia hasta 30 metros en interior y 90 metros en el exterior [29], o la conexión a una red Ethernet para tener acceso a y desde Internet [30].



Figura 22. Arduino Ethernet Shield. Imagen extraída de [30]

- ORACLE

SunSPOT es la solución que ofrece Oracle para el mundo de las WSN, ofreciendo un producto que corre con Java y por tanto fácil de programar debido al conocimiento extendido de este lenguaje [31].



Figura 23. SunSpot

Sus principales características técnicas son las siguientes:

Característica	Valor
RAM Memory	1 Mb
Flash Memory	8 Mb
Clock Speed	133,3248 MHz
Serial Lines	3
Digital Pins	4
Analog Lines	4
Sensors	Accelerometer, IR Receiver and IR Transceiver
Other	Lo-fi audio speaker, tricolor LED.

Tabla 3. Características Técnicas de SunSpot de Oracle

En el capítulo destinado al desarrollo experimental, se hablará con profundidad de las SunSPOT, ya que serán éstas las utilizadas para ello.

- **Libelium**

Libellium, tal y como definen en su web “diseña y fabrica **tecnología de hardware** para la implementación de **redes sensoriales inalámbricas** y **redes malladas** para que integradores de sistemas, ingenierías y consultorías puedan llevar a cabo soluciones fiables a usuarios finales en el mínimo tiempo de salida al mercado” [32].



Figura 24. Logo de Libelium

Su principal producto es el WaspMote, “dispositivo sensorial de bajo consumo para la creación de redes sensoriales inalámbricas que integra más de 50 sensores diferentes” [32].

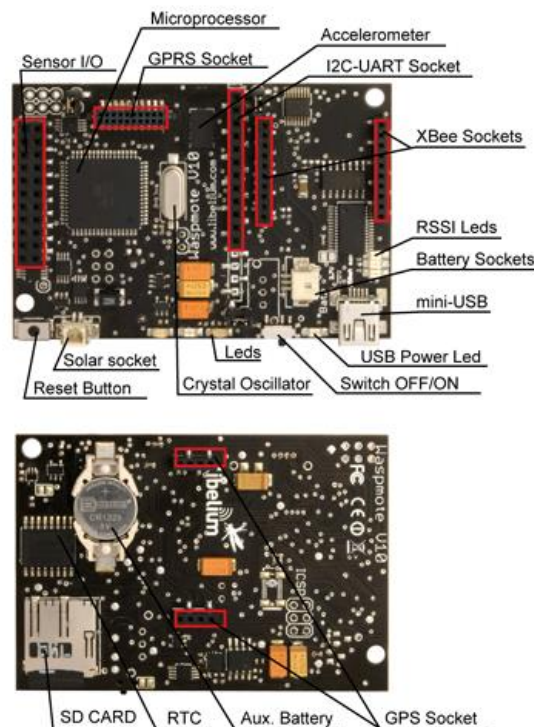


Figura 25. Placa base de WaspMote



Entre sus características técnicas cabe destacar las mostradas en la siguiente tabla:

Característica	Valor
Microcontrolador:	ATmega1281
Frecuencia:	8MHz
SRAM:	8KB
EEPROM:	4KB
FLASH:	128KB
SD Card:	2GB
Peso:	20gr
Dimensiones:	73.5 x 51 x 13 mm
Rango de Temperatura:	[-20°C, +65°C]
Reloj:	RTC (32KHz)
Consumo ON:	9mA
Consumo Sleep:	62uA
Consumo Deep Sleep:	62uA
Consumo Hibernate:	0,7uA
Entradas	Analógicas, 8 Digitales (I/O), 1 PWM, 2 UARTs, 1 I2C, 1 USB
Tensión de Batería	3,3 V – 4,2 V
Carga USB	5 V – 100 mA
Carga Placa Solar	6 – 12 V – 240 mA
Tensión Batería auxiliar	3 V
Sensores incorporados	Temperatura y Acelerómetro

Tabla 4. Características técnicas de WaspMote de Libelium

A la placa base expuesta, se le pueden añadir diferentes módulos de comunicaciones, como pueden ser:

- **Módulo ZigBee:** Dependiendo del modelo elegido, se ofrecen unas características de transmisión:



Model	Protocol	Frequency	Tx power	Sensitivity	Range*
XBee-802.15.4	802.15.4	2.4GHz	1mW	-92dB	500m
XBee-802.15.4-Pro	802.15.4	2.4GHz	100mW	-100dBm	7000m
XBee-ZB	Zigbee-Pro	2.4GHz	2mW	-96dBm	500m
XBee-ZB-Pro	Zigbee-Pro	2.4GHz	50mW	-102dBm	7000m
XBee-868	RF	868MHz	315mW	-112dBm	40Km
XBee-900	RF	900MHz	50mW	-100dBm	10Km
XBee-XSC	RF	900MHz	100mW	-106dBm	24Km

Tabla 5. Modelos del módulo ZigBee

Las características del resto de elementos que conforman el módulo son:

Antenna	
2.4GHz:	2dBi/5dBi
868/900MHz:	0dBi/4.5dBi
Connector:	RPSMA
Encryption: AES 128b	
Signal Control: RSSI	
Standards	
XBee-802.15.4:	802.15.4 - Compliant
XBee-ZB:	ZigBee-Pro v2007 - Compliant

Tabla 6. Características técnicas módulo ZigBee

- **Módulo GSM/GPRS:** Gracias a este módulo se podrán realizar y recibir llamadas, enviar y recibir SMS, realizar conexión a internet , tener servicio de correo mediante SMTP y POP3 y cargar y descargar archivos mediante FTP.

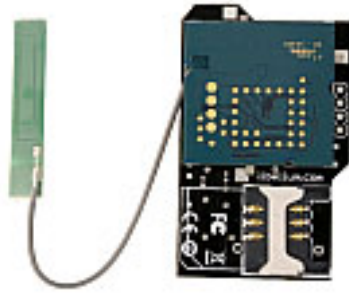


Figura 26. Módulo GSM/GPRS para WaspMote de Libelium.

Característica	Valor
Model:	Hilo (Sagem)
Quadband:	850MHz/900MHz/1800MHz/1900MHz
Txpower:	2W(Class 4) 850MHz/900MHz, 1W(Class 1) 1800MHz/1900MHz
Sensitivity:	-106dBm
Antenna connector:	UFL
External antenna:	OdBi

Tabla 7. Características del módulo GSM/GRPS

- **Módulo GPS:** Con este módulo GPS se podrán obtener datos en lo que respecta a la latitud, longitud, altitud, velocidad, dirección y fecha entre otros.



Figura 27. Módulo GPS para WaspMote de Libelium

- Las características técnicas de este módulo son:

Característica	Valor
Model:	A1084 (Vincotech)
Sensitivity in motion:	-159dBm
Sensitivity acquisition:	-142dBm
Hot-start boot time:	<1s
Warm-start boot time:	<32s
Cold-start boot time:	<35s
Antenna connector:	UFL
External Antenna:	26dBI

Tabla 8. Características del Módulo GPS para Waspote de Libelium

- Sensinode.

"Sensinode es pionero y líder mundial en tecnología y soluciones software para Internet of Things; la siguiente evolución de internet donde dispositivos de todas las clases y posibilidades podrán ser accedidos mediante Web Services.[...]. Nuestros principales mercados son la automatización de edificios, salud inalámbrica, logística [...]." [33]. Sensinode ofrece una amplia y completa oferta incluyendo nodos y gateways.



Figura 28. Logo de Sensinode

Su principal producto es **NanoStack 2.0** plataforma y radio independientes que forman un stack de comunicaciones para redes de sensores inalámbricas basadas en IP.

Sus características técnicas son:



2.4 GHz (CC2430) chip support
868/915 MHz (CC1110) chip support
32-64 kB of ROM typical (inc. application)
4-8 kB of RAM typical
6LoWPAN, IPv6, UDP, ICMPv6 standards
IEEE 802.15.4-2006 standard
NanoMesh™ routing
Network autoconfiguration
128-bit AES security support
Network processor and library versions

Tabla 9. Características técnicas de NanoStack 2.0 de Sensinode

3.1.3. Gateways y sumideros

Debido a la necesidad de disponer de los datos recogidos por los diferentes nodos de los cuales estará compuesta una WSN, será necesario disponer de **dispositivos que interconecten la red con el exterior**, para así poder ser accedidos a través de un ordenador o incluso desde la propia red Internet. Para ello, los Gateways servirán como puerta de enlace entre la WSN y el exterior y los cuales ofrecerán múltiples opciones para poder acceder a ellos: USB, Wifi, GSM/GPRS, Bluetooth...

3.2. Tecnologías de comunicación

En este punto se van a detallar las características de 2 tecnologías inalámbricas para la interconexión de los elementos que forman una Red de Sensores Inalámbricos: ZigBee y Bluetooth Low Energy.

3.2.1. ZigBee

ZigBee [34] es una serie de estándares dirigidos a conseguir ofrecer unos protocolos para la comunicación entre elementos de una red basada en tecnología inalámbrica buscando un bajo gasto y un bajo consumo de energía.

Zigbee opera sobre la banda de 2,4 GHz a nivel mundial y más específicamente, en la banda de 868 MHz a nivel europeo.

En la Figura 29 se puede observar la pila de protocolos que sigue ZigBee:

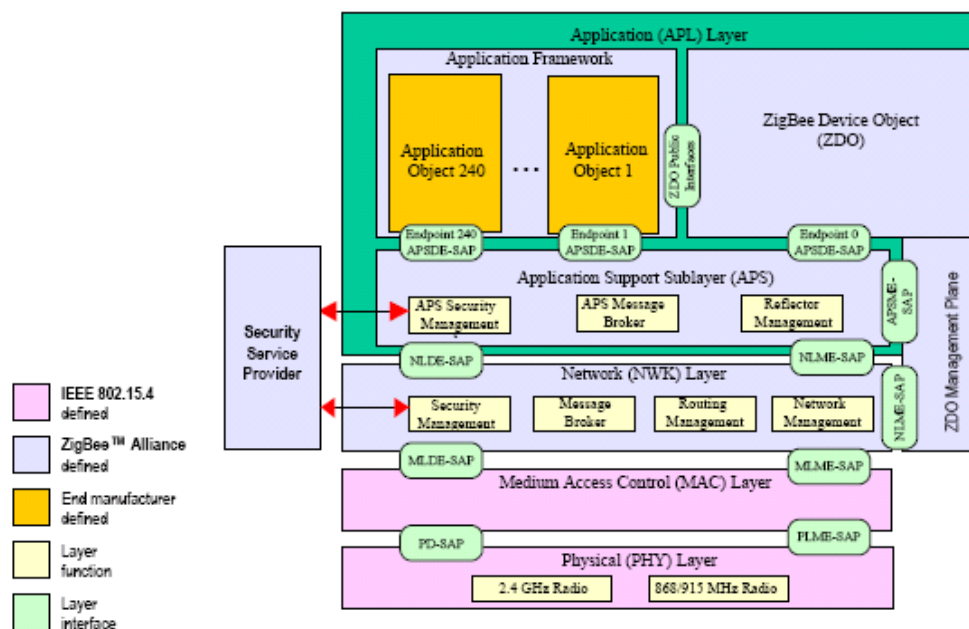


Figura 29. Pila de Protocolos de Zigbee (extraída de [35])

La pila de protocolos que sigue ZigBee se compone de 4 niveles principales [35]:

- **PHY:** Nivel Físico.
- **MAC:** Nivel de Control de Acceso al Medio.
- **NHK:** Nivel de Red.
- **APL:** Nivel de aplicación

Si bien los dos niveles más bajos (Nivel Físico y Nivel de Control de Acceso al Medio) están definidos en el estándar 802.15.4-2003, los otros



dos niveles (Nivel de Red y Nivel de Aplicación) están definidos en la Especificación ZigBee.

3.2.1.1. Nivel Físico (PHY)

Tal y como se ha comentado anteriormente, el Nivel Físico de la torre de protocolos de ZigBee se encuentra especificado en el Estándar IEEE 802.15.4-2003 [36].

El nivel físico se encargará de:

- Activar y desactivar el transceptor de radio.
- Detección de Energía en el canal actual.
- Indicación de la Calidad del Enlace para los paquetes recibidos.
- Selección de la frecuencia del canal.
- Transmisión y recepción de los datos.

Según la región del mundo donde se opere, se operará sobre una determinada frecuencia y con unas características de transmisión y recepción diferentes.

Tres son los posibles casos:

- A nivel mundial, en el rango de frecuencias de los 2,4 GHz.
- En Europa, en el rango de frecuencias de los 868 MHz.
- En América, en el rango de frecuencias de los 915 MHz.

PHY (MHZ)	Banda Frecuencia (MHZ)	Modulación	Tasa de bits (kbps)	Tasa de Símbolos (ksymbols /s)	Símbolos
868	868 - 868.6	BPSK	20	20	Binario
915	902 - 928	BPSK	40	40	Binario
2450	2400 - 2483.5	O-QPSK	250	62,5	16-ary Orthogonal

Figura 30. Características de transmisión según la banda de frecuencia (Tabla extraída de [36])



3.2.1.2. Nivel de Control de Acceso al Medio (MAC)

El Nivel MAC se encarga de manejar todos los accesos que se intentan realizar al nivel de radio físico, siendo el responsable de:

- Generar balizas de red si el dispositivo es el coordinador.
- Sincronización de las balizas.
- Soporte de asociación y disociación de Personal Area Networks (PAN).
- Soportar seguridad al dispositivo.
- Emplear CSMA-CD para acceder al canal.
- Mantener el "Tiempo de Slot Garantizado".
- Proveer de un enlace fiable entre entidades MAC pares.

3.2.1.3. Nivel de red (NWK)

El nivel de red (NWK) proporciona la funcionalidad necesaria para asegurar que el nivel MAC, definido en la IEEE 802.15.4-2003 sobre el cual está construido y del cual se ha hablado anteriormente, opere de forma correcta, además de ofrecer una interfaz de servicio adecuada para el nivel de aplicación (APL). Para realizar esa unión con el nivel de aplicación, NWK incluye dos entidades de servicio, las cuales proporcionarán las funcionalidades necesarias:

- **NWK Layer data Entity (NLDE)** que proporciona el servicio de transmisión de datos a través del SAP asociado, el NLDE-SAP.
- **NWK Layer Management Entity (NLME)**, que proporciona el servicio de administración a través del SAP asociado, el NLME-SAP.

El Nivel de red (NWK) de ZigBee soporta 3 tipos de topologías:

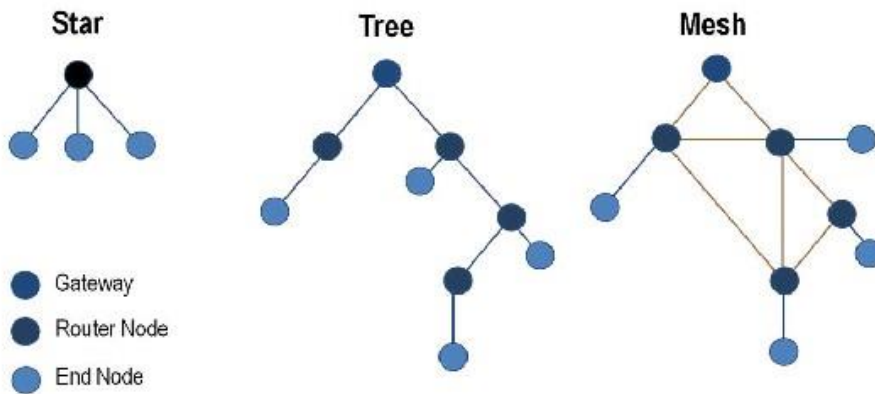


Figura 31. Topologías de Red en Zigbee. Imagen extraída de [37]

- **Topología en Estrella.** En este tipo de topología, la red es controlada por un único dispositivo denominado ZigBee Coordinator. Este tipo de dispositivo dentro de una red ZigBee es el encargado de inicializar y mantener los dispositivos en la red. Existirá únicamente un ZigBee Coordinator por red, y será éste precisamente el que la inicie. El resto de dispositivos que conforman esta topología, son los denominados *dispositivos finales*, y se conectarán directamente al ZigBee Coordinator.
- **Topología en Árbol.** Tal y como ya se ha comentado, el ZigBee Coordinator será el encargado de inicializar la red y de elegir los parámetros de red, pero lo que añade este tipo de topología (y la topología en malla que se explica en el siguiente punto) es la posibilidad de expandir la red a través de nodos que actúan de Router (ZigBee Routers). Éstos serán los encargados de mover y controlar el intercambio de mensajes entre los elementos de la red.
- **Topología en Malla.** Es la topología que más posibilidades ofrece, ya que a través de los ZigBee Routers explicados en la Topología en Árbol, se conseguirá una comunicación total entre nodos a través del “multi-salto” y así conseguir que cualquier nodo de la red pueda enviar datos a cualquier otro nodo.



3.2.1.4. Nivel de Aplicación (APL)

El nivel de Aplicación de la torre de protocolos ZigBee será el nivel desarrollado por los fabricantes para personalizar el dispositivo para la finalidad con la que ha sido concebido y el cual está compuesto de 3 secciones: Application Support Sublayer (APS), ZigBee Device Objects (ZDOs) y Application Framework.

3.2.1.5. Versiones de ZigBee

Principalmente existen tres versiones de Zigbee: **ZigBee 2006**, **ZigBee 2007** y **ZigBee Pro**. La elección de uno u otro dependerá de las características de la plataforma hardware sobre el que se va a implementar.

En la Tabla 10 se muestra una comparativa de las principales versiones:



Característica	ZigBee2006	ZigBee2007	ZigBeePro
Size in ROM/RAM	Smallest	Small	Bigger
Stack Profile	0x01	0x01	0x02
Maximum hops	10	10	30
Maximum nodes in network	31.101	31.101	65.540
Mesh networking	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Broadcasting	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Tree routing	<input type="checkbox"/>	<input type="checkbox"/>	-
Frequency Agility	-	<input type="checkbox"/>	<input type="checkbox"/>
Bandwidth Used By Protocol	Least	More	Most
Fragmentacion	-	<input type="checkbox"/>	<input type="checkbox"/>
Multicasting	-	-	<input type="checkbox"/>
Source Routing	-	-	<input type="checkbox"/>
Symmetric Links	-	-	<input type="checkbox"/>
Standard Security (AES 128 bit)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
High Security	-	-	<input type="checkbox"/>

Tabla 10. Comparativa entre distintas versiones de ZigBee¹

3.2.2. Bluetooth Low Energy

Bluetooth Low Energy surge como parte del desarrollo de la especificación de Bluetooth 4.0 [38]. Gracias al diseño de Bluetooth Low Energy, el consumo de energía se reduce sustancialmente con respecto a las radios Bluetooth clásicas. Es por ello, por lo que esta tecnología está pensada para ser usada con baterías pequeñas, permitiendo que pueda pasar hasta 1 año sin necesidad de ser cargada.

¹ Tabla extraída de [51], Pág. 390.



Además de tener un consumo energético reducido, también se ve mermado su tamaño a la hora de ser desplegado y su coste. Todo ello hace que sea idóneo para aplicaciones con tasas de datos bajas.

3.3. Gestión del conocimiento. Almacenamiento distribuido.

Una Wireless Sensor Network ofrece la oportunidad de recoger una gran cantidad de datos del entorno en el cual está desplegada la red. Ahora bien, los datos por sí solos no significan nada si no existe la posibilidad de transformarlos en información. El objetivo es analizar tanto en el tiempo como en el espacio los datos recogidos por los diferentes sensores y ser capaces de extraer conclusiones reales y fidedignas sobre esos datos.

Es por ello por lo que aparece la necesidad de identificar esos datos con información adicional (fecha, hora, lugar, usuario), lo que comúnmente se denomina *metadata*, y que por el contrario, entorpece en una WSN ya que entra en conflicto con las capacidades de memoria y transmisión de los sensores inalámbricos. Es por esto por lo que será necesario que existan nodos intermedios que den contenido semántico a los datos provistos por los diferentes sensores que conforman la red.

Los datos recogidos por los sensores pueden ser enviados a sumideros o a nodos vecinos para su procesamiento, si bien es necesario que se intente minimizar el número de transmisiones para ahorrar energía y ancho de banda. Si bien controlar estos puntos permite un ahorro en los aspectos anteriormente citados, hay que tener en cuenta que es posible que no se recojan cambios en los datos percibidos, es decir, que haya medidas que se pierdan.



Para evitar esto, se abre la posibilidad de realizar una comparación de los datos previa al envío al sumidero para evitar errores o para asegurar la veracidad de la información y así enviar únicamente información relevante. Realizar esto supondría llevar a cabo un procesamiento en el propio nodo.

Si se analizan las redes de las cuales las aplicaciones no recogerán los datos en tiempo real, se llega a la conclusión de que los datos han de ser almacenados dentro de la red (por lo menos de forma temporal), hasta que algún observador los solicite.

De esta forma, se podrían clasificar las aplicaciones [39]:

- **Monitorización offline:** Se realiza un despliegue en el cual los sensores recogerán información sobre un determinado fenómeno cuyos datos serán recogidos y analizados a posteriori. Si bien los datos son almacenados continuamente, solo son leídos una vez.
 - **ZebraNet** [40] es un ejemplo de este tipo de red donde se puede observar una red de sensores encargada de la monitorización de vida salvaje y con la cual se pretende profundizar en asuntos como patrones de migración, estructuras sociales y modelos de movilidad. En esta aplicación, se adhieren sensores a los animales. Los científicos recogen los datos cuando conducen por la zona de la cual se quiere obtener información. La recogida de datos no sigue ningún patrón, siendo impredecible.
 - **Realidad Aumentada Dinámica.** En este tipo de aplicación, los sensores recogen información de eventos que van ocurriendo, siendo por tanto la información solicitada por los usuarios que se encuentran en la red, unas peticiones que pueden ser sobre datos actuales o históricos. En este tipo de



redes, se accede a la información en múltiples ocasiones, y por tanto la importancia de un dato variará con el tiempo.

En las aplicaciones descritas, la cantidad de datos a almacenar en la red puede variar, y por tanto, se buscarán protocolos de gestión del espacio de almacenamiento que controlen de forma eficiente el espacio disponible para poder seguir almacenando datos durante el máximo tiempo posible sin que se produzcan pérdidas de información que puedan resultar relevantes a la hora de tratar los resultados.

La gestión del almacenamiento de la información podría clasificarse de la siguiente forma:

- **Almacenamiento local:** Solución más simple en la cual cada nodo almacena su información localmente. Protocolo eficiente en cuanto al gasto energético ya que no precisa de realizar tareas de comunicaciones con otros nodos, si bien el gasto energético para el almacenamiento es alto debido a la gran cantidad de datos a almacenar. Este tipo de gestión resulta ineficiente ya que no existe colaboración entre los nodos, y por tanto existirá información replicada en nodos cercanos.
- **Colaboración:** Los nodos colaborarán entre sí, tanto para estimar la redundancia que se puede dar en los datos, así como el intercambio de datos para, por ejemplo, realizar balanceo de carga. Con esto conseguimos 2 beneficios:
 - **Es necesario almacenar menos datos.** Las medidas recogidas por nodos que se encuentran cerca suelen ser muy parecidos, es por ello por lo que entre nodos vecinos se podrán obviar recoger información para evitar redundancia en los datos recogidos.
 - **Balanceo de carga,** consiguiendo equilibrar la cantidad de datos almacenados por varios nodos ya sí evitar el colapso en alguno de ellos.



Cabe destacar el gasto energético que se produce al realizar el intercambio de datos entre los nodos que se encuentran colaborando para poder almacenar los datos, si bien este será reducido una vez se haya establecido la comunicación.

En [39], documento del cual se ha extraído la información de este sub-apartado, aparecen explicados algunos protocolos de almacenamientos basados en colaboración. Uno de ellos, el *Cluster Based Collaborative Storage (CBCS) protocol*, en el cual los nodos se organizan en clústeres, siendo uno de ellos el *clúster head*, encargado de recibir los datos de los nodos y de llevar a cabo la agregación de datos y almacenamiento.

Otro de estos protocolos es el *Coordinated Local Storage (CLS)*, en el cual los sensores se coordinan periódicamente para evitar redundancia innecesaria.



4

Propuesta real de despliegue



Jorge Gálvez Alfaro

Desarrollo de sistema de gestión de alumbrado público mediante WSN



POLITÉCNICA



4.1. Introducción

Con el fin de realizar un despliegue real de una red de sensores inalámbricos, **se propone al Excelentísimo Ayuntamiento de Pedro Muñoz realizarlo en dicha localidad.** En el presente apartado se expondrá la propuesta presentada, y en él, se realizará un estudio del consumo actual del alumbrado público en Pedro Muñoz, obteniendo así las características técnicas de los diferentes elementos que conforman el alumbrado público, así como la potencia instalada, las horas de funcionamiento y el consumo eléctrico.

Se presentará el Proyecto Piloto del cual era fin el documento presentado, donde se expondrán los **objetivos**, la **zona de actuación**, una **propuesta de sustitución de lámparas**, una **comparativa del consumo** actual con el consumo que se obtendría con la implantación del Proyecto Piloto, así como una **explicación de los diferentes elementos** que lo conforman, el **funcionamiento** y el **presupuesto** para el despliegue.

4.2. Situación actual de la red de alumbrado público

En esta sección se realiza una recopilación de los datos de consumo e instalación actual, extraídos del documento [41], facilitado por el Excelentísimo Ayuntamiento de Pedro Muñoz para la realización de este Proyecto Fin de Carrera.



4.2.1. Características técnicas de las lámparas según potencia

La distribución de las 2.396 lámparas instaladas en el municipio de Pedro Muñoz se queda de la siguiente forma:

Tipo Lámpara	Nº Lámparas	Potencia (W)
VSAP Ovoide	1552	100
VSAP Ovoide	466	150
VSAP Ovoide	82	250
VSAP Tubular	100	250
HM Ovoide	154	150
HM Tubular	42	125

Tabla 11. Distribución de lámparas en el municipio de Pedro Muñoz.

Se observa que la mayoría de las lámparas instaladas corresponden a **Vapor de Sodio de Alta Presión (VSAP) Ovoide**.

4.2.2. Potencia instalada

El consumo de potencia de las lámparas instaladas es el siguiente:



Tipo Lámpara	Nº Lámparas	Potencia (W) Lámpara	Potencia (W) Equipo	Potencia Total (W) Lámpara	Potencia Total (W) Equipo	Potencia Total (W)
VSAP Ovoide	1552	100	80	155200	124160	279360
VSAP Ovoide	466	150	120	69900	55920	125820
VSAP Ovoide	82	250	200	20500	16400	36900
VSAP Tubular	100	250	200	25000	20000	45000
HM Ovoide	154	150	120	23100	18480	41580
HM Tubular	42	125	100	5250	4200	9450

Tabla 12. Potencia total instalada en el municipio de Pedro Muñoz.

Por tanto, el total de Potencia instalada es de **538110 W**.

4.2.3. Horas de funcionamiento

Las horas de funcionamiento de la instalación de alumbrado público dependiendo del mes son:



Tipo Lámpara	Horas funcionamiento / día	Días / mes	Horas funcionamiento / mes
ENERO	14,4	31	446,4
FEBRERO	13,3	28	372,4
MARZO	12,1	31	375,1
ABRIL	10,8	30	324
MAYO	9,6	31	297,6
JUNIO	9	30	270
JULIO	9,2	31	285,2
AGOSTO	10,2	31	316,2
SEPTIEMBRE	11,5	30	345
OCTUBRE	12,8	31	396,8
NOVIEMBRE	14	30	420
DICIEMBRE	14,6	31	452,6

Tabla 13. Horas de funcionamiento del alumbrado público.

Los datos recogidos en la Tabla 13 son los obtenidos de la célula fotoeléctrica ubicada en los cuadros.

De esta forma, obtenemos que el total de horas al año de utilización de la instalación eléctrica asciende a **4301,3 horas**.

4.2.4. Consumo eléctrico

Atendiendo a los datos expuestos en los anteriores puntos, los kWh consumidos por las lámparas en un año son:

$$538110 \text{ W} * 4303,3 \text{ horas/año} = 2314572543 \text{ Wh/año} = \mathbf{2314572,543}$$

kWh/año

Las distintas tarifificaciones que UNIÓN FENOSA aplica al consumo eléctrico es:



Tipo Lámpara	€/kWh (Tarifas a 2008)
Tarifa Social	0,109612
1.0	0,087373
2.0.1	0,109612
2.0.2	0,107994
2.0.3	0,106888
3.0.1	0,107338
3.0.2	0,101941

Tabla 14. Tarifaciones

Calculando el promedio de dichas tarifas, obtenemos que el precio del kW/h es de **0,104394 €/KWH**

Al aplicar el precio medio del kWh consumido al total de kWh consumidos al año, obtenemos que el gasto económico originado por el consumo eléctrico de las farolas, asciende a **293.824,48 €**

Los datos económicos de consumo se expondrán más detalladamente en el apartado "4.3.5. Comparativa de consumo eléctrico"

4.3. Propuesta de Proyecto Piloto

En los siguientes puntos se desarrollará el Proyecto Piloto que se presentó al Excelentísimo Ayuntamiento de Pedro Muñoz para ser desplegado en la localidad.



4.3.1. Objetivo

El objetivo del documento presentado es realizar un Proyecto Piloto en parte del alumbrado público de la localidad de Pedro Muñoz, que permita desplegar una Red de Sensores Inalámbricos, la cual permitirá realizar una gestión automatizada de las farolas.

Esta red permitirá:

- **Regular el consumo** de energía de las bombillas, de acorde a 2 factores:
 - Iluminación.
 - Presencia.

Estos factores se desarrollarán en el punto "4.3.6. Red Inalámbrica de Nodos con Sensores"

- **Conocer en tiempo real el consumo de las farolas** involucradas en el despliegue de este Proyecto Piloto.

Estos factores provocarán un ahorro considerable en la factura energética debido a la regulación constante del consumo eléctrico de las farolas.

4.3.2. Zona de actuación

Si bien en el presente documento se realiza un análisis a nivel general del impacto que tendría la implementación de la propuesta realizada, el Proyecto Piloto que se ofrece a realizar abarca la primera parte de la Calle Amós Olivares en su inicio en la Plaza de España. El número de luminarias en las que se implementaría el piloto sería de 6, suficiente para poder mostrar las posibilidades del diseño que se presenta.



Figura 32. Zona de actuación del proyecto piloto presentado.

4.3.3. Luminarias actuales

El número de luminarias sobre las cuales se desarrollaría el Proyecto Piloto son 6, tal y como se ha desarrollado en el apartado anterior. Estas luminarias (Figura 33) son de tipo Fernandino y actualmente poseen bombillas de Halogenuro metálico de 150 W.



Figura 33. Luminaria tipo "Fernandino"



4.3.4. Sustitución de luminarias

4.3.4.1. Inducción Magnética

Debido a que las actuales luminarias cuentan con bombillas de Halogenuro metálico, el encendido no es instantáneo, necesitándose de un tiempo hasta que se consigue el completo encendido. Además, este tipo de bombillas son de tensión fija, por lo que no es posible una regulación de la intensidad. Por ello, se propone el cambio de estas bombillas por unas de **inducción magnética**.

Entre las características que hacen de este tipo de tecnología de iluminación la apropiada para este proyecto Piloto, se encuentran las siguientes:

- Encendido instantáneo
- Vida útil: 100.000 horas.
- Rendimiento Lumen/W: 80-90 L/W
- Coste mantenimiento: Muy bajo
- Regulación: hasta el 60 %

A continuación se muestra una tabla comparativa entre lámparas de Halogenuro Metálico e Inducción Magnética:



Factor	INDUCCIÓN	HM
Encendido	Instantáneo	5-10 Minutos
Re-encendido instantáneo	SÍ	NO
Vida útil hasta agotarse	100.000	12.000
Contenido Mercurio	2 mg amalgama	10-1000 mg
Rendimiento Lumen/W	80-90 L/W	50-80 L/W
Coste mantenimiento	muy bajo	alto
Temperatura en superficie	60-90 °C	>300 °C
Voltaje	110 a 270 V	180 a 235 V
Picos arrancada	NO	35%
Regulación	Hasta el 60%	Hasta el 25%

Tabla 15. Comparativa entre bombillas de INDUCCIÓN y HALOGENURO METÁLICO

Se observa como la opción de las bombillas de Inducción Magnética se postula como la idónea para realizar la sustitución de las actuales bombillas de Halogenuro metálico, debido principalmente a la posibilidad de regulación, encendido inmediato, vida útil, y sobre todo, consumo, ya que con una bombilla de 85 W (e incluso una de 55 W a una altura adecuada), se conseguirá la misma, o incluso más visibilidad que con una bombilla de 150 W de Halogenuro metálico.

Añadir además que aunque el consumo teórico de la bombilla de Halogenuro Metálico es de 150 W, en realidad éste asciende hasta los 188 W, mientras que la lámpara de Inducción Magnético, su consumo teórico es de 85 W, y el real es de 88 W, consiguiendo por tanto una diferencia en consumo máximo de 100 W.

Más concretamente, las características técnicas de las lámparas de Inducción Magnética que se proponen son:



INDUCCIÓN 85 W	
Consumo	85 W
Frecuencia	250
Lúmenes	6800
Eficiencia (Lm/W)	80
Índice Color	>80
Tª color °K	2700 / 4500 / 5200 / 6500
Voltaje	120 a 277
Corriente (A)	0.31
Frecuencia (Hz)	50 a 60
Factor de Potencia	0.95
Armónicos	2% a 5%
Consumo Total	89 W
Tª Trabajo	65°

Tabla 16. Características de una lámpara de Inducción

En apartados posteriores, se expondrán los precios de este tipo de lámparas y se realizará una comparativa del consumo entre las lámparas actualmente instaladas.

4.3.4.2. LED

El segundo tipo de bombillas sobre el cual se propone el cambio es bombillas de LED. Si bien su precio es mayor que las de Inducción Magnética expuestas en el punto anterior, el ahorro conseguido es mayor, debido principalmente a que la regulación de la intensidad es total, permitiendo por lo tanto ajustar más los valores con los que funcionará la farola.



Las características técnicas son las siguientes:

LED 70 W	
Consumo	74 W
Lúmenes	5100-6000lm
Eficiencia (Lm/W)	70 %
Índice Color	>85
Tª color °K	2700 / 4500 / 5200 / 6500
Voltaje	110 a 230V
Factor de Potencia	0.95
Armónicos	0
Consumo Total	74 W

Tabla 17. Características de una lámpara LED de 70 W

Además, estas bombillas emiten en los 360° y tienen una vida útil de 50.000 horas, lo que supondría un cambio cada 10 años aproximadamente.

En apartados posteriores, se expondrán los precios de este tipo de lámparas y se realizará una comparativa del consumo entre las lámparas actualmente instaladas.

4.3.5. Comparativa de consumo eléctrico

A continuación se expone la comparativa realizada en cuanto al consumo eléctrico actual, y el consumo que obtendríamos realizando el cambio a lámparas de Inducción Magnética y LED. En primer lugar se mostrarán los datos realizando el estudio sobre todas las farolas instaladas actualmente en Pedro Muñoz, y después se concretará al alcance del Proyecto Piloto que se presentó en la propuesta al Excmo. Ayuntamiento de Pedro Muñoz.



4.3.5.1. Estudio del consumo eléctrico en la totalidad de Pedro Muñoz

En este apartado se muestra el estudio de consumo teórico de toda la infraestructura de alumbrado público instalada en Pedro Muñoz. En primer lugar haremos un estudio realizando la integración de la red de sensores y un cambio en las lámparas actuales por lámparas de Inducción.

4.3.5.1.1. Inducción Magnética

A continuación se realiza un estudio energético y económico de la situación actual del alumbrado público en toda la localidad de Pedro Muñoz, y su equivalente sustituyendo las actuales bombillas por unas de Inducción Magnética:



Tipo lámpara	pto Luz	Potencia (W) / punto	% de funcionamiento			kWh / día	kwh/año	Consumo	Pot Instalación	Consumo Pot Inst kwh/año	Consumo Pot Inst kwh/año	Consumo	CONSUMO TOTAL
			100%	80%	60%								
Comparativa de 1 farola de Vapor de Sodio (VSAP) frente a 1 farola de LEDs+													
VSAP Ovoide	1	100	12	0	0	0	438	52,56 €	80	350,4	42,05 €	94,61 €	
VSAP Ovoide	1	150	12	0	0	1,8	657	78,84 €	120	525,6	63,07 €	141,91 €	
VSAP Ovoide	1	250	12	0	0	3	1095	131,40 €	200	876	105,12 €	236,52 €	
VSAP Tubular	1	250	12	0	0	3	1095	131,40 €	200	876	105,12 €	236,52 €	
HM Ovoide	1	150	12	0	0	1,8	657	78,84 €	120	525,6	63,07 €	141,91 €	
HM Tubular	1	125	12	0	0	1,5	547,5	65,70 €	100	438	52,56 €	118,26 €	
INDUCCIÓN	1	55	4	1	0	7	0,2868	104,682	10	43,8	5,26 €	17,82 €	
INDUCCIÓN	1	85	4	1	0	7	0,4428	161,622	12	52,56	6,31 €	25,70 €	
INDUCCIÓN	1	110	4	1	0	7	0,5728	209,072	15	65,7	7,88 €	32,97 €	
Comparativa de todas las luminarias existentes frente a luminarias LEDs+													
VSAP Ovoide	1552	100	12	0	0	0	1.862,40	679.776,00	80	1489,92	543820,8	146.831,62 €	
VSAP Ovoide	466	150	12	0	0	0	838,80	36.739,44 €	120	671,04	244929,6	66.130,99 €	
VSAP Ovoide	82	250	12	0	0	0	246,00	10.774,80 €	200	196,8	71832	19.394,64 €	
VSAP Tubular	100	250	12	0	0	0	300,00	109.500,00	200	240	87600	23.652,00 €	
HM Ovoide	154	150	12	0	0	0	277,20	12.141,36 €	120	221,76	80942,4	21.854,45 €	
HM Tubular	42	125	12	0	0	0	63,00	2.759,40 €	100	50,4	18396	4.966,92 €	
TOTALES							3.587,40	1.309.401,00		2.869,92	1.047.520,80	282.830,62 €	
INDUCCIÓN	1552	55	4	1	0	7	445,11	162.466,46	10	186,24	67977,6	27.653,29 €	
INDUCCIÓN	662	85	4	1	0	7	293,13	106.993,76	12	95,328	34794,72	17.014,62 €	
INDUCCIÓN	182	110	4	1	0	7	104,25	38.051,10	15	32,76	11957,4	6.001,02 €	
TOTALES							738,25	269.460,23		95,33	34.794,72	44.667,91 €	
Se considera como precio del kWh/h = 0,12												238.162,71 €	
TOTAL AHORRO AL AÑO CONSUMO ELÉCTRICO												84,21%	
% ahorro consumo eléctrico anual													

Tabla 18 Estudio de Consumo Eléctrico y Coste Económico en la instalación eléctrica y comparativa con INDUCCIÓN



Se ha tenido en cuenta los siguientes factores para la elaboración de esta tabla:

- El número de bombillas y su consumo ha sido extraído de la Auditoría facilitada por el Excmo. Ayuntamiento de Pedro Muñoz.
- Se considera la siguiente equivalencia en cuanto a la sustitución de las bombillas, atendiendo su equivalencia en intensidad en la iluminación:
 - VSAP Ovoide de 100 W → Inducción 55 W
 - VSAP Ovoide de 150 W → Inducción 85 W
 - VSAP Ovoide de 250 W → Inducción 110 W
 - VSAP Tubular de 250 W → Inducción de 110 W
 - HM Ovoide de 150 W → Inducción de 85 W
 - HM Tubular de 125 W → Inducción de 85 W
- El consumo del Equipo es el extraído de la Auditoría. Por tanto, en los datos referentes al Consumo del Equipo, se tiene en cuenta el consumo real de la bombilla y de todo el equipamiento instalado.
- Se tiene en cuenta un uso de 12 horas, si bien, dependiendo de la época de año, este número de horas se incrementa o se decremента.
- El precio del kW/h usado es de 0,12 €/kW/h, precio de media de las distintas tarificaciones existentes en 2008, fecha de realización de la Auditoría

Se observa cómo, una sustitución de las bombillas actuales por bombillas de Inducción Magnética, conllevaría al Excelentísimo Ayuntamiento de Pedro Muñoz a un ahorro de **238.162,71 €** al año en la factura eléctrica.



4.3.5.1.2. LED

A continuación se realiza un estudio energético y económico de la situación actual del alumbrado público en toda la localidad de Pedro Muñoz, y su equivalente sustituyendo las actuales bombillas por unas de LED.



Tipo lámpara	pto Luz	Potencia (W) / punto	% de funcionamiento				LÁMPARA				EQUIPO				CONSUMO TOTAL
			100%	80%	60%	20%	kWh / día	kWh/año	Consumo	Pot Instalación	Consumo Pot Inst kVh/día	Consumo Pot Inst kVh/año	Consumo		
Comparativa de 1 farola de Vapor de Sodio (VSAP) frente a 1 farola de LEDs+															
VSAP Ovoide	1	100	12	0	0	0	1,2	438	52,56 €	80	0,96	350,4	42,05 €	94,61 €	
VSAP Ovoide	1	150	12	0	0	0	1,8	657	78,84 €	120	1,44	525,6	63,07 €	141,91 €	
VSAP Ovoide	1	250	12	0	0	0	3	1095	131,40 €	200	2,4	876	105,12 €	236,52 €	
VSAP Tubular	1	250	12	0	0	0	3	1095	131,40 €	200	2,4	876	105,12 €	236,52 €	
HM Ovoide	1	150	12	0	0	0	1,8	657	78,84 €	120	1,44	525,6	63,07 €	141,91 €	
HM Tubular	1	125	12	0	0	0	1,5	547,5	65,70 €	100	1,2	438	52,56 €	118,26 €	
LEDs+	1	36	4	1	0	7	0,188	68,62	8,23 €	2	0,024	8,76	1,05 €	9,29 €	
LEDs+	1	70	4	1	0	7	0,3648	133,152	15,98 €	4	0,048	17,52	2,10 €	18,08 €	
LEDs+	1	90	4	1	0	7	0,4688	171,112	20,53 €	6	0,072	26,28	3,15 €	23,69 €	
Comparativa de todas las luminarias existentes frente a luminarias LEDs+															
VSAP Ovoide	1552	100	12	0	0	0	1.862,40	679.776,00	81.573,12 €	80	1489,92	543820,8	65.258,50 €	146.831,62 €	
VSAP Ovoide	466	150	12	0	0	0	838,80	306.162,00	36.739,44 €	120	671,04	244929,6	29.391,55 €	66.130,99 €	
VSAP Ovoide	82	250	12	0	0	0	246,00	89.790,00	10.774,80 €	200	196,8	71832	8.619,84 €	19.394,64 €	
VSAP Tubular	100	250	12	0	0	0	300,00	109.500,00	13.140,00 €	200	240	87600	10.512,00 €	23.652,00 €	
HM Ovoide	154	150	12	0	0	0	277,20	101.178,00	12.141,36 €	120	221,76	80942,4	9.713,09 €	21.854,45 €	
HM Tubular	42	125	12	0	0	0	63,00	22.995,00	2.759,40 €	100	50,4	18396	2.207,52 €	4.966,92 €	
TOTALES									157.128,12 €		2.869,92	1.047.520,80	125.702,50 €	282.830,62 €	
LEDs+	1552	36	4	1	0	7	291,78	106.498,24	12.779,79 €	2	37,248	13595,52	1.631,46 €	14.411,25 €	
LEDs+	662	70	4	1	0	7	241,50	88.146,62	10.577,59 €	4	31,776	11598,24	1.391,79 €	11.969,38 €	
LEDs+	182	90	4	1	0	7	85,32	31.142,38	3.737,09 €	6	13,104	4782,96	573,96 €	4.311,04 €	
TOTALES									27.094,47 €		31,78	11.598,24	1.391,79 €	30.691,68 €	
Se considera como precio del kW/h = 0,12														252.138,94 €	
														89,15%	
TOTAL AHORRO AL AÑO CONSUMO ELÉCTRICO															
% ahorro consumo eléctrico anual															

Tabla 19. Estudio de Consumo Eléctrico y Coste Económico en la instalación eléctrica y comparativa con LED



Se ha tenido en cuenta los siguientes factores para la elaboración de esta tabla:

- El número de bombillas y su consumo ha sido extraído de la Auditoría facilitada por el Excmo. Ayuntamiento de Pedro Muñoz.
- Se considera la siguiente equivalencia en cuanto a la sustitución de las bombillas, atendiendo su equivalencia en intensidad en la iluminación:
 - VSAP Ovoide de 100 W □ LED 36 W
 - VSAP Ovoide de 150 W □ LED 70 W
 - VSAP Ovoide de 250 W □ LED 90 W
 - VSAP Tubular de 250 W □ LED 90 W
 - HM Ovoide de 150 W □ Inducción de 70 W
 - HM Tubular de 125 W □ Inducción de 70 W
- El consumo del Equipo es el extraído de la Auditoría. Por tanto, en los datos referentes al Consumo del Equipo, se tiene en cuenta el consumo real de la bombilla y de todo el equipamiento instalado.
- Se tiene en cuenta un uso de 12 horas, si bien, dependiendo de la época de año, este número de horas se incrementa o se decrementa.
- El precio del kW/h usado es de 0,12 €/kW/h, precio de media de las distintas tarificaciones existentes en 2008, fecha de realización de la Auditoría

Se observa cómo, una sustitución de las bombillas actuales por bombillas de Inducción Magnética, conllevaría al Excelentísimo Ayuntamiento de Pedro Muñoz a un ahorro de **252.138,94 €** al año en la factura eléctrica.



4.3.5.1.3. Conclusiones sobre la sustitución de bombillas

Una vez realizado el estudio realizando la sustitución por ambos tipos de bombillas, se obtienen los siguientes resultados:

Tipo de bombilla	Ahorro anual (€)	Ahorro anual (%)
INDUCCIÓN	238.162,71 €	84,21 %
LED	252.138,94 €	89,15 %

Tabla 20. Ahorro anual en el consumo energético según el tipo de bombilla

De los resultados expuestos, se extrae que la sustitución por lámparas LED conlleva un 4,94 % más de ahorro que con inducción magnética, es decir, **13.976,23 €**.

4.3.5.2. Estudio del consumo eléctrico en el Proyecto Piloto

En este apartado se muestra el consumo de la parte que estaría incluida en el Proyecto Piloto. Al igual que en el apartado anterior, se realizará en primer lugar un estudio suponiendo un cambio de las actuales bombillas por unas de Inducción, para después realizar el mismo estudio sustituyéndose por bombillas LED.

4.3.5.2.1. Inducción Magnética

A continuación se realiza un estudio energético y económico del conjunto de farolas del cual se compondría el Proyecto Piloto, y su equivalente sustituyendo las actuales bombillas por unas de Inducción:



Tipo lámpara	pto Luz	LÁMPARA						EQUIPO				CONSUMO TOTAL		
		Potencia (W) / punto	% de funcionamiento				kWh / día	kwh/año	Consumo	Pot Instalación	Consumo Pot Inst kWh/día		Consumo Pot Inst kWh/año	Consumo
			100%	80%	60%	20%								
Comparativa de 1 farola de Vapor de Sodio (VSAP) frente a 1 farola de INDUCCIÓN														
VSAP Ovoide	1	150	12	0	0	0	1,8	657	78,84 €	80	0,96	350,4	42,05 €	120,89 €
INDUCCIÓN	1	85	4	1	0	7	0,4428	161,622	19,39 €	12	0,048	17,52	2,10 €	21,50 €
Comparativa de todas las luminarias existentes frente a luminarias INDUCCIÓN														
VSAP Ovoide	6	150	12	0	0	0	10,80	3.942,00	473,04 €	80	5,76	2102,4	252,29 €	725,33 €
TOTALES							10,80	3.942,00	473,04 €		5,76	2.102,40	252,29 €	725,33 €
INDUCCIÓN	6	85	4	1	0	7	2,66	969,73	116,37 €	12	0,288	105,12	12,61 €	128,98 €
TOTALES							2,66	969,73	116,37 €		0,29	105,12	12,61 €	128,98 €
Se considera como precio del kW/h = 0,12										TOTAL AHORRO AL AÑO CONSUMO ELÉCTRICO				596,35 €
										% ahorro consumo eléctrico anual				86,72%

Tabla 21. Estudio de Consumo Eléctrico y Coste Económico en el Proyecto Piloto y comparativa con INDUCCIÓN

Las consideraciones tenidas en cuenta en este apartado son las mismas que en apartados anteriores.

Se observa cómo, una sustitución de las bombillas actuales por bombillas de Inducción Magnética, conllevaría al Excelentísimo Ayuntamiento de Pedro Muñoz a un ahorro, solamente en las 6 luminarias incluidas en este Proyecto Piloto, de **596,35 € al año** en la factura eléctrica.

4.3.5.2.2. LED

A continuación se realiza un estudio energético y económico del conjunto de farolas del cual se compondría el Proyecto Piloto, y su equivalente sustituyendo las actuales bombillas por unas de LED:

Tipo lámpara	pto Luz	LÁMPARA						EQUIPO				CONSUMO TOTAL		
		Potencia (W) / punto	% de funcionamiento				kWh / día	kwh/año	Consumo	Pot Instalación	Consumo Pot Inst kWh/día		Consumo Pot Inst kWh/año	Consumo
			100%	80%	60%	20%								
Comparativa de 1 farola de Vapor de Sodio (VSAP) frente a 1 farola de LED+														
VSAP Ovoide	1	150	12	0	0	0	1,8	657	78,84 €	120	1,44	525,6	63,07 €	141,91 €
LEDs+	1	70	4	1	0	7	0,3648	133,152	15,98 €	4	0,016	5,84	0,70 €	16,68 €
Comparativa de todas las luminarias existentes frente a luminarias LED+														
VSAP Ovoide	6	150	12	0	0	0	10,80	3.942,00	473,04 €	120	8,64	3153,6	378,43 €	851,47 €
TOTALES							10,80	3.942,00	473,04 €		8,64	3.153,60	378,43 €	851,47 €
LEDs+	6	70	4	1	0	7	2,19	798,91	95,87 €	4	0,096	35,04	4,20 €	100,07 €
TOTALES							2,19	798,91	95,87 €		0,10	35,04	4,20 €	100,07 €
Se considera como precio del kW/h = 0,12										TOTAL AHORRO AL AÑO CONSUMO ELÉCTRICO				751,40 €
70 W en LED emiten con la misma intensidad que 150 W en VAS										% ahorro consumo eléctrico anual				92,73%

Tabla 22. Estudio de consumo eléctrico y coste económico en el Proyecto Piloto y comparativa con LED

Las consideraciones tenidas en cuenta en este apartado son las mismas que en apartados anteriores.



Se observa cómo, una sustitución de las bombillas actuales por bombillas LED, conllevaría al Excelentísimo Ayuntamiento de Pedro Muñoz a un ahorro, solamente en las 6 luminarias incluidas en este Proyecto Piloto, de **751,40 € al año** en la factura eléctrica.

4.3.5.2.3. Conclusiones

Una vez realizado el estudio realizando la sustitución por ambos tipos de bombillas, obtenemos los siguientes resultados:

Tipo de bombilla	Ahorro anual (€)	Ahorro anual (%)
INDUCCIÓN	603,44 €	70,87 %
LED	710,09 €	83,40 %

Tabla 23. Comparativa de ahorro entre bombillas.

De los resultados expuestos, se extrae que la sustitución por lámparas LED conlleva un **13,47 % más de ahorro** que con inducción magnética, es decir, **93,35 €**.

4.3.6. Red Inalámbrica de Nodos con Sensores

Para realizar una gestión adecuada del alumbrado público, se propone implementar una Red Inalámbrica de Sensores. Como su propio nombre indica, esta red estará compuesta por diferentes nodos instalados en las farolas, los cuales estarán, por un lado, compuestos por 2 sensores, que se encargarán de recoger los datos necesarios del exterior, y por otro lado, estarán dotados de la posibilidad de comunicarse de forma inalámbrica para intercambiar información útil que permita obtener unos resultados más óptimos en cuanto al funcionamiento de la red.

Dicha regulación se realizará en base a los datos obtenidos por los siguientes sensores:



- Sensor de Iluminación
- Sensor de Movimiento.

En los siguientes apartados se hace una presentación de los sensores encargados de recoger la información del exterior, así como la explicación del funcionamiento de la instalación.

4.3.6.1. Sensores

4.3.6.1.1. Iluminación

El principal factor a partir del cual se definirá la máxima intensidad con la que emitirá cada una de las luminarias en las que se realice la instalación, será el de la iluminación que se mida en cada momento. De esta forma, desde el momento en el que empieza a anochecer hasta que se hace la noche completamente, el máximo de intensidad con el cual esté funcionando la luminaria irá sufriendo variaciones. De esta forma se consigue que la luminaria no esté emitiendo al máximo de potencia desde el momento en que ésta se enciende, consiguiendo el correspondiente ahorro en el consumo.

Para medir la intensidad y actuar en consecuencia, se hará uso del sensor que se muestra en la Figura 34:



Figura 34. Sensor de Iluminación.

Este sensor dotado de un fotoresistor, será capaz de medir la intensidad de la luz ambiental, generando según el resultado de la medición una determinada tensión.

La intensidad máxima con la que emitirá la farola estará determinada por el resultado de esta medición, y se hará efectiva cuando el sensor de movimiento detecte presencia, tal y como se explica en el siguiente apartado.

4.3.6.1.2. Movimiento

El segundo factor que se tendrá en cuenta para decidir la intensidad con la que la farola operará, vendrá dado por la detección de presencia en el rango de acción del sensor que se encargará de tal fin.

Para detectar la presencia y actuar en consecuencia, se hará uso del sensor que se muestra en la Figura 35:



Figura 35. Módulo con sensor PIR

Dicho sensor posee las siguientes características:

- Ángulo de detección: 120°.
- Máxima distancia de detección: 6 metros.

Este sensor además cuenta con unos potenciómetros que permiten regular la distancia máxima (hasta un máximo de 6 metros) a detectar, así como el tiempo de detección de movimiento hasta activarse.

En las siguientes figuras se muestra una simulación en 3D de la zona de actuación y la cobertura de los sensores de movimiento:

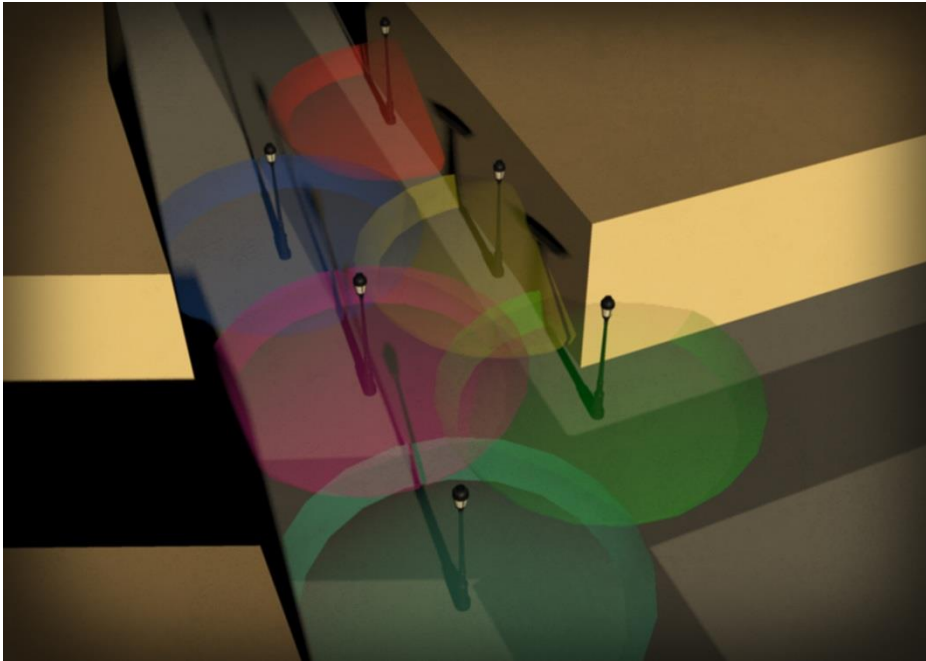


Figura 36. Simulación en 3D de la zona de detección de los sensores PIR.

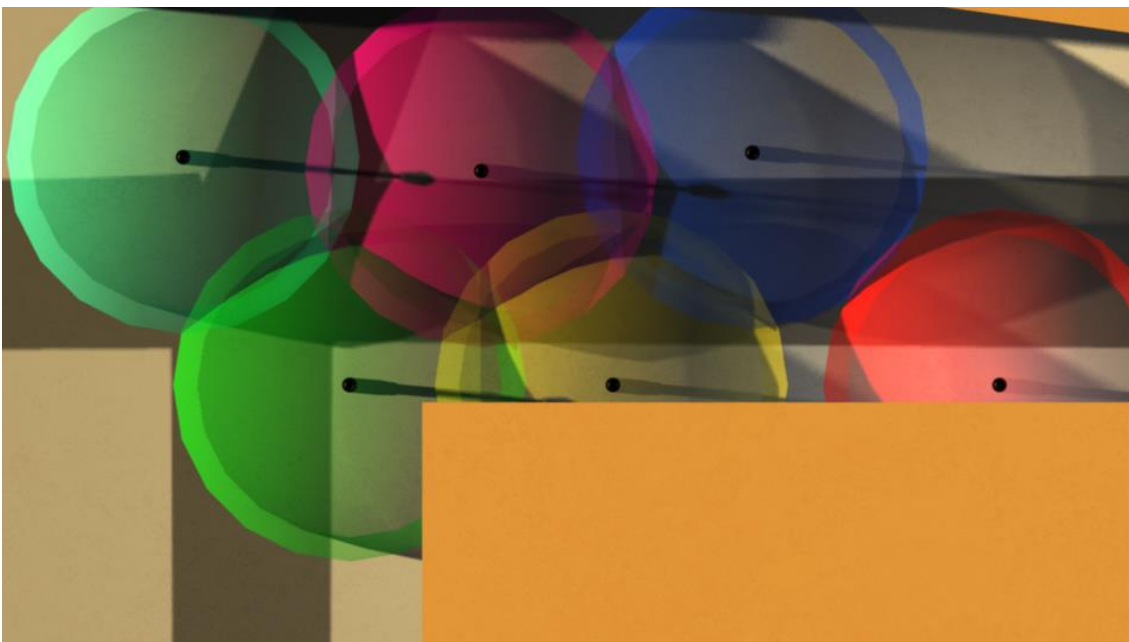


Figura 37. Vista aérea de la zona de detección de los sensores PIR

En las figuras anteriores se puede observar el rango de acción de cada uno de los sensores instalados en cada uno de los nodos, representados con la parte correspondiente de un cono de 120° , ángulo de detección del sensor PIR.



Las imágenes muestran una recreación real, con medidas reales de las farolas sobre las que se desarrollaría el Proyecto Piloto.

4.3.6.2. Funcionamiento de la infraestructura

El funcionamiento de la infraestructura sería el siguiente:

- El sensor de iluminación, estará realizando medidas constantemente de la intensidad de la luz ambiente. Se establecerán unos valores a partir de los cuales la bombilla funcionará con una determinada intensidad. Para ejemplificarlo, piénsese en los minutos en los cuales está anocheciendo, durante estos minutos de transición del día a la noche, la bombilla trabajaría, por ejemplo, a un 75 % de máxima intensidad, hasta llegar a la noche completamente, momento en el que el máximo de intensidad de la bombilla sería del 100%. Esto permite no solo el control de la noche y el día, sino también de los días en que debido a una situación meteorológica adversa como puede ser que el cielo esté completamente nublado y se necesite hacer uso de la infraestructura eléctrica, se consiga tener una iluminación de acuerdo a las necesidades de ese momento.
- Como se ha dicho en el punto anterior, el sensor de movimiento establecerá el máximo de intensidad con el cual trabajará la farola. Ese máximo de intensidad se dará cuando el sensor de movimiento instalado en el nodo correspondiente de cada farola detecte algún tipo de presencia, ya sea de una persona o de un vehículo. En ese momento esa farola emitirá al máximo de potencia. Esto se puede ver en la Figura 38:



Figura 38. Recreación en 3D del funcionamiento de la instalación propuesta.

- En esta ilustración se observa cómo la farola 1 (numérense las farolas de arriba abajo siguiendo un zigzag) emite al máximo de potencia ya que ha detectado la presencia de una persona. La posibilidad de transmisión de datos de forma inalámbrica permitirá avisar a las farolas contiguas (farolas 2 y 3) para que aumenten su intensidad del mínimo a un valor intermedio, quedando el resto (farolas 4, 5 y 6) trabajando al mínimo de intensidad.
- La persona sigue andando y entraría en el rango de acción de la farola 3, tal y como se muestra en la Figura 39:



Figura 39. Recreación en 3D del funcionamiento de la instalación propuesta

- Al igual que se ha descrito en el paso anterior, la farola que en este caso se iluminaría al máximo de intensidad sería la farola número 3, quedando las contiguas (farolas 1, 2, 4 y 5) trabajando con una intensidad intermedia, ya que la farola que ha detectado el movimiento ha avisado a estas últimas para que así sea.
- Por último, la persona entraría en el rango de acción de la farola número 5 tal y como se puede ver en la Figura 40.

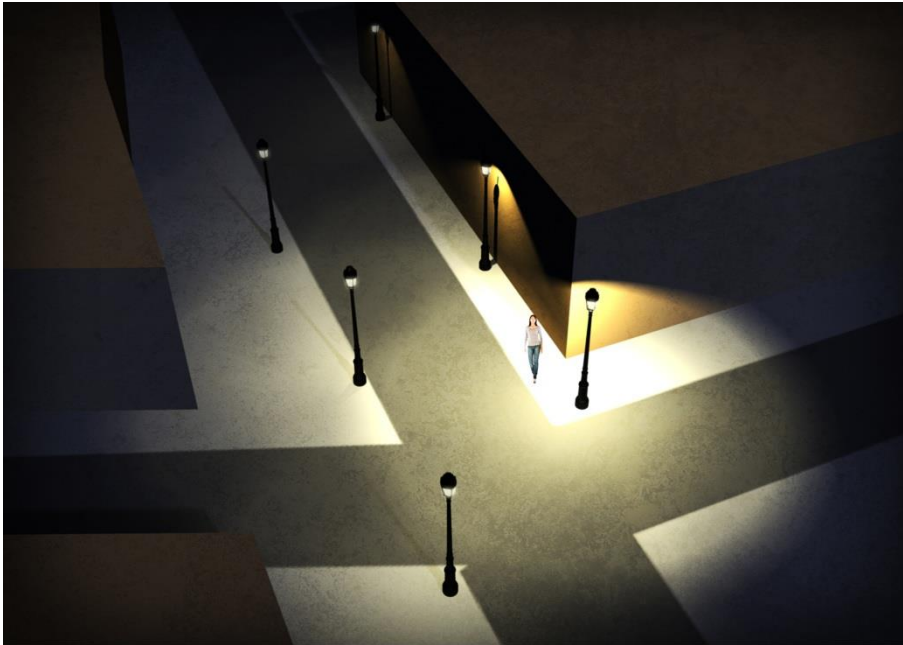


Figura 40. Recreación en 3D del funcionamiento de la instalación propuesta.

- Al igual que en los pasos anteriores, la farola número 5 quedaría iluminada al 100%, comunicándoles a las contiguas (farolas 3,4 y 6) que iluminen a una intensidad intermedia, y por tanto, quedando las farolas más alejada (farolas 1 y 2) trabajando a la intensidad mínima.

A través de este sistema, se consigue un uso eficiente de la infraestructura de iluminación pública, evitando un consumo innecesario en los momentos en los que no se necesita iluminación completa. Además, permitiría una adaptación constante al cambio en la hora en la que anoche y amanece.

Más, adelante, en el capítulo dedicado a la simulación de la infraestructura, se explicará de forma más detallada, la operativa de la infraestructura en cuanto a comunicación entre elementos, recogidas de datos y demás operativa.



4.3.7. Presupuesto

En este apartado se va a recoger el presupuesto necesario para implementar la red inalámbrica de sensores, realizando además una estimación del cambio de lámparas. Primero se realizará un estudio de la implementación total del alumbrado público, y posteriormente, un estudio de la implementación del Proyecto Piloto.

4.3.7.1. Presupuesto de los elementos individualizados.

4.3.7.1.1. Nodos

- Nodos con sensores.

Cada uno de los nodos instalados en cada una de las farolas estaría compuesto por los siguientes elementos:

Elemento	Unidades	€/Unidad	Total Sin IVA	Total Con IVA
Arduino Fio	1	19,90 €	19,90 €	24,08 €
Conectores módulos conectar y listo	2	0,17 €	0,33 €	0,40 €
Módulo Sensor PIR	1	10,75 €	10,75 €	13,01 €
Módulo Sensor LUZ	1	4,25 €	4,25 €	5,14 €
Placa perforada	1	2,45 €	2,45 €	2,96 €
cables	2	1,39 €	2,78 €	3,236€
Transformador 220V/12V	1	5,75 €	5,75 €	6,95 €
Módulo Xbee	1	23,50 €	23,50 €	28,43 €
Módulo Consumo Eléctrico	1	14,25 €	14,25 €	17,25 €
		TOTAL	83,96 €	99,66 €

Tabla 24. Presupuesto por nodo con sensores



- Cada nodo, tal y como se puede apreciar en la Tabla 21 está compuesto por
 - **Arduino FIO:** Placa controladora que se encargará de recoger los datos de los sensores y actuar en consecuencia.
 - **Conectores módulos Conectar y Listo:** Conectores necesarios para realizar el conexionado de los módulos que albergan los sensores de Presencia e Iluminación.
 - **Módulo Sensor PIR:** Módulo con el sensor de Presencia mediante infrarrojos.
 - **Módulo Sensor LUZ:** Módulo con el sensor de Iluminación.
 - **Placa Perforada:** Placa necesaria para realizar el conexionado de los conectores.
 - Cables.
 - **Transformador 220V/12V:** Transformador para alimentar la placa y los sensores.
 - **Módulo Xbee:** Módulo que permite la conexión inalámbrica de los nodos mediante tecnología ZigBee.
- Nodos sin sensores.

El nodo sin sensores hará las funciones de central, y se encargará de recoger los datos de todos los sensores y así poder obtener información de la situación actual de cada uno de los nodos instalados en las farolas.

Elemento	Unidades	€/Unidad	Total Sin IVA	Total Con IVA
Arduino Fio	1	19,90 €	19,90 €	23,48 €
Transformador 220V/12V	1	5,75 €	5,75 €	6,79 €
Módulo Xbee	1	23,50 €	23,50 €	27,73 €
		TOTAL	49,15 €	58,00 €

Tabla 25. Presupuesto por nodo sin sensores

Cada nodo, tal y como se puede apreciar en la Tabla 22 está compuesto por:

- **Arduino FIO:** Placa controladora que se encargará de recoger los datos de los sensores y actuar en consecuencia.
- **Transformador 220V/12V:** Transformador para alimentar la placa y los sensores.
- **Módulo Xbee:** Módulo que permite la conexión inalámbrica de los nodos mediante tecnología ZigBee.
- Material complementario necesario para la instalación de los nodos:

Además de los elementos necesarios descritos en los apartados anteriores, para poder realizar la fase de programación de los nodos y de los módulos de comunicación inalámbrica, es necesario lo siguiente:

Elemento	Unidades	€/Unidad	Total Sin IVA	Total Con IVA
Cable Programador	1	12,90 €	12,90 €	15,22 €
Xbee Explorer	1	19,90 €	19,90 €	23,48 €
		TOTAL	32,80 €	38,70 €

Tabla 26. Presupuesto del material complementario necesario para la instalación.

El material necesario para la programación de los nodos mostrados en la Tabla 24 y en la Tabla 25 , está compuesto por:

- **Cable Programador.** Cable necesario para realizar la programación de los nodos.
- **Xbee Explorer.** Adaptador necesario para realizar la programación de los módulos Xbee encargados de la comunicación inalámbrica.



4.3.7.1.2. Luminarias

- Inducción:

Los precios estimados de las diferentes lámparas de inducción son los siguientes:

Elemento	Unidades	€/Unidad	Total Sin IVA	Total Con IVA
Lámpara Inducción Magnética 55W	1	140 €	140 €	165,20 €
Lámpara Inducción Magnética 85W	1	180 €	180 €	212,40 €
Lámpara Inducción Magnética 110W	1	210 €	210 €	247,80 €

Tabla 27. Presupuesto estimado del cambio a lámparas de Inducción.

- LED

Los precios estimados de las diferentes lámparas LED son los siguientes:

	Unidades	€/Unidad	Total Sin IVA	Total Con IVA
Lámpara LED 36W	1	165 €	165 €	194,70 €
Lámpara LED 70W	1	225 €	225 €	265,50 €
Lámpara LED 90 W	1	400 €	400 €	472 €

Tabla 28. Presupuesto estimado del cambio a lámparas LED

4.3.7.2. Presupuesto total implementación completa del alumbrado

El presupuesto para la implementación del proyecto en toda la instalación de alumbrado público de la localidad quedaría de la siguiente forma:



	INDUCCIÓN				LED	
	Uds	€/Ud	Total Sin IVA	Total Con IVA	Total Sin IVA	Total Con IVA
Nodos con Sensores	2396	69,71 €	167.025,16 €	197.089,69 €	167.025,16 €	197.089,69 €
Nodos sin Sensores	150	49,15 €	7.372,50 €	8.699,55 €	7.372,50 €	8.699,55 €
Material complementario	1	32,80 €	32,80 €	38,70 €	32,80 €	38,70 €
Lámpara Ind. Mag. 55W	1552	140 €	217.280 €	256.390,40 €	-	-
Lámpara Ind. Mag. 85W	662	180 €	119.160 €	140.608,80 €	-	-
Lámpara Ind. Mag. 110W	182	210 €	38.220 €	45.099,60 €	-	-
Lámpara LED 36W	1552	165 €	-	-	256.080 €	302.174,40 €
Lámpara LED 70W	662	225 €	-	-	148.950 €	175.761 €
Lámpara LED 90 W	182	400 €	-	-	72.800 €	85.904 €
		TOTAL	549.090,46 €	647.926,74 €	652.260,46 €	769.667,34 €

Tabla 29 Presupuesto Total para la implementación del proyecto en la red de alumbrado completa

Tal y como se comentaba anteriormente, estos precios se reducirían debido a los descuentos aplicables por los distribuidores en compras de varios elementos a la vez.



4.3.7.4. Presupuesto total implementación Proyecto Piloto

El presupuesto para la implementación del Proyecto Piloto quedaría de la siguiente forma:

	Unidades	€/Unidad	INDUCCIÓN		LED	
			Total Sin IVA	Total Con IVA	Total Sin IVA	Total Con IVA
Nodos con Sensores	6	83,96 €	503,66 €	493,55 €	418,26 €	493,55 €
Nodos sin Sensores	1	49,15 €	49,15 €	58,00 €	49,15 €	58,00 €
Material complementario	1	32,80 €	32,80 €	38,70 €	32,80 €	38,70 €
Lámpara Ind. Mag. 85W	6	180 €	1.080,00 €	1.274,40 €	-	-
Lámpara LED 70W	6	225 €			1.350 €	1.593 €
		TOTAL	1.580,21 €	1.864,65 €	1.850,21 €	2.183,25 €

Tabla 30. Presupuesto Total para la implementación del Proyecto Piloto.

Se observa cómo el presupuesto total queda de la siguiente forma:

- Si se opta por la opción de **Inducción**, el presupuesto para la implementación del Proyecto Piloto sería de **1.864,65 €**.
- Si se opta por la opción de **LED**, el presupuesto para la implementación del Proyecto Piloto sería de **2.183,25 €**

4.3.7.5 Mano de obra

A continuación se muestra el apartado presupuestario correspondiente a la mano de obra suponiendo que el plazo en el que se ejecuta el proyecto piloto es de 1 mes:



Descripción	Cantidad	Precio Unitario	Total
Coste de Ingeniería: Desarrollo de la idea, documentación, elaboración del proyecto	1	3.000 €	3.000 €
Ingeniero Técnico de Telecomunicaciones: Ingeniero encargado de apoyar y validar todo el sistema, así como la supervisión del despliegue	22	90 €	1.980 €
Desarrollador web: Diseñador encargado de maquetar la página web en la que se mostrará la información	7	75 €	525 €
Técnico electricista: Encargado de realizar la sustitución de las luminarias, así como el anclaje de cada uno de los nodos del sistema y su conexionado	4	60 €	240 €
Alumno: Introducción de todos los datos de la red y corrección de problemas	22	25 €	550 €
TOTAL			6.295 €

Tabla 31. Presupuesto mano de obra

5.3.7.4 Comentarios sobre el presupuesto

- El tiempo aproximado de amortización de la inversión realizada en el despliegue del Proyecto Piloto sería:
 - **LED:**
 - **Inversión:** 2.183,25 €
 - **Ahorro anual:** 710,09 €
 - **Amortización aproximada:** 3 años
 - **INDUCCIÓN:**
 - **Inversión:** 1.846,65 €



- **Ahorro anual:** 603,44 €
- **Amortización aproximada:** 3 años

Se observa que el tiempo de amortización es prácticamente igual. Si bien realizando la instalación con lámparas de Inducción, la inversión inicial va a ser menor, el ahorro que se conseguirá con LED será mayor, debido a su mejor regulación de la intensidad.

- Debido a que no es objeto de esta propuesta la instalación de las nuevas bombillas, sería necesario solicitar presupuestos más exactos sobre el coste de instalación, si bien se ha buscado una aproximación en este documento.
- No se ha incluido el soporte necesario para la instalación de los nodos con los sensores, ya que se intentará buscar la solución más factible y económica a la hora de realizar la instalación, algo que se vería compensado con el descuento que realizaría la tienda distribuidora al hacer la compra de los elementos, que para el proyecto piloto es del 10 %, lo que conllevaría una rebaja de 59,25 € en el precio final.



5

Desarrollo Experimental



Jorge Gálvez Alfaro

Desarrollo de sistema de gestión de alumbrado público mediante WSN



POLITÉCNICA



5.1. Introducción

En el desarrollo de la parte experimental de este Proyecto Fin de Carrera ha sido necesario hacer uso de **múltiples plataformas y tecnologías**, desde el Kit de Desarrollo de Oracle para el desarrollo del código a ejecutar en las SunSPOT desplegadas, hasta un servidor Apache, para poder hacer uso del lenguaje de programación para páginas web en PHP, pasando por una máquina virtual de java, un servidor MySQL, o el uso de diferentes API de Google.

El uso de estas tecnologías ha sido necesario para poder implementar el sistema por el cual, una red de alumbrado público, sin más despliegue tecnológico que la propia farola, se vea evolucionada hasta un punto que permita un control exhaustivo del estado del funcionamiento de las mismas, su consumo, etc. Estas mejoras tecnológicas, tal y como ya se ha explicado en puntos anteriores, permitirán un ahorro, tanto energético, como económico, a la entidad que pudiera desplegarlo.

En los próximos apartados:

- Se dará una breve **explicación de los diferentes elementos que conforman el sistema**, y algunos de los cuales ya se han nombrado en el primer párrafo de este apartado.
- Se expondrá el **escenario** escogido y los **objetivos** perseguidos con la elaboración de estas pruebas.
- Se analizará también el **funcionamiento del sistema**, donde se detallará, entre otros, la puesta en marcha de las SunSPOT instaladas en las farolas, el algoritmo seguido para descubrir qué farolas se califican como “vecinas” de otra, el funcionamiento según las medidas recogidas por los sensores de luminosidad y presencia, o la estructura de la página web de monitorización.

5.2. Elementos

Debido a la naturaleza del sistema a desarrollar, múltiples son las tecnologías, y por tanto, múltiples los elementos a utilizar.

Para tener una primera visión a nivel general del sistema, se muestra la siguiente figura²:

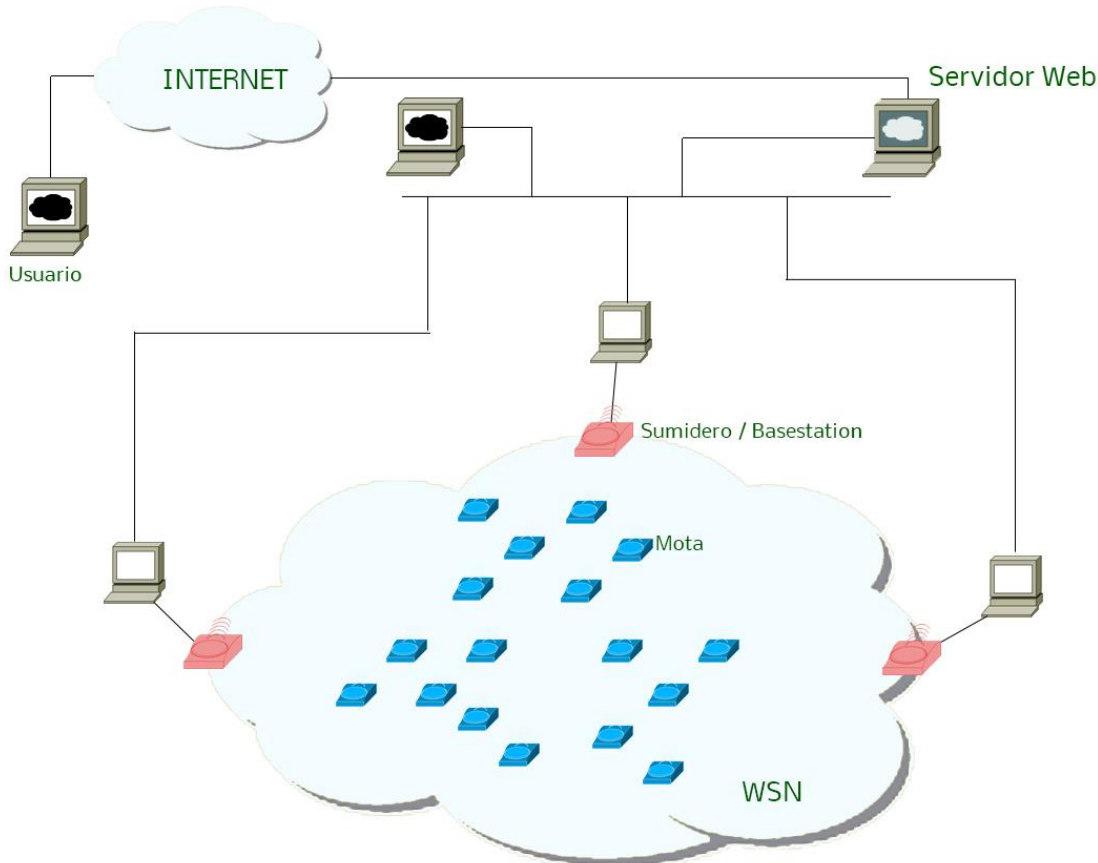


Figura 41. Esquema general del sistema.

Se pueden diferenciar 3 partes en el esquema del sistema:

- **WSN:** formada por las SunSPOT, las cuales irán instaladas en cada una de las farolas, además de los sumideros, que serán los encargados de enviar la información necesaria para su puesta en marcha a las SunSPOT, así como para recoger los datos necesarios para su monitorización y control.

² Iconos extraídos de <http://www.cisco.com/web/about/ac50/ac47/2.html>

- **SERVIDOR WEB:** En él se encontrarán instaladas las aplicaciones necesarias para poder, por un lado, almacenar la información de las farolas (id, localización,...) y de las SunSPOT (MAC, ID,...), así como para poder presentar de forma dinámica la página web a los usuarios.
- **USUARIO:** Serán los puestos que, bien a través de Internet, o bien a través de la Red de Área Local (LAN), accederán a la página web de monitorización del sistema.

Para la realización de las pruebas y desarrollo, el esquema seguido es el siguiente:

Máquina Virtual VMWare Ubuntu

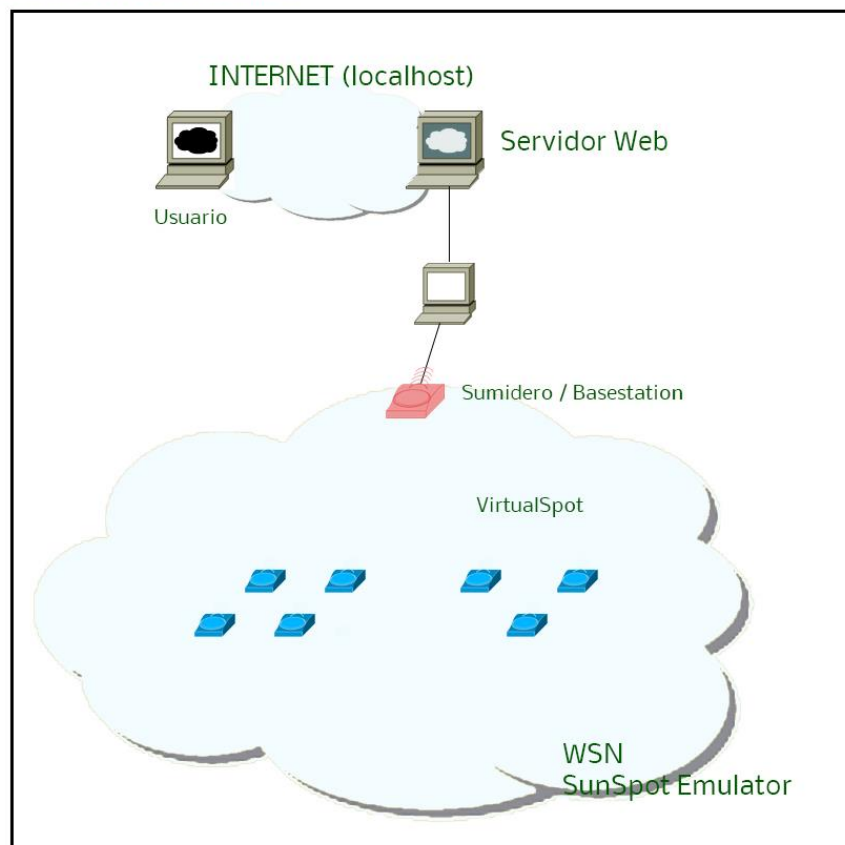


Figura 42. Esquema general del sistema implementado para la realización de las pruebas experimentales.

A continuación se realiza una descripción más detallada de todos los elementos que forman parte.



5.2.1. Máquina Virtual

Para el desarrollo de todo el sistema se ha utilizado una **máquina virtual emulada en VMWare** con las siguientes características:

- Memoria RAM: 1 Gb.
- Procesadores: 1.
- Disco Duro: 8 Gb.

El sistema operativo emulado ha sido la distribución 10.04 LTS de Ubuntu, junto al escritorio GNOME Version 2.30.2.

5.2.2. Software JAVA

Para poder utilizar tanto la propia máquina virtual JAVA, como el Kit de Desarrollo de Software específico para SunSPOT, se requiere como base, un Entorno de Tiempo de Ejecución (JRE), es decir, la Máquina Virtual JAVA, compuesta por las bibliotecas y componentes necesarios para poder ejecutar una aplicación JAVA.

Más específicamente, para poder desarrollar aplicaciones para que posteriormente sean ejecutadas en las SunSPOT, se necesita un SDK (Software Development Kit), Kit de Desarrollo de Software específico para Sunspot.

Seis son el número de versiones que han aparecido hasta el momento. La más actual, la "v6.0 Yellow", siendo ésta la versión utilizada para el desarrollo de este PFC.

Para la instalación y futuras actualizaciones de los diferentes elementos que forman el entorno de desarrollo, Oracle / Sun facilita una aplicación, SPOT Manager, la cual facilita al desarrollador todo el proceso de puesta en marcha.



Figura 43. Captura de Sun SPOT Manager Tool

Además, se podrá, entre otras cosas, administrar toda la red de SunSPOT conectadas, ver la documentación, elegir el SDK a utilizar.

Sun SPOT Manager cuenta también con la aplicación **Solarium**, gracias a la cual, podremos realizar lo siguiente [42]:

- **Descubrimiento de SPOTs:** Cualquier SunSPOT, ya esté conectada mediante USB al puesto en el que se está ejecutando Solarium, o bien que sea alcanzable a través de la radio comunicación que poseen dichas SunSPOTs, será mostrada en el programa para poder ser gestionadas.
- **Gestión de SPOTs.** A través de Solarium, se podrá interactuar con las SunSPOTs descubiertas, pudiendo realizar operaciones como iniciar/finalizar/pausar/parar aplicaciones, obtener estadísticas de uso y consumo, etc.
- **Gestión de redes de SPOT.** Gestionar redes de SPOTs de manera fácil, pudiendo por ejemplo, desplegar a la vez la misma aplicación en un grupo de SPOTs.

- **Emulación de SPOTs.** Característica más importante para el desarrollo de este PFC, ya que podremos virtualizar SPOTs y probar aplicaciones en ellas.

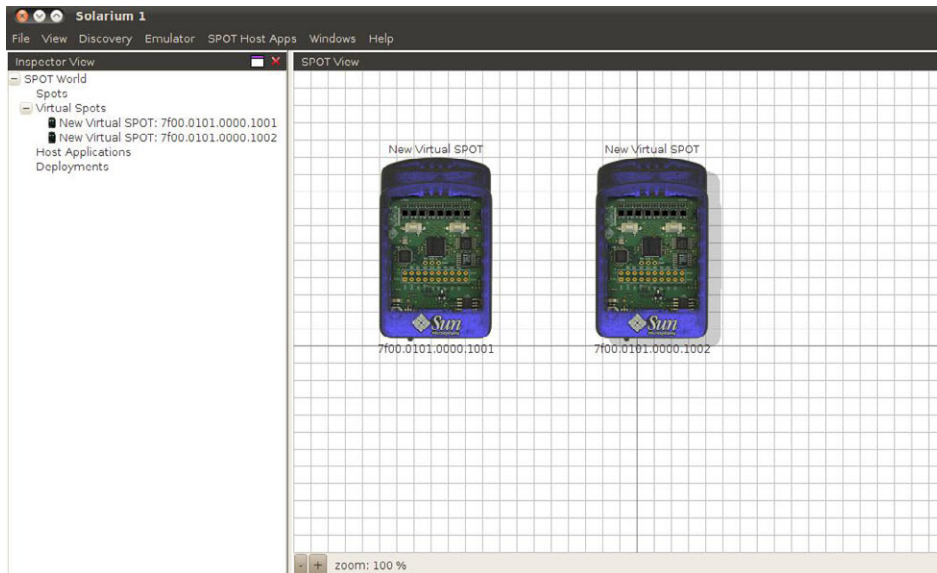


Figura 44. Captura programa Solarium

Las características de estas SPOTs virtuales son idénticas a las de las SPOTs reales. Éstas son algunas de sus características principales:

- Memoria RAM: 1 Mb.
- Memoria Flash: 8 Mb.
- Dimensiones: 41 x 23 x 70 mm.
- Procesador: Basado en AT91SAM9G20 a 133.3248 MHz.
- Sistema Operativo: Squawk Virtual Machine.
- Comunicaciones: Radio 2.4 GHz IEEE 802.15.4 con antenna integrada
- Alimentación: Batería lón-litio recarable de 3.7 V y 770 mAh.

A continuación se muestra una ilustración de las diferentes piezas por las que están formadas las Sun SPOTs:

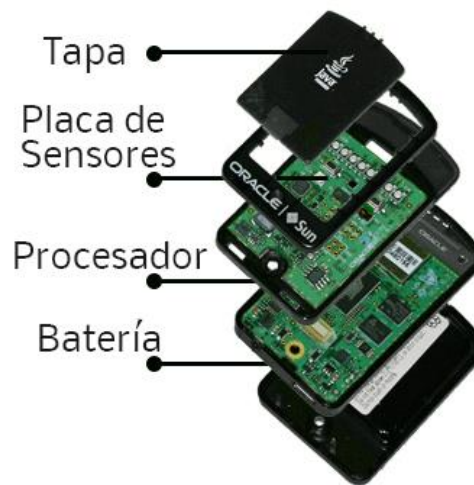


Figura 45. Estructura de una Sun SPOT (Imagen extraída de³

Debido a su naturaleza, y por tanto, la necesidad de que el consumo y la capacidad de procesamiento estén equilibrados, las Sun SPOT trabajan bajo una máquina virtual de JAVA denominada **Squawk Virtual Machine**, de código abierto, y diseñada para ser implementada en sistemas embebidos y dispositivos pequeños [43].

Lo que diferencia al resto de máquinas virtuales es que, mientras éstas suelen estar en su mayor parte codificadas en lenguaje C y Ensamblador, **Squawk está prácticamente escrita en JAVA**, lo que permite una mayor portabilidad, así como su integración con otros sistemas operativos. En la siguiente ilustración se observa esta diferencia:

³ <http://www.sunspotworld.com/products/index.html>

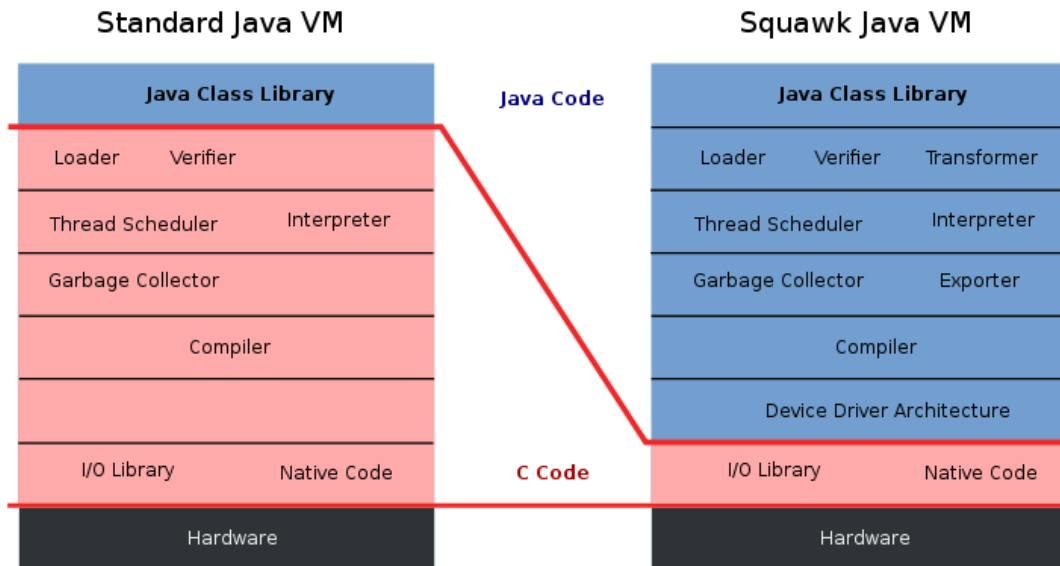


Figura 46. Comparativa de la máquina virtual Squawk de JAVA, con una máquina virtual estándar de JAVA (Imagen extraída de ⁴)

En lo que respecta a los elementos JAVA utilizados para la realización de este PFC, cabe destacar el JDBC [44], API que nos permitirá entrelazar los programas en JAVA con una Base de Datos, en este caso, MySQL. Más concretamente, mediante JDBC podremos:

- Conectarse a una Base de Datos.
- Enviar Querys a la Base de Datos.
- Recibir los resultados a las Querys.

Gracias a ello, se podrá almacenar en una Base de Datos toda la información necesaria para el correcto funcionamiento del sistema (vecindades entre farolas, medidas recibidas), permitiendo un fácil acceso a dicha información.

JDBC, entre otros, tiene los siguientes componentes:

- **JDBC API**, que proporciona las clases y métodos necesarios para poder operar con la base de datos.
- **JDBC Driver**, que servirá de interconexión entre el programa JAVA y la Base de Datos.

⁴ http://en.wikipedia.org/wiki/File:Standard_Java_VM_vs_Squawk_Java_VM.svg

A continuación, se muestra un esquema de la relación entre los diversos componentes que forman parte en la interrelación entre el programa en JAVA y la Base de Datos:

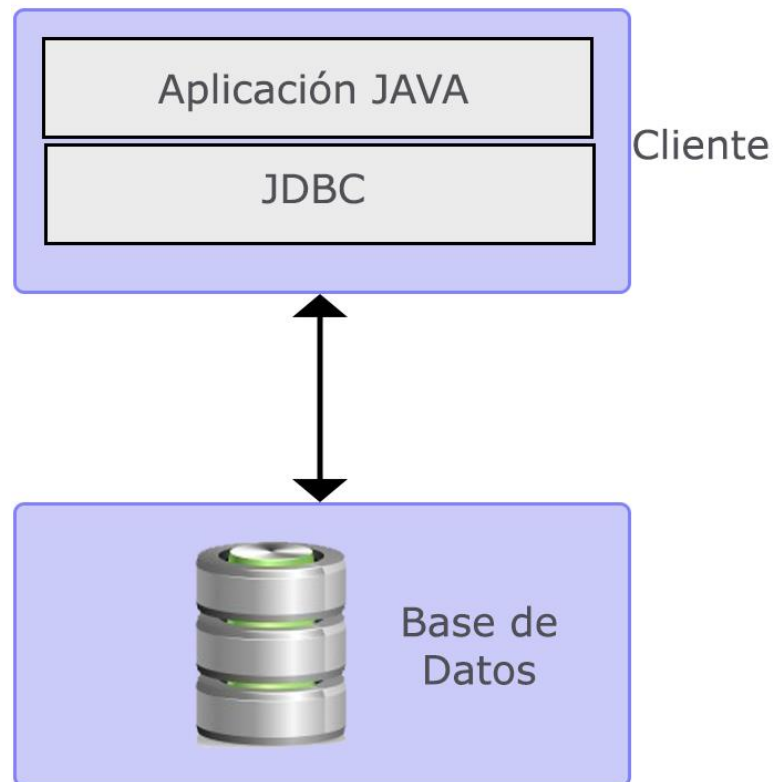


Figura 47. Relaciones de los elementos JAVA con la Base de Datos

5.2.3. Base de Datos

Para almacenar toda la información necesaria para el correcto funcionamiento del sistema, se hace uso de MySQL, sistema de gestión de bases de datos relacionales, compatible con una gran cantidad de lenguajes y sistemas operativos [45]. Para la realización de este proyecto se ha utilizado la distribución para Linux.



Más adelante, se expondrán los modelos relacionales utilizados para el desarrollo de este PFC.

5.2.4. Gestor Web

Para poder mostrar la información al usuario a través de una página web, y que dicha información sea mostrada de forma dinámica, se hace uso de la tecnología de programación web PHP, lo que conlleva la necesidad de instalación de APACHE, un servidor HTTP capaz de soportar dicho lenguaje y mostrar la página de forma correcta. Para ello, consta por un lado de la distribución básica, llamada "core", y posteriormente de diversos módulos que permiten añadir funcionalidades. En este caso, se utiliza el módulo de PHP "mod_php", que permite el manejo de páginas dinámicas en PHP [46].

5.2.5. APIs de Google

Aunque se explicará el uso concreto que se hace de ellas más adelante, mencionar que en este proyecto se hace también uso de algunas **APIs de Google** [47], más concretamente **API Google Chart** y **API Google Maps**, las cuales permitirán la inserción, por un lado, de mapas de Google Maps con sus correspondientes marcadores, y por otro lado, presentar gráficas al usuario con cierta información. Desde la página de google destinada a los Desarrolladores, se puede tener acceso a éstas y otras APIs para manejar y presentar todo tipo de información.



5.3. Escenario y Objetivos

En este apartado se va a tratar, por un lado, el escenario desplegado en la parte práctica: localización geográfica, número de elementos que forman parte del sistema, así como qué objetivos se pretenden conseguir con su desarrollo: mejora eficiencia energética, monitorización,...

5.3.1. Objetivos

Debido a la tecnología que actualmente se encuentra en uso en la mayor parte de los sistemas de alumbrado público, tanto el consumo energético como el gasto económico es elevado. Un simple cambio de bombilla, pasando de bombillas VSAP (Vapor de Sodio) a LED, conlleva un ahorro considerable en estos dos campos. Sin embargo, si completamos dicho cambio con otros, además del ahorro, que puede ser aún mayor, se nos abre un amplio abanico de funcionalidades a implementar, logrando que la eficiencia sea mayor. Más concretamente, en el desarrollo de este PFC se busca:

- Ofrecer una opción de **sustitución de lámparas**, ya expuesta en puntos anteriores, que permitan un ahorro, tanto energético, como económico.
- La sustitución de las lámparas es necesaria para poder llevar a cabo el desarrollo, ya que las que actualmente están instaladas, no permiten una modificación de la intensidad. Y es que este es otro de los objetivos que se persiguen: **modificar la intensidad de funcionamiento** para adecuarla a las condiciones ambientales. Esto se consigue mediante dos vías:



- Un **sensor de luminosidad**, que permite poder controlar el momento de encendido y apagado de las farolas, y por tanto, ajustarlo más a la situación real de necesidad de alumbrado.
- Un **sensor de presencia**, que permite tener las farolas “en reposo”, funcionando a una intensidad menor, cuando no exista movimiento en su rango de acción. Con ello se conseguirá que no se realice un gasto innecesario de energía, y por tanto, optimizar aún más su funcionamiento.
- Ofrecer una herramienta de monitorización telemática a través de la cual, se consiga tener una visión exhaustiva de los diferentes elementos que conforman la red, pudiendo así, tener conocimiento inmediato, por ejemplo, en caso de rotura, tener actualizado el consumo, etc.

Otras son las aplicaciones que se podrían implementar en este sistema. Cuenta de ellas se darán en el “apartado 6. Conclusiones y desarrollos futuros” donde se propondrán mejoras para el futuro.

5.3.2. Escenario

El escenario escogido para la representación del Sistema ha sido la localidad de Pedro Muñoz (Ciudad Real). Durante meses se mantuvieron con el Alcalde de la localidad, D. José Juan Fernández, diversas reuniones para poder realizar un despliegue experimental en dicha localidad. Debido a los problemas económicos del consistorio no se pudo realizar, si bien la actitud siempre fue muy positiva. El Excelentísimo Ayuntamiento de Pedro Muñoz puso a disposición toda la información requerida para la elaboración de este proyecto: planos, memorias energéticas,... Es por ello por lo que, aun no habiendo podido realizar el

despliegue real en la localidad, se han realizado las simulaciones tomando como referencia los datos facilitados.

Las simulaciones se van a realizar sobre parte de la Calle Costa Rica, más concretamente en 2 fragmentos de dicha calle, los cuales están compuestos por 4 y 3 farolas respectivamente:

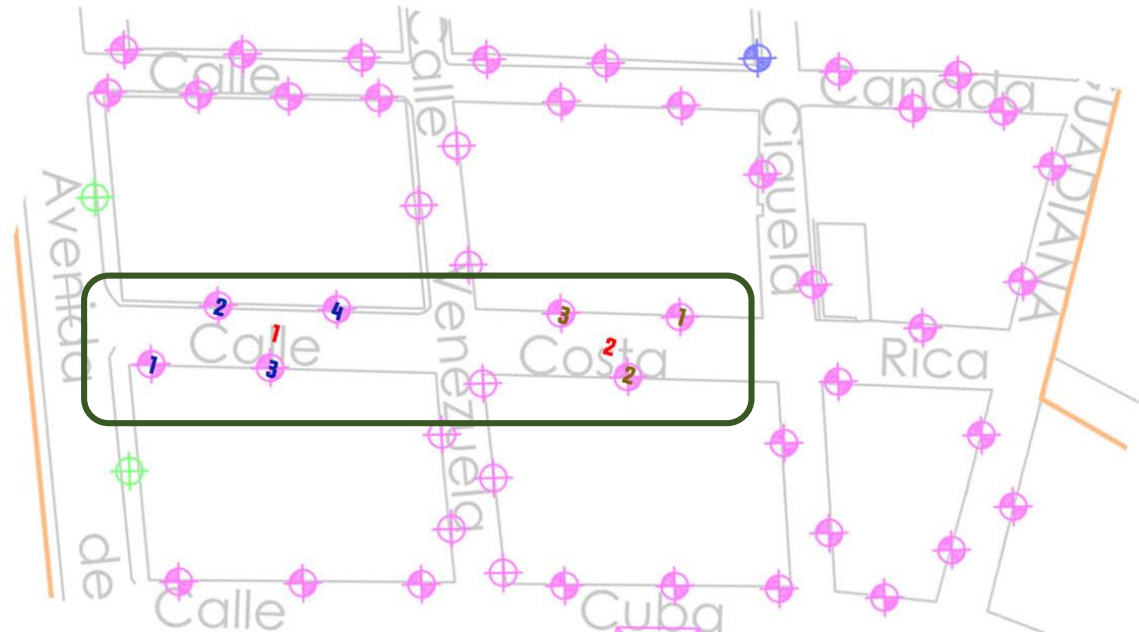


Figura 48. Localización sobre las que se han realizado las simulaciones

Más adelante se explicará el criterio de asignación de IDs a las diferentes SunSPOT.

5.4. Funcionamiento

5.4.1. Visión General

En los próximos apartados, se explicará el funcionamiento del sistema, analizando cada uno de los pasos y situaciones que se darán durante el funcionamiento real. La visión general del proceso es el siguiente:



Al conectar la SunSPOT, por un lado se inicia la lectura del nivel de iluminación para, dependiendo de la lectura, la farola funcione o esté apagada. Además, se establecerá una comunicación con la Basestation para conocer los vecinos con los que va a interactuar posteriormente. A partir de ese momento, comienzan a enviarse avisos a la Basestation cada vez que una SunSPOT detecte presencia, o sea avisado por una SunSPOT vecina, datos que se almacenarán en la base de datos para su análisis.

En los siguientes apartados se explica detalladamente cada proceso.

5.4.2. Puesta en marcha

Cada una de las SunSPOT, implementan la clase "Mota", la cual contendrá los métodos necesarios para que el sistema funcione de forma correcta:

```
package net.java.dev.netbeansspot;

import com.sun.spot.resources.transducers.Condition;
import com.sun.spot.resources.transducers.SensorEvent;
import com.sun.spot.sensorboard.peripheral.ILightSensor;
import com.sun.spot.resources.Resources;
import com.sun.spot.resources.transducers.ISwitch;
import com.sun.spot.resources.transducers.ISwitchListener;
import com.sun.spot.resources.transducers.LEDColour;
import com.sun.spot.resources.transducers.ITriColourLEDArray;
import com.sun.spot.resources.transducers.SwitchEvent;
import com.sun.spot.io.j2me.radiogram.*;
import com.sun.spot.resources.transducers.IConditionListener;
import com.sun.spot.resources.transducers.LightSensorEvent;
import java.util.Calendar;
import java.util.Date;
import java.io.*;
import javax.microedition.io.*;
import javax.microedition.midlet.MIDlet;
```



```
import javax.microedition.midlet.MIDletStateChangeException;
import com.sun.spot.peripheral.*;
import com.sun.spot.resources.transducers.IMeasurementInfo;

/**
 *
 * @author Jorge Galvez Alfaro
 *
 */
public class Mota extends MIDlet implements ISwitchListener,
IConditionListener
{
```

Al instalar la SunSPOT y ponerla en funcionamiento, los primeros métodos que se ejecutan son los siguientes:

```
protected void startApp() throws MIDletStateChangeException
{
    start = System.currentTimeMillis();
    try
    {
        sw1.addISwitchListener(this); //Activamos pulsador izdo
        sw2.addISwitchListener(this); //Activamos pulsador dcho
        monitorizacionLuminosidad.start(); //Se inicia la monitorización de
la intensidad lumínica
    }
    catch (Exception tex)
    {
    }
}
```

Se inicializa la variable "start" a la fecha y hora actual (contado en milisegundos), que servirá para calcular la diferencia de tiempo entre una medida y otra. Más concretamente, esta primera asignación servirá para la primera medida que se enviará. Posteriormente, tras cada medida, esta variable se va actualizando.

Después, se ponen en funcionamiento los dos botones que posee la SunSPOT. Debido a no poder simular la presencia a través de Solarium, se ha utilizado uno de estos botones (izquierdo, "sw1"). Este, cada vez que sea presionado, representará la entrada o salida de alguna presencia en el rango de detección del supuesto sensor. El otro interruptor (Derecho,



“sw2”), sirve para inicializar las SunSPOTs y de esta forma poder saber cuáles son los vecinos con los que se tendrá que comunicar cuando se detecte, o se deje de detectar, una presencia.

Puestos en funcionamiento los pulsadores, se inicia la monitorización de la intensidad lumínica. Para controlarlo, se usará un “Condition”, a través del cual, se puede ejecutar una parte de código si se cumple cierta condición.

El funcionamiento será el siguiente:

- La SunSPOT estará leyendo el valor de la intensidad lumínica cada 3 segundos. Existirá una variable “bloqueo_inicial” que valdrá 0 hasta que se haya inicializado la SunSPOT pulsando el botón derecho. Una vez inicializada la SunSPOT, esta variable pasará a valer 1, y a actuar en función de la intensidad leída.
 - Si la intensidad es > 200 (ésta es la medida que obtiene el sensor. La equivalencia en lux es la siguiente: $100 \text{ lx} \rightarrow 50$ [48]. Por tanto, 400 lx , intensidad aproximada al anochecer [49], sería 200 de medida del sensor), y es la primera lectura por encima de ese valor, procederá a enviar la información a la Basestation. Después, se bloqueará con “bloqueo_apagada” para que en las sucesivas lecturas por encima de ese valor, no se vuelva a enviar la información a la basestation, puesto que el detectar presencia en ese periodo, no afectará al porcentaje de funcionamiento de la farola, ya que éste será 0 durante todo momento.
 - Si la intensidad es < 200 , y es la primera lectura por debajo de ese valor, se procederá a enviar nuevamente la información a la Basestation. Esta vez, como sí depende del hecho de que haya presencia en ese momento, o haya vecinos con presencia, sí se tendrá en cuenta estos factores a la hora de enviar la información. Una vez realizado el envío, se bloquea



para que no efectúe ninguna operación hasta que la intensidad lumínica vuelva a ser superior, y se habilita la variable "lum_alcanzada" para que al presionar el switch izquierdo, el que simula la presencia, se efectúen las comunicaciones con los vecinos. El funcionamiento se explicará posteriormente.

A continuación se muestra un diagrama con el funcionamiento explicado:

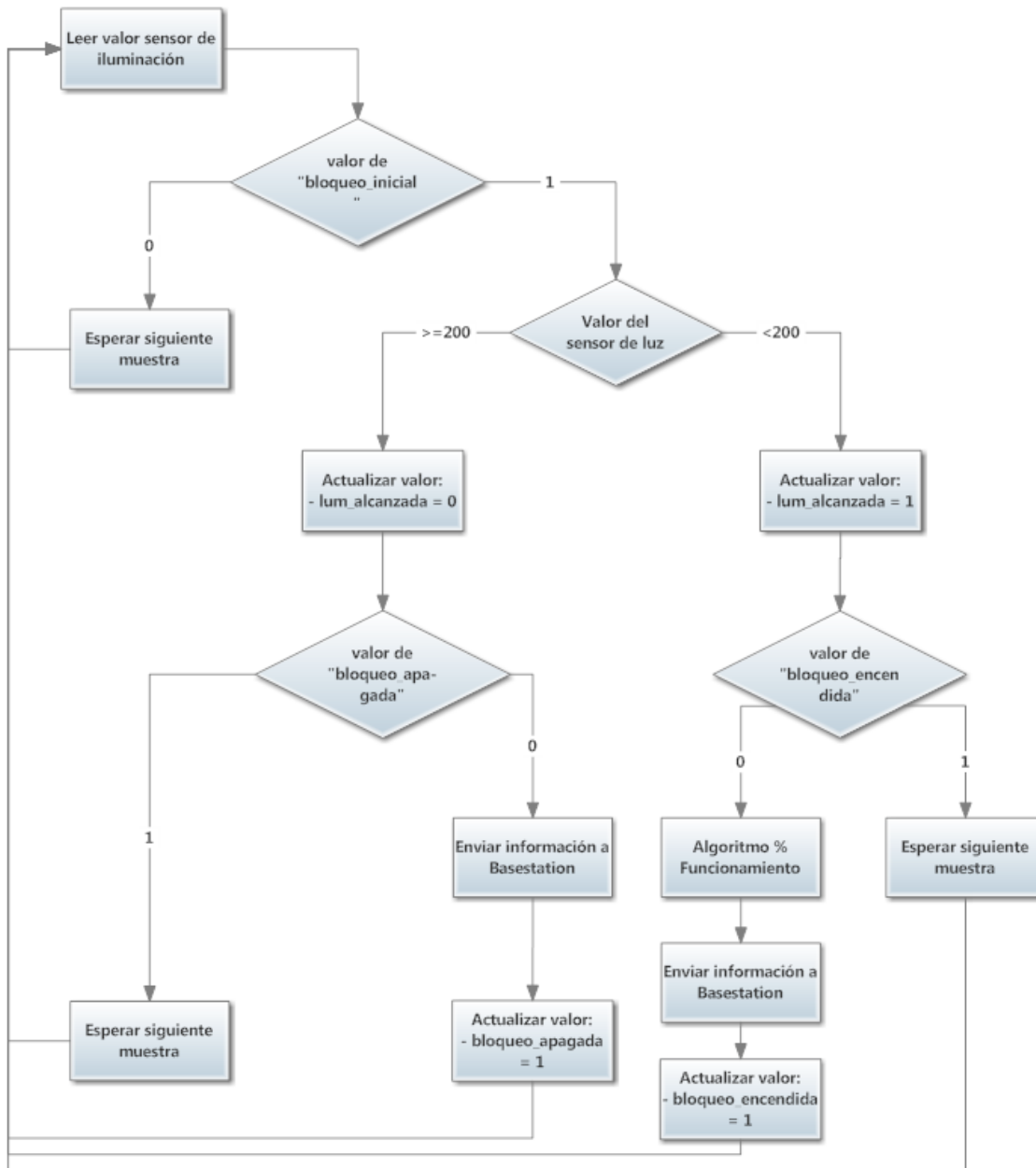


Figura 49. Diagrama de funcionamiento de la lectura de los sensores

A continuación se muestra el código correspondiente a esta parte:

```
private IConditionListener lightListener = new IConditionListener()
{
    public void conditionMet(SensorEvent evt, Condition condition)
    {
    }
};

private Condition monitorizacionLuminosidad = new Condition(light,
lightListener, SAMPLE_INTERVAL)
```



```
{
    public boolean isMet(SensorEvent evt)
    {
        if (bloqueo_inicial == 0) // valdrá 0 una vez inicializada la mota con
SW 2
        {
            int val = ((LightSensorEvent) evt).getValue();
            if (val >= 200)
            {
                lum_alcanzada = 0;
                if (bloqueo_apagada ==0)
                {
                    try
                    {
                        bloqueo_apagada = 1;
                        bloqueo_encendida = 0;
                        rCon = (DatagramConnection)
Connector.open("radiogram://broadcast:" + DATA_SINK_PORT); // Basestation
estará escuchando en ese puerto
                        dg = rCon.newDatagram(rCon.getMaximumLength());
                        dg.reset();
                        now = System.currentTimeMillis();
                        dif = now-start;
                        start = now;
                        dg.writeInt(0);
                        cal.setTime(new Date(now));
                        ts = cal.get(Calendar.YEAR) + "-" +
(1 + cal.get(Calendar.MONTH)) + "-" +
cal.get(Calendar.DAY_OF_MONTH) + " " +
cal.get(Calendar.HOUR_OF_DAY) + ":" +
cal.get(Calendar.MINUTE) + ":" +
cal.get(Calendar.SECOND);
                        dg.writeUTF(ts);
                        dg.writeInt(porcentaje_funcionamiento);
                        porcentaje_funcionamiento=0;
                        dg.writeInt(porcentaje_funcionamiento); //% funcionamiento
anterior
                        enciendeLEDS2(8,AZUL);
                        dg.writeLong(dif);
                        rCon.send(dg);
                        rCon.close();
                        dg.reset();
                    }
                }
            }
        }
        catch (IOException ioe)
        {
        }
    }
}
```




```
    }
    return true; // Luminosidad por encima de lo necesario para que
las farolas se enciendan
    }
    else
    {
        lum_alcanzada = 1;
        bloqueo_apagada=0;
        if (bloqueo_encendida==0)
        {
            try
            {
                bloqueo_encendida=1;
                rCon = (DatagramConnection)
Connector.open("radiogram://broadcast:" + DATA_SINK_PORT);
                dg = rCon.newDatagram(rCon.getMaximumLength());
                dg.reset();
                now = System.currentTimeMillis();
                dif = now-start;
                start = now;
                dg.writeInt(0);
                cal.setTime(new Date(now));
                ts = cal.get(Calendar.YEAR) + "-" +
                    (1 + cal.get(Calendar.MONTH)) + "-" +
                    cal.get(Calendar.DAY_OF_MONTH) + " " +
                    cal.get(Calendar.HOUR_OF_DAY) + ":" +
                    cal.get(Calendar.MINUTE) + ":" +
                    cal.get(Calendar.SECOND);
                dg.writeUTF(ts);
                if (presencia==1)
                {
                    dg.writeInt(porcentaje_funcionamiento); // % funcionamiento
anterior tramo
                    porcentaje_funcionamiento=100;
                    dg.writeInt(porcentaje_funcionamiento); //% funcionamiento
tramo actual
                    dg.writeLong(dif);
                }
            }
            else
            {
                dg.writeInt(porcentaje_funcionamiento); // % funcionamiento
anterior tramo
                if (vecinos_ON==0)
                {
                    porcentaje_funcionamiento=25;
                }
            }
        }
    }
}
```



```
        {
            porcentaje_funcionamiento=75;
        }
        dg.writeInt(porcentaje_funcionamiento);  %% funcionamiento
tramo actual
        dg.writeLong(dif);
    }
    rCon.send(dg);
    rCon.close();
    analizar(presencia,1);
}
catch (IOException ioe)
{
}
}
return false;
}
}
else
{
return false;
}
}
};
```

Tal y como se ha citado, comenzarán a ser efectivas las medias y funcionamiento una vez inicializada la SunSPOT con el switch derecho. Dicha inicialización sigue el siguiente esquema:

- Una vez pulsado el switch derecho, se abrirán 2 DatagramConnection, uno de entrada y uno de salida.
 - El de salida irá contra un broadcast al puerto contra el que esté escuchando la basestation, en este caso, puerto 69. Con esto se conseguirá que en caso de tener que instalar varias basestation, poder generalizar y que la conexión se haga contra la que tenga a su alcance.
 - En entrada, quedará esperando a recibir los datos en un puerto determinado, en este caso, puerto 99.
- La SunSPOT enviará un datagram solicitando los datos. Para que la basestation pueda diferencia si se trata de una

petición de datos para inicializarse, o un envío de información, proceso que se explicará más adelante, se sigue el siguiente protocolo para la estructura del mensaje:

- Solicitud de información:

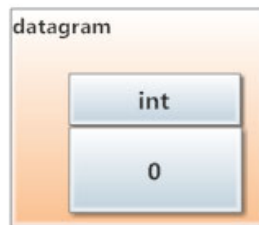


Figura 50. Formato del datagram de Solitiud de Información

- Envío de información:

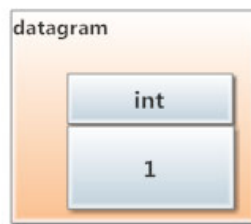


Figura 51. Formato del datagram de envío de Información

- La Basestation, al recibir el datagrama, leerá el valor y procederá a buscar la información. La puesta en marcha de la Basestation, algoritmo de búsqueda de vecinos y envío de información a la SunSPOT se detallarán en apartados posteriores.
- La SunSPOT recibirá un datagrama con el siguiente formato:
 - Un **integer** con el número de vecinos que ha encontrado.
 - Un **string** por cada uno de los vecinos enviados.
- Su representación es la siguiente:



Figura 52. Formato datagrama recepción de datos

- La SunSPOT leerá el primer valor del datagrama, para después, lea tantas veces como su valor indique, e ir guardando cada dirección MAC en un array. Este array queda almacenado en la SunSPOT para su posterior uso a la hora de abrir las conexiones con los vecinos para el envío de información.
- A continuación se muestra el código correspondiente:
 - Solicitud a la Basestation de la información (código de la clase Mota):

```
public void switchReleased (SwitchEvent evt)
{
    if (evt.getSwitch()==sw1) //Pulsador Izquierdo -> Cambio en
presencia
    {
        .
        . //Se explicará más adelante
        .
    }
}

else
{
    if (evt.getSwitch()==sw2) //Pulsador derecho activado
    {
        DatagramConnection rRes = null;
        Datagram rg = null;
        String dir_mac = null;
        try
        {
            rCon = (DatagramConnection) Connector.open
("radiogram://broadcast:" + DATA_SINK_PORT); //Conexión de salida
            rRes = (DatagramConnection)
Connector.open("radiogram://:99"); // Conexión de entrada
            rg = rRes.newDatagram(rRes.getMaximumLength()); //Se
crea el datagrama
            rg.reset();
            dg = rCon.newDatagram(rCon.getMaximumLength());
```



```
        dg.reset();
        dg.writeInt(1);
        //Se escribe 1 en el datagram, correspondiente a petición de información
        rCon.send(dg);
        System.out.println("Esperando respuesta...");
        rRes.receive(rg);
        //Se envía la información y se queda esperando a la recepción de la
        respuesta.
```

- Recepción de la solicitud en la Basestation y envío de respuesta (Código en clase Base)

```
public class Base {
    .
    .
    .
    private static final int DATA_SINK_PORT = 69;
    private DatagramConnection rCon = null;
    private RadiogramConnection auxCon = null;
    .
    .
    .
    public void estado_recepcion() throws Exception
    {
        .
        .
        .
        rCon = (DatagramConnection) Connector.open("radiogram://:" +
DATA_SINK_PORT);
        dg = rCon.newDatagram(rCon.getMaximumLength());

        while (true) {
            try {
                rCon.receive(dg);
                tipo_pet = dg.readInt();
                if (tipo_pet == 1) //Petición para conocer los vecinos
                {
                    mac = dg.getAddress();
                    .
                    .
                    /*Algoritmo de búsqueda de vecinos. Será expuesto más adelante*/
                    .
                    .
                    auxCon = (RadiogramConnection)
Connector.open("radiogram://" + mac + ":99");
                    auxCon.setTimeout(-1);
```



```
sg = auxCon.newDatagram(auxCon.getMaximumLength());
sg.writeInt(contador_vecinos);
for (int i=0; i<contador_vecinos;i++)
{
    sg.writeUTF(aux_macs[i]);
}
auxCon.send(sg);

auxCon.close();
//auxCon.close();
for (int i=0; i<contador_vecinos;i++)
{
    aux_macs[i]=null;
}
contador_vecinos = 0;
}
}
```

- Recepción en la SunSPOT de la información solicitada:

```
public void switchReleased (SwitchEvent evt)
{
    if (evt.getSwitch()==sw1) //Pulsador Izquierdo -> Cambio en
presencia
    {
        .
        .
        .
        rRes.receive(rg); // Se reciben los datos
        num_vecinos = rg.readInt();//Leemos el número de
vecinos
        listaVecinos = new String[num_vecinos];
        for (int i=0; i<num_vecinos;i++) //Vamos guardando
la mac de cada vecino
        {
            dir_mac = rg.readUTF();
            listaVecinos[i]=dir_mac;
        }
        rRes.close();
        rCon.close();

        //Enviamos a la base la información de la situación actual para que sea
almacenada en la base de datos.
        rCon = (DatagramConnection) Connector.open(
            "radiogram://broadcast:" + DATA_SINK_PORT);
        dg = rCon.newDatagram(rCon.getMaximumLength());
        dg.reset();
        now = System.currentTimeMillis();
    }
}
```



```
dif = now-start;
start = now;
dg.writeInt(0);
cal.setTime(new Date(now));
ts = cal.get(Calendar.YEAR) + "-" +
    (1 + cal.get(Calendar.MONTH)) + "-" +
    cal.get(Calendar.DAY_OF_MONTH) + " " +
    cal.get(Calendar.HOUR_OF_DAY) + ":" +
    cal.get(Calendar.MINUTE) + ":" +
    cal.get(Calendar.SECOND);
dg.writeUTF(ts);
dg.writeInt(porcentaje_funcionamiento);
porcentaje_funcionamiento=25;
dg.writeInt(porcentaje_funcionamiento);
enciendeLEDS(2,VERDE);
dg.writeLong(dif);
rCon.send(dg);
rCon.close();
dg.reset();
bloqueo_inicial=0; //Comienza a funcionar la lectura
del sensor de luminosidad.
iniciarOperativa(); //Esta función se explicará más
adelante.
```

- Una vez realizado este proceso, la SunSPOT quedaría preparada para funcionar, conociendo de esta forma, a qué SunSPOTs deberá enviar información cuando así se precise. A continuación se muestra un diagrama que resume las acciones llevadas a cabo en este proceso de puesta en marcha de la SunSPOT.

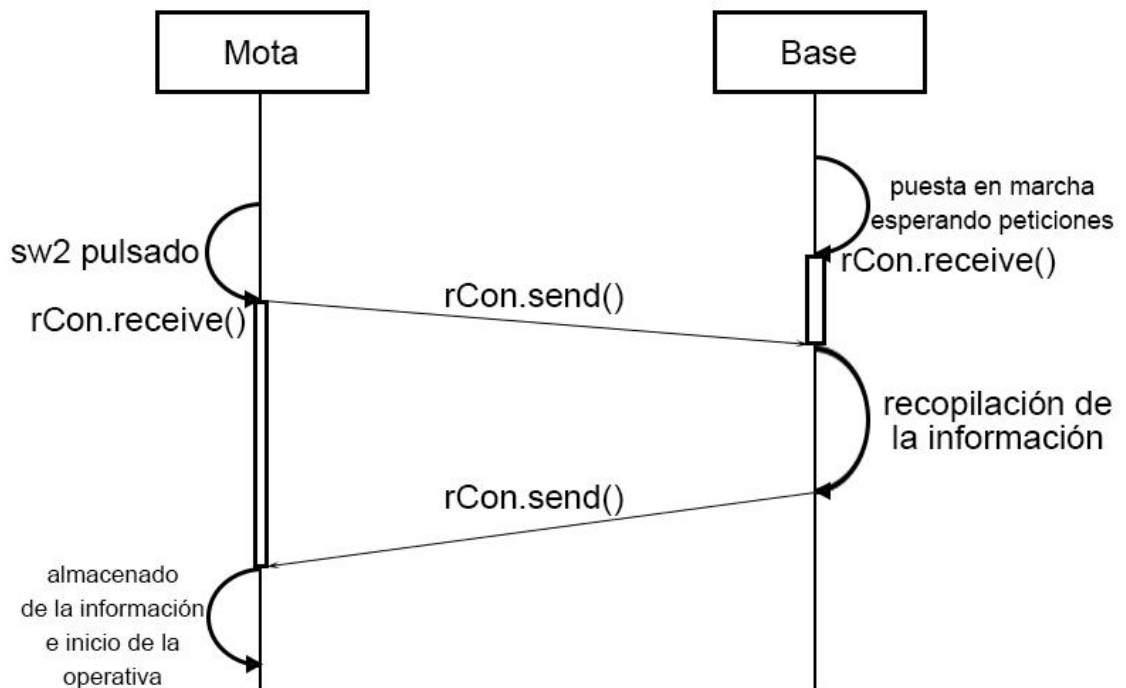


Figura 53. Diagrama del proceso de Puesta en Marcha de la Sun SPOT

En los siguientes sub-apartados se van a explicar las acciones necesarias para completar la secuencia anteriormente descrita:

- Asignación de identificadores a cada SunSPOT.
- **Estructura de la base de datos** donde se almacena la información.
- **Configuración de la Basestation** para el acceso a la Base de Datos.
- Algoritmo para seleccionar las vecindades.

5.4.2.1. Asignación de identificadores

Cada una de las farolas, con su correspondiente SunSPOT, tendrá 3 valores que la identificarán unívocamente del resto:

- **Id_calle** : Identificador del fragmento de calle.
- **Id_farola**: Identificador de la farola.
- **Mac**: Dirección MAC de la SunSPOT.

Podrá haber varias farolas con el mismo ID_farola, pero si se da este caso, estarán en calles distintas, cuyos identificadores sí serán únicos. De esta forma la dupla formada por ID_calle , ID_farola, será única e identificará posicionalmente a cada una de las farolas.

El método para asignar los identificadores a las calles (número desde el 1), es indiferente, siempre que sea único. Sin embargo, para que el algoritmo de descubrimiento de vecinos sea correcto, la asignación de identificadores "ID_farola" debe de seguir un orden. Para ilustrarlo, se considerará la siguiente zona de la localidad:

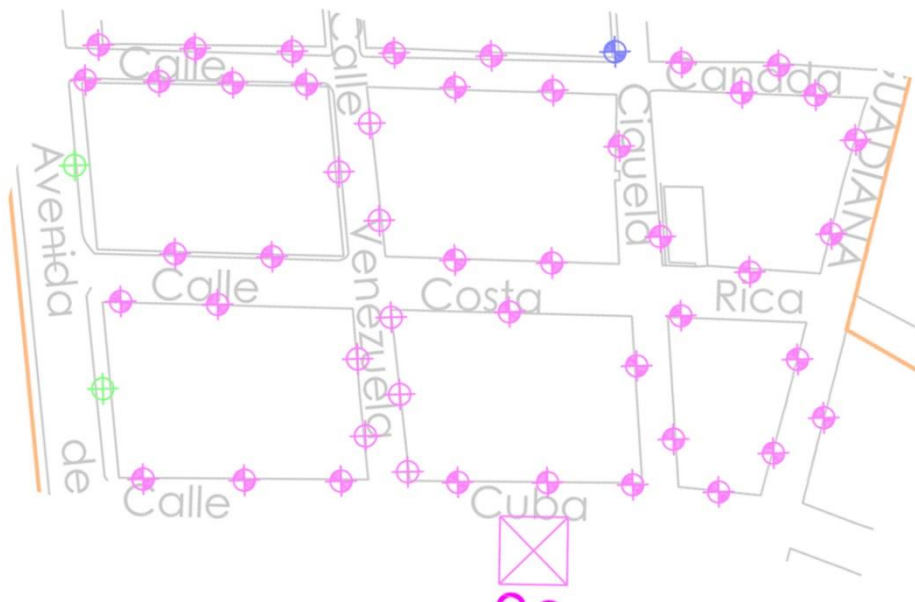


Figura 54. Zona de la localidad de Pedro Muñoz.

- Se elige en primer lugar un fragmento de calle, y se procede a numerar:

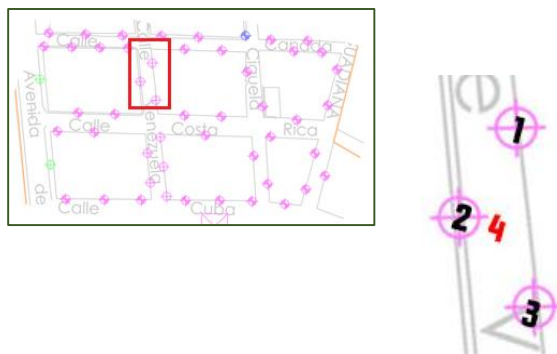


Figura 55. Proceso de asignación de IDs

- Se consideran ahora 2 partes en la calle, parte alta y parte baja. La baja será la esquina cercana a la SunSPOT con id_menor y la parte alta será la esquina con valor de SunSPOT mayor. De esta forma:
 - En las calles que confluyen con la parte baja, se comienza a numerar de menor a mayor:

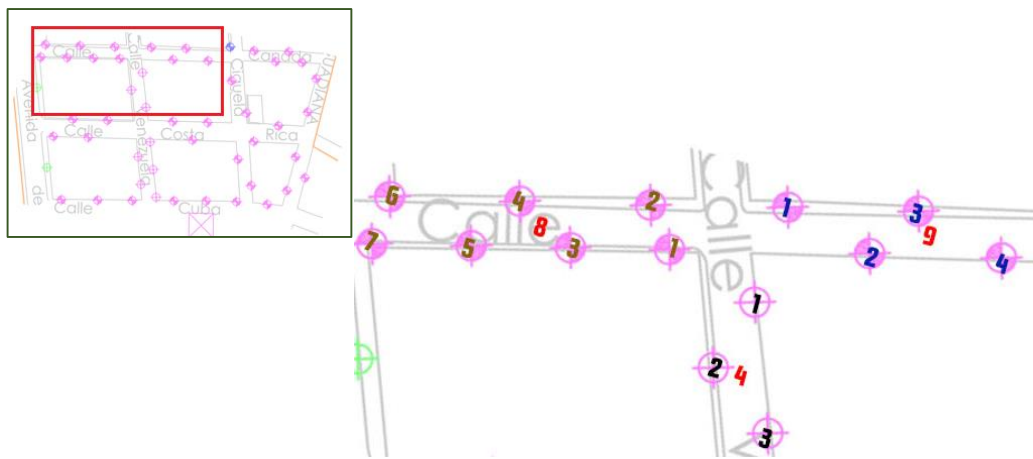


Figura 56. Proceso de asignación de IDs

- De la misma forma, en las calles que confluyen con la parte alta, se comienza a numerar de mayor a menor:

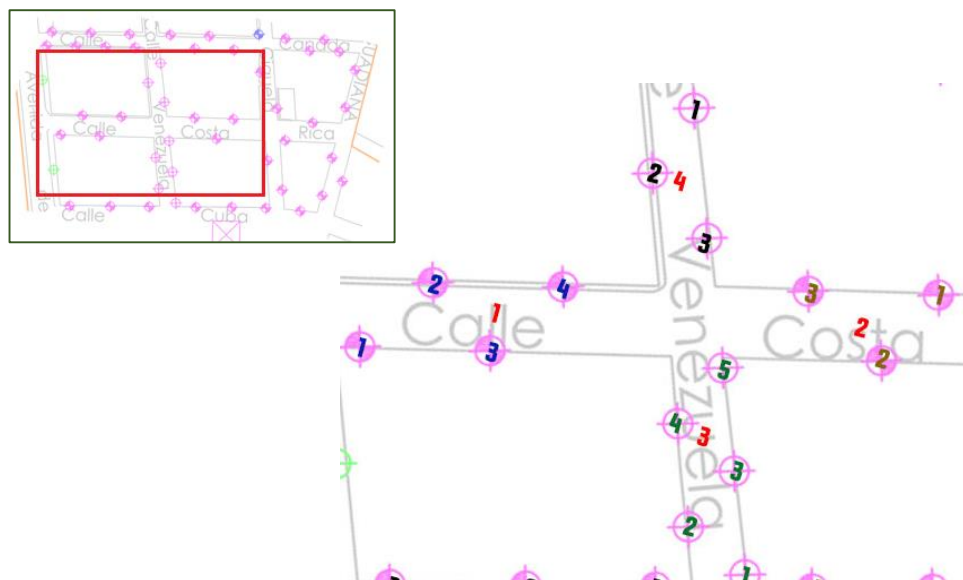


Figura 57. Proceso de asignación de IDs

- Siguiendo este proceso, las SunSPOTs y las farolas de todas las calles ejemplificadas quedaría de la siguiente forma:

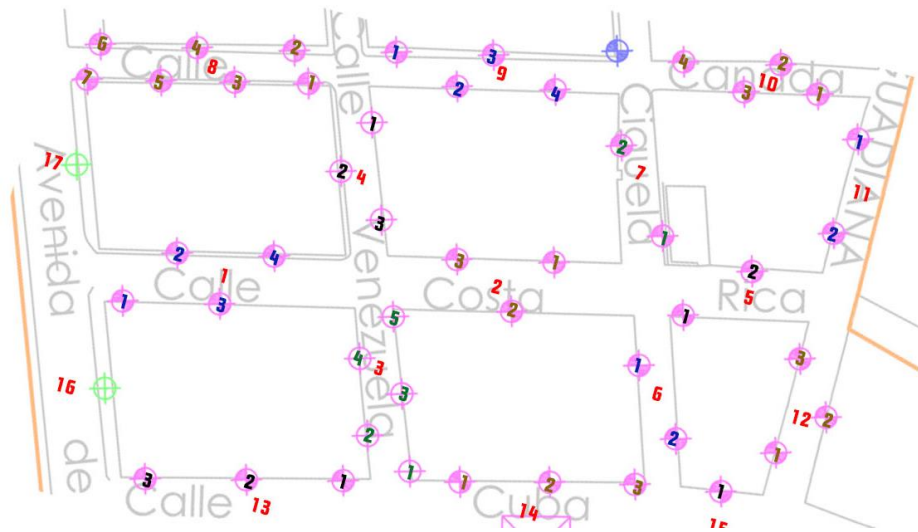


Figura 58. Proceso de asignación de IDs

5.4.2.2. Estructura de la Base de Datos

Tal y como se ha comentado en apartados anteriores, la Base de Datos a utilizar correrá en MySQL. Dicha Base de Datos contendrá la información necesaria para:

- Tener un registro de todas las farolas y SunSPOT pertenecientes al sistema.
- Recibir las consultas necesarias para determinar las vecindades de cada una de las SunSPOTs.
- **Almacenar la información** referente al consumo y funcionamiento con fines estadísticos.

La estructura de la Base de Datos y sus correspondientes tablas será la siguiente:

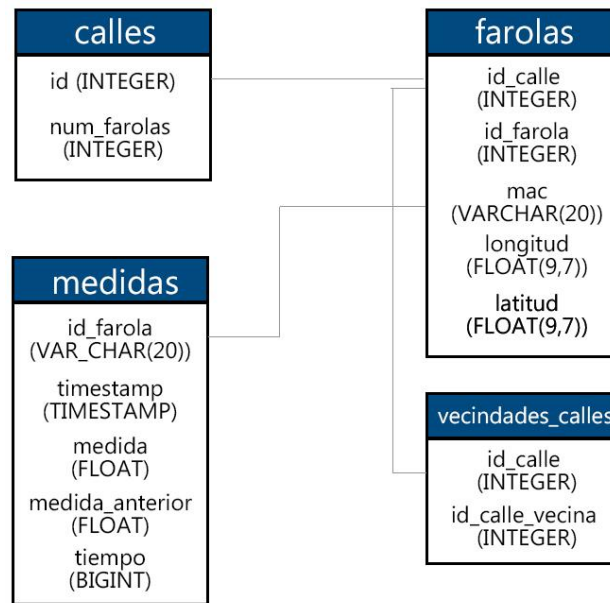


Figura 59. Estructura de la Base de Datos

- **Tabla “calles”**, que contendrá la información referente a las calles:
 - **Id (INTEGER)**: Identificador de la calle.
 - **num_farolas (INTEGER)**: Número de farolas que hay en la calle.
- **Tabla “farolas”**, que contendrá la información referente a cada farola:
 - **Id_calle (INTEGER)**. Identificador de la calle.
 - **Id_farola (INTEGER)**. Identificador de la farola.
 - **Mac (VARCHAR(20))**. Mac de la SunSPOT.
 - **Longitud (FLOAT(9,7))**. Longitud de la farola en el mapa en grados.
 - **Latitud (FLOAT(9,7))**. Latitud de la farola en el mapa en grados.
- **Tabla “medidas”** que contendrá las medidas recibidas por las SunSPOTs al detectar o dejar de detectar presencia:
 - **Id_farola (VARCHAR(20))**. MAC de la SunSPOT.



- **Timestamp (TIMESTAMP):** Fecha y hora de recogida de la información.
- **Medida (FLOAT):** Intensidad a la que pasa a funcionar la farola en el momento de la inserción de la entrada.
- **Medida_anterior (FLOAT):** Intensidad con la que estaba funcionando la farola antes de la inserción de la entrada.
- **Tiempo (BIGINT):** Tiempo en milisegundos que ha estado funcionando la farola con la intensidad anterior.
- **Tabla “vecindades calles”,** que contendrá las vecindades que hay entre las distintas calles:
 - **Id_calle (INTEGER):** Identificador de la calle.
 - **Id_calle_vecina (INTEGER):** Identificador de la calle con la que es vecina id_calle.
 - **Alto_bajo (INTEGER):** Identificará si la vecindad existe entre la parte alta (IDs más altos) de la calle, o por la parte baja, necesaria para ejecutar correctamente el algoritmo de descubrimiento de vecinos que se describirá más adelante.
 -

5.4.2.3. Configuración de la Basestation para el acceso a la Base de Datos

Tal y como se comentó en puntos anteriores, para poder interactuar entre un programa escrito en JAVA y una Base de Datos, es necesario un driver denominado JDBC. Dicho driver permite disponer de los métodos y clases necesarios para poder conectar con una base de datos, y así poder realizar las consultas y modificaciones necesarias sobre ella.

Para configurar el acceso en la clase Base, encargada de interactuar con la Base de Datos, hay que realizar las siguientes acciones:



```
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import java.sql.SQLException;

public class Base {
    private static final String DATABASE_URL =
"jdbc:mysql://localhost:3306/";
    private static final String DATABASE_NAME = "GESTION_FAROLAS";
    private static final String DATABASE_USER = "root";
    private static final String DATABASE_PASSWORD = "1234";
    private static final String JDBC_DRIVER = "com.mysql.jdbc.Driver";
    private static final String TABLA_FAROLAS = "farolas";
    private static final String TABLA_CALLES = "calles";
    private static final String TABLA_MEDIDAS = "medidas";
    private static final String TABLA_VECINDADES_CALLES =
"vecindades_calles";
    private void inicializacion() throws Exception{
        System.out.println("Inicializando BBDD");
        try {
            Class.forName(JDBC_DRIVER);
            String url = DATABASE_URL + DATABASE_NAME;
            dbCon = DriverManager.getConnection(url, DATABASE_USER,
                DATABASE_PASSWORD);
            System.out.println("URL: " + url);
            System.out.println("Conectado: " + dbCon);
        } catch (Exception e) {
            System.err.println("Fallo en la inicialización " + e);
            System.err.println("Error conectando a la BBDD");
            throw e;
        }
        try {
            stmt = dbCon.createStatement();
            stmt2 = dbCon.createStatement();
            stmt3 = dbCon.createStatement();
            stmt4 = dbCon.createStatement();
            query_insercion = dbCon.createStatement();
            System.out.println("Inicialización Correcta");
        } catch (Exception e) {
            System.err.println("Fallo en la inicialización " +
e.getMessage());
            throw e;
        }
    }
}
```



Si todo funciona correctamente, la Basestation quedaría de esta forma conectada ya a la Base de Datos, y creados los diferentes Statement para operar contra ella.

5.4.2.4. Algoritmo de selección de vecinos

En este apartado se explicará el algoritmo seguido para escoger qué farolas serán vecinas de otras. La vecindad en este caso se establece para que exista una luminosidad gradual a lo largo de la calle, es decir, que no sólo esté funcionando al máximo de potencia la farola que ha detectado movimiento, estando el resto prácticamente apagadas, sino que se buscará una gradualidad en la iluminación, estando la que ha detectado presencia funcionando al máximo de potencia, y las 2 farolas vecinas a un 75 %. Dicho funcionamiento ya se explicó con más detalle en el apartado "4.3.6.2. Funcionamiento de la infraestructura".

La principal dificultad para desarrollar dicho algoritmo se encuentra en las intersecciones entre calles, ya que es necesario iluminar farolas de cada una de ellas, para que el sujeto móvil pueda tener iluminación suficiente para dirigirse a cualquier destino.

A continuación se ofrece una explicación del proceso, ilustrado con sus correspondientes diagramas de flujo:

- Una vez se reciba en la Basestation la petición por parte de la SunSPOT para conocer cuáles son sus vecinos, la Basestation obtendrá la dirección MAC de la SunSPOT solicitante, para así poder realizar las consultas necesarias a la Base de Datos. Dependiendo del número de farolas que se encuentren en la calle, se operará se manera diferente:

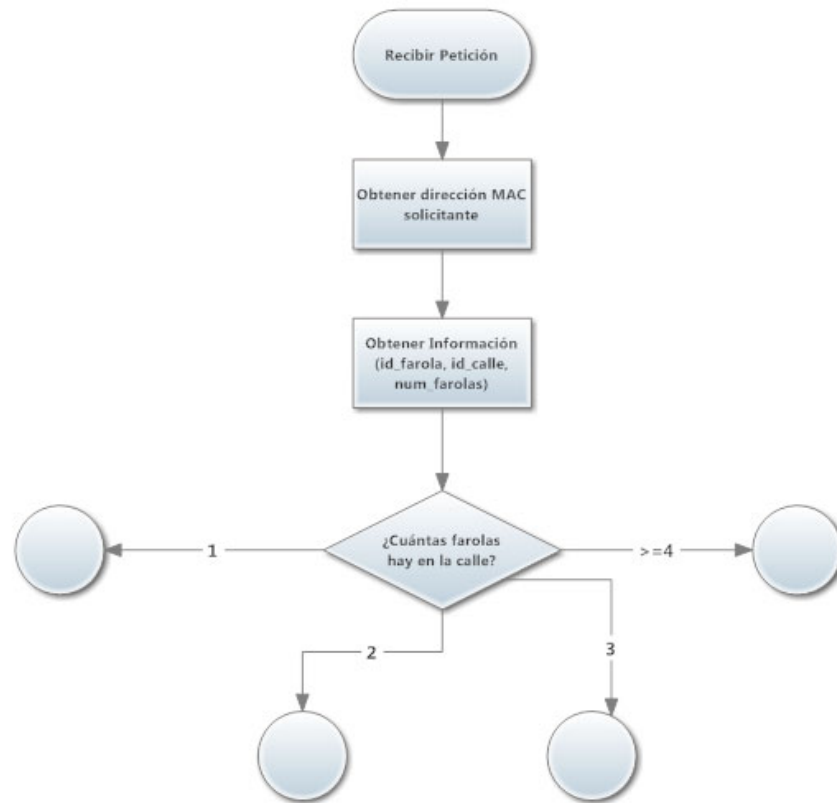


Figura 60. Algoritmo de selección de vecinos.

El código correspondiente a esta parte es el siguiente:

```
public void estado_recepcion() throws Exception
{
    .
    .
    .
    while (true) {
        try {
            rCon.receive(dg);
            tipo_pet = dg.readInt();
            if (tipo_pet == 1) //Petición para conocer los vecinos
            {
                mac = dg.getAddress(); //Obtenemos la MAC del
solicitante

                res = stmt.executeQuery("SELECT * " +
                    "from " + TABLA_FAROLAS +
                    " WHERE mac LIKE \'%" + mac + "%\'"); //Query para
obtener la información sobre la mota
                while (res.next())
                {
                    id_farola = res.getInt("id_farola"); //Obtenemos el
ID de la farola
                    mac_far = res.getString("mac"); //Obtenemos la MAC
```




```
de la calle
    id_calle = res.getInt("id_calle"); //obtenemos el ID
}
res = stmt.executeQuery ("SELECT num_farolas from " +
TABLA_CALLES + " WHERE id=" + id_calle);
//Query para obtener el número de farolas que hay en la
calle
while (res.next())
{
    num_farolas_calle = res.getInt("num_farolas");
//Recuperamos el número de farolas en la calle
}
aux_macs = new String[16]; //Array para ir almacenando
las MACs de los vecinos
switch (num_farolas_calle)
{
```

- **En la calle hay una farola:** En este caso, se obtendrán las calles que confluyen con ella, por ambos extremos. En caso de haber 1 o 2 farolas en la calle vecina, se obtienen la o las direcciones MAC, ya que entra dentro de la directriz de obtener 2 vecinos por cada lado. En caso de haber 3 o más farolas, hay que diferenciar si la conexión de la calle por ese extremo es en la parte alta de la calle, es decir, donde están las MACs altas, o por el contrario, si se trata de la parte baja, donde están las MACs con menor ID.

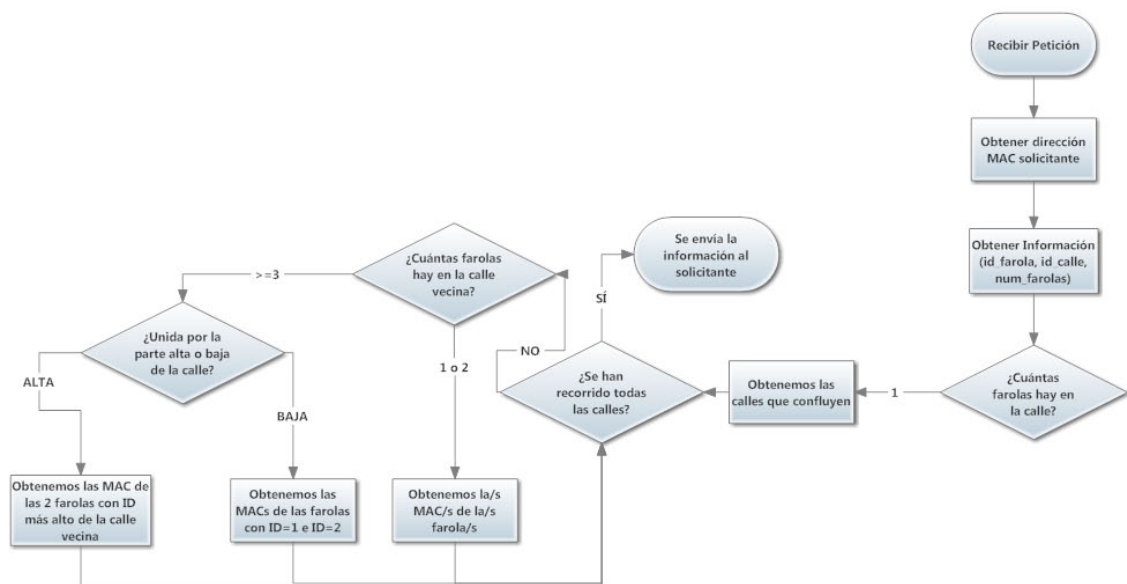


Figura 61. Algoritmo de selección de vecinos.



El código correspondiente a esta parte es el siguiente:

```

case 1: //Solo hay una farola en la calle y cogera todas las vecindados
tanto altas y bajas de la bbdd
id_calle_vecina, alto_bajo      res      =      stmt.executeQuery      ("SELECT
id_calle="+id_calle);          from      "+TABLA_VECINDADES_CALLES+"      WHERE

vecinas                        while (res.next()) //Vamos recorriendo las calles

                                {
                                id_calle_vecina      =
res.getInt("id_calle_vecina");  aux_alto_bajo = res.getInt("alto_bajo");
                                tes =stmt2.executeQuery ("SELECT num_farolas
FROM "+ TABLA_CALLES + " WHERE id=" + id_calle_vecina);
                                ses = stmt3.executeQuery ("SELECT mac FROM "
+ TABLA_FAROLAS + " WHERE id_calle="+id_calle_vecina+" ORDER BY mac");
                                while (tes.next()) //Sacamos el número de
                                farolas que hay en la calle vecina.
                                {
                                aux_farolas_calle      =
tes.getInt("num_farolas");      switch (aux_farolas_calle) //Número de
                                farolas en la calle vecina
                                {
                                case 1: //Si en la calle vecina hay
                                una o dos farolas, se cogen directamente
                                case 2:
                                while (ses.next())
                                {
                                aux_macs[contador_vecinos]=ses.getString("mac");
                                contador_vecinos++;
                                }
                                break;
                                default:
                                es parte alta o baja
                                switch(aux_alto_bajo) //Vemos si
                                {
                                case 0: //Parte baja.
                                Cogemos por tanto las farolas con ID 1 y 2
                                les = stmt4.executeQuery
                                ("SELECT mac FROM " + TABLA_FAROLAS + " WHERE id_calle="+id_calle_vecina+" AND
                                (id_farola=1 OR id_farola=2) ORDER BY mac");
                                while (les.next())
                                {
                                aux_macs[contador_vecinos]=les.getString("mac");
                                contador_vecinos++;
                                }
                                break;

```



```
case 1: //Parte alta.
Cogemos las 2 farolas con ID más alto.

        les = stmt4.executeQuery
("SELECT mac FROM " + TABLA_FAROLAS + " WHERE id_calle="+id_calle_vecina+" AND
(id_farola="+(aux_farolas_calle-1)+" OR id_farola="+(aux_farolas_calle)+")
ORDER BY mac");

        while (les.next())
        {
aux_macs[contador_vecinos]=les.getString("mac");

                contador_vecinos++;
        }
        break;
    }
    break;

}

}

break; //Fin de que solo hay una farola en la calle. (Fin case 1)
```

- **En la calle hay dos farolas:** En este caso, obtendremos el ID de la farola.
 - **Si es = 1:** Se obtendrán las calles que confluyen con la que pertenece la farola en la parte BAJA. De estas calles vecinas, se elegirán las SunSPOTs con ID = 1 (si solo hay una farola en esa calle), o las de ID=1 e ID=2 en caso de que haya más de 2. A continuación se muestra el diagrama:

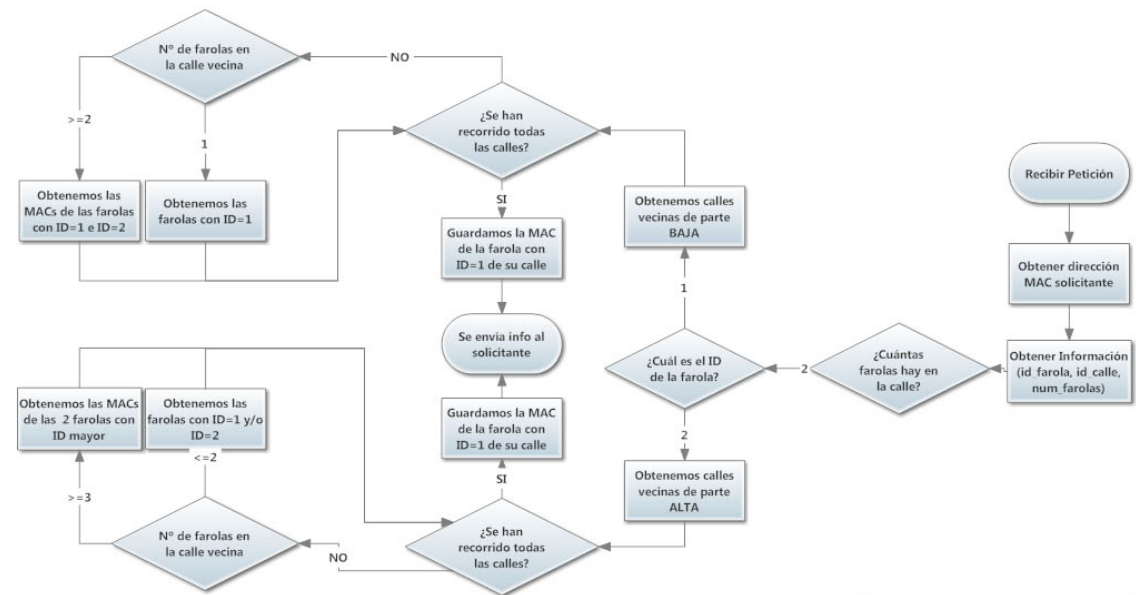


Figura 62. Algoritmo de selección de vecinos.

El código correspondiente a esta parte es el siguiente:

```

case 2: //En caso de que haya 2 farolas, hay que coger las de las calles
vecinas y la otra de la calle propia
        switch(id_farola) //vemos si el ID es 1 o 2 para
coger las vecindades correspondientes
        {
            case 1: //Sacamos las calles vecinas con
alto_bajo=0
                res = stmt.executeQuery ("SELECT
id_calle_vecina,alto_bajo from "+TABLA_VECINDADES_CALLES+"
WHERE
(id_calle="+id_calle+") AND (alto_bajo=0)");
                while (res.next())
                {
                    id_calle_vecina =
res.getInt("id_calle_vecina");
                    tes =stmt2.executeQuery ("SELECT
num_farolas FROM "+ TABLA_CALLES + " WHERE id=" + id_calle_vecina);
                    ses = stmt3.executeQuery ("SELECT
mac FROM " + TABLA_FAROLAS + " WHERE id_calle="+id_calle_vecina+" ORDER BY
mac");
                    while (tes.next())
                    {
                        aux_farolas_calle =
tes.getInt("num_farolas");
                        switch (aux_farolas_calle)
                        {
                            case 1: //Si en la calle
vecina hay una farola, se coge directamente.
                                while (ses.next())
                                {

```



```
aux_macs[contador_vecinos]=ses.getString("mac");
                                contador_vecinos++;
                                }
                                break;

                                default: //En caso de haber más de 1, se cogen las de ID=1 e ID=2
                                les = stmt4.executeQuery
("SELECT mac FROM " + TABLA_FAROLAS + " WHERE id_calle="+id_calle_vecina+" AND
(id_farola=1 OR id_farola=2) ORDER BY mac");
                                while (les.next())
                                {
aux_macs[contador_vecinos]=les.getString("mac");
                                contador_vecinos++;
                                }
                                break;
                                }
                                }
                                break;

                                case 2: //Sacamos las calles vecinas con
alto_bajo=1
                                res = stmt.executeQuery ("SELECT
id_calle_vecina,alto_bajo from "+TABLA_VECINDADES_CALLES+" WHERE
(id_calle="+id_calle+") AND (alto_bajo=1)");
                                while (res.next())
                                {
                                id_calle_vecina =
res.getInt("id_calle_vecina");
                                tes =stmt2.executeQuery ("SELECT
num_farolas FROM "+ TABLA_CALLES + " WHERE id=" + id_calle_vecina);
                                ses = stmt3.executeQuery
("SELECT mac FROM " + TABLA_FAROLAS + " WHERE id_calle="+id_calle_vecina+" ORDER
BY mac");
                                while (tes.next())
                                {
                                aux_farolas_calle =
tes.getInt("num_farolas");
                                switch (aux_farolas_calle)
//Número de farolas en la calle vecina
                                {
                                case 1: //Si en la calle
vecina hay una o dos farolas, se cogen directamente
                                case 2:
                                while (ses.next())
```

```
                                {
aux_macs[contador_vecinos]=ses.getString("mac");
contador_vecinos++;
                                }
                                break;
                                default: //Cogemos las
farolas con las 2 IDs más altas
                                les
                                =
stmt4.executeUpdate ("SELECT mac FROM " + TABLA_FAROLAS + " WHERE
id_calle="+id_calle_vecina+" AND (id_farola="+aux_farolas_calle-1)+" OR
id_farola="+aux_farolas_calle)+" ORDER BY mac");
                                while (les.next())
                                {
aux_macs[contador_vecinos]=les.getString("mac");
contador_vecinos++;
                                }
                                break;
                                }
                                }
                                break;
```

- **En la calle hay 3 farolas:** En este caso, se tratará de una forma u otra dependiendo del ID. La visión general es la siguiente:

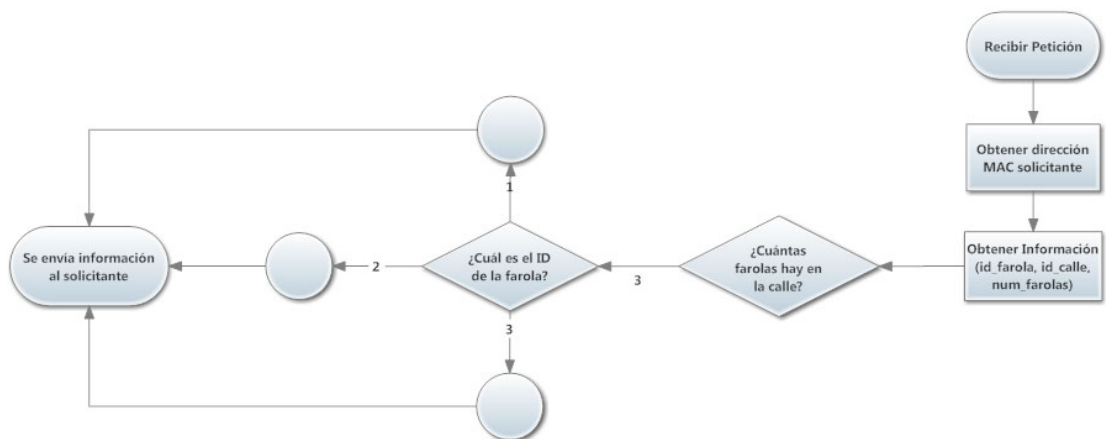


Figura 63. Algoritmo de selección de vecinos.

- A continuación se procede a explicar los 3 procesos:
 - **ID Farola = 1:** Se buscarán las confluencias con calles por la parte baja, y en las cuales, se extraerá, en caso de haber

una única farola, esa misma, o en caso de haber dos, ambas farolas. En caso de haber más farolas en la calle vecina, se cogerán las de ID=1 e ID=2. Además añadiremos las farolas con ID=2 e ID=3 de su propia calle. A continuación el diagrama explicativo:

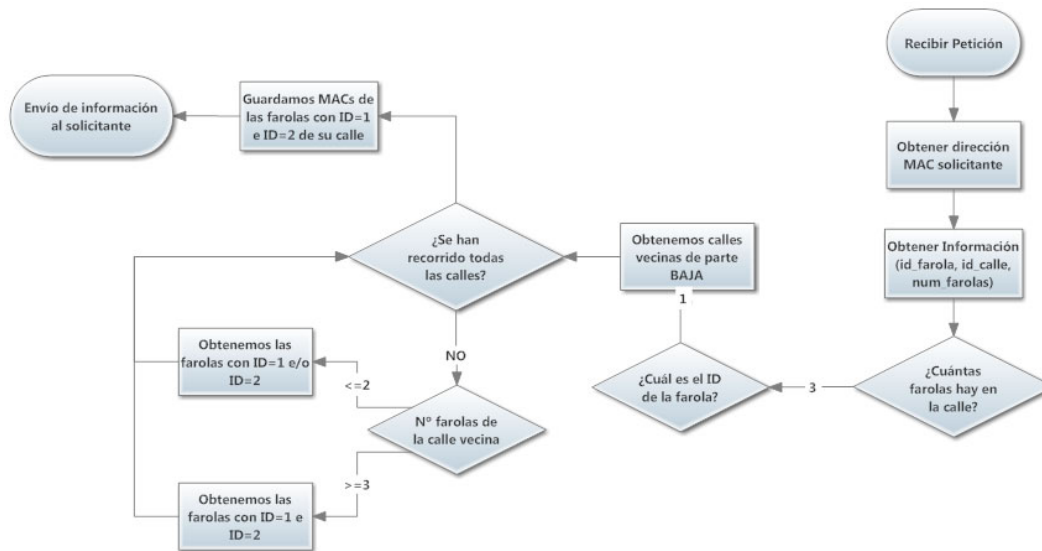


Figura 64. Algoritmo de selección de vecinos

El código queda de la siguiente forma:

```
case 3: // En caso de que haya 3 farolas
    switch(id_farola) //Vemos si el ID es 1,2 o 3
    para coger las vecindades correspondientes
    {
        case 1:
            res = stmt.executeQuery ("SELECT
            id_calle_vecina,alto_bajo from "+TABLA_VECINDADES_CALLES+" WHERE
            (id_calle="+id_calle+") AND (alto_bajo=0)");
            while (res.next())
            {
                id_calle_vecina =
                res.getInt("id_calle_vecina");
                tes =stmt2.executeQuery
                ("SELECT num_farolas FROM "+ TABLA_CALLES + " WHERE id=" + id_calle_vecina);
                ses = stmt3.executeQuery
                ("SELECT mac FROM " + TABLA_FAROLAS + " WHERE id_calle="+id_calle_vecina+" ORDER
                BY mac");
                while (tes.next())
                {
                    aux_farolas_calle =
                    tes.getInt("num_farolas");
                    switch
                    (aux_farolas_calle) //Número de farolas en la calle vecina
```



```

{
    case 1: //Si en
        la calle vecina hay una o dos farolas, se cogen directamente
        case 2:
            while
                (ses.next())
                {
                    aux_macs[contador_vecinos]=ses.getString("mac");
                    contador_vecinos++;
                    break;
                }
            default:
                les = stmt4.executeQuery
                ("SELECT mac FROM " + TABLA_FAROLAS + " WHERE
                id_calle="+id_calle_vecina+" AND (id_farola=1 OR id_farola=2) ORDER BY mac");
                while
                    (les.next())
                    {
                        aux_macs[contador_vecinos]=les.getString("mac");
                        contador_vecinos++;
                    }
                break;
            }
        }

        les = stmt4.executeQuery
        ("SELECT mac FROM " + TABLA_FAROLAS + " WHERE id_calle="+id_calle+" AND
        (id_farola=2 OR id_farola=3) ORDER BY mac");
        while (les.next())
        {
            aux_macs[contador_vecinos]=les.getString("mac");
            contador_vecinos++;
        }
        break;
}

```

- **ID Farola = 2:** Se buscarán las confluencias con calles, tanto por la parte baja, como por la alta, y en las cuales, se extraerá, en caso de haber una única farola, esa misma, o en caso de haber dos, ambas farolas. En caso de haber más farolas en la calle vecina, se cogerán las de ID=1 e ID=2 si la confluencia es por la parte baja, y las dos IDs más altas en



caso de ser la confluencia por la parte alta. Además añadiremos las farolas con ID=1 e ID=3 de su propia calle. A continuación el diagrama explicativo:

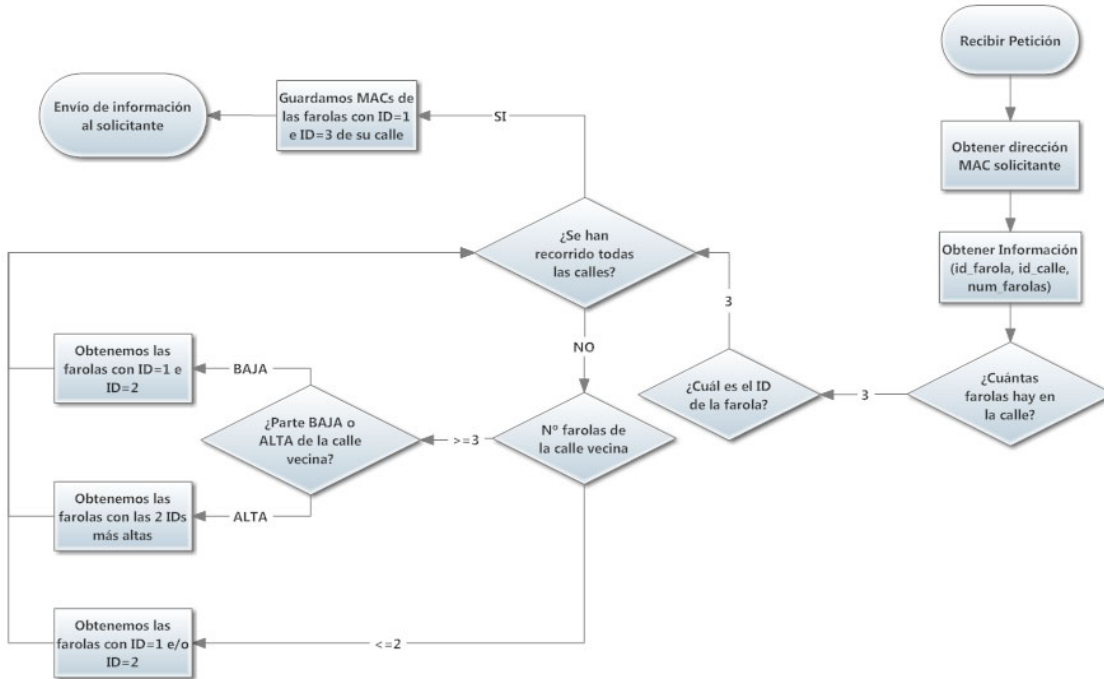


Figura 65. Algoritmo de selección de vecinos

A continuación se expone el código correspondiente a esta parte:

```

case 2:
id_calle_vecina,alto_bajo      from      res      =      stmt.executeQuery      ("SELECT
(id_calle="+id_calle+"");      "+TABLA_VECINDADES_CALLES+"      WHERE

while (res.next())
{
id_calle_vecina      =
res.getInt("id_calle_vecina");
tes      =stmt2.executeQuery
("SELECT num_farolas FROM "+ TABLA_CALLES + " WHERE id=" + id_calle_vecina);
ses      = stmt3.executeQuery
("SELECT mac FROM " + TABLA_FAROLAS + " WHERE id_calle="+id_calle_vecina+" ORDER
BY mac");
while (tes.next())
{
aux_farolas_calle      =
tes.getInt("num_farolas");
aux_alto_bajo      =
res.getInt("alto_bajo");
switch
(aux_farolas_calle) //Número de farolas en la calle vecina
{

```



```

la calle vecina hay una o dos farolas, se cogen directamente
case 1: //Si en
case 2:

while
{
aux_mac[s[contador_vecinos]]=ses.getString("mac");
contador_vecinos++;
}
break;
default:
{
case 0:
les = stmt4.executeQuery ("SELECT mac FROM " + TABLA_FAROLAS + " WHERE
id_calle="+id_calle_vecina+" AND (id_farola=1 OR id_farola=2) ORDER BY mac");
while (les.next())
{
aux_mac[s[contador_vecinos]]=les.getString("mac");
contador_vecinos++;
}
break;
case 1:
les = stmt4.executeQuery ("SELECT mac FROM " + TABLA_FAROLAS + " WHERE
id_calle="+id_calle_vecina+" AND (id_farola="+aux_farolas_calle+" OR
id_farola="+aux_farolas_calle+" ) ORDER BY mac");
while (les.next())
{
aux_mac[s[contador_vecinos]]=les.getString("mac");
contador_vecinos++;
}
break;
}
}
les = stmt4.executeQuery
("SELECT mac FROM " + TABLA_FAROLAS + " WHERE id_calle="+id_calle+" AND
(id_farola=1 OR id_farola=3) ORDER BY mac");

```

```

DEB: "+contador_vecinos);

                                System.out.println("--
                                while (les.next())
                                {
                                aux_mac[s(contador_vecinos)=les.getString("mac");
                                contador_vecinos++;
                                }
                                break;

```

- **ID Farola = 3:** Se buscarán las confluencias con calles por la parte alta, y en las cuales, se extraerá, en caso de haber una única farola, esa misma, o en caso de haber dos, ambas farolas. En caso de haber más farolas en la calle vecina, se cogerán las dos IDs más altas. Además añadiremos las farolas con ID=1 e ID=3 de su propia calle. A continuación el diagrama explicativo:

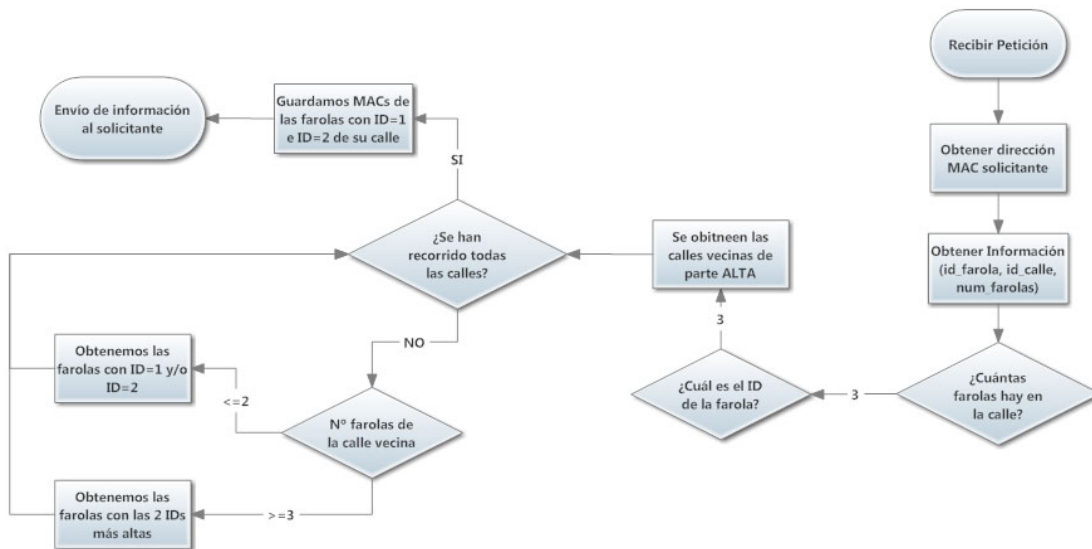


Figura 66. Algoritmo de selección de vecinos

A continuación se expone el código correspondiente:

```

case 3:
id_calle_vecina,alto_bajo      res      =      stmt.executeQuery      ("SELECT
(id_calle="+id_calle+") AND (alto_bajo=1)");
                                while (res.next())
                                {
                                id_calle_vecina      =
res.getInt("id_calle_vecina");

```



```
num_farolas FROM "+ TABLA_CALLES + " WHERE id=" + id_calle_vecina);
tes =stmt2.executeQuery ("SELECT
("SELECT mac FROM " + TABLA_FAROLAS + " WHERE id_calle="+id_calle_vecina+" ORDER
BY mac");
ses = stmt3.executeQuery
while (tes.next())
{
tes.getInt("num_farolas");
aux_farolas_calle =
switch (aux_farolas_calle)
{
case 1: //Si en la calle
case 2:
while (ses.next())
{
aux_macs[contador_vecinos]=ses.getString("mac");
contador_vecinos++;
}
break;
default:
les =
stmt4.executeQuery ("SELECT mac FROM " + TABLA_FAROLAS + " WHERE
id_calle="+id_calle_vecina+" AND (id_farola="+aux_farolas_calle-1)+" OR
id_farola="+aux_farolas_calle+") ORDER BY mac");
while (les.next())
{
aux_macs[contador_vecinos]=les.getString("mac");
contador_vecinos++;
}
break;
}
}
}
les = stmt4.executeQuery ("SELECT
mac FROM " + TABLA_FAROLAS + " WHERE id_calle="+id_calle+" AND (id_farola=1 OR
id_farola=2) ORDER BY mac");
while (les.next())
{
aux_macs[contador_vecinos]=les.getString("mac");
contador_vecinos++;
}
}
break;
```

- **En la calle hay más de 4 farolas:** Por último se expone el caso en el que en la calle de la SunSPOT solicitante, existen más de 4 farolas. La visión general es la siguiente:

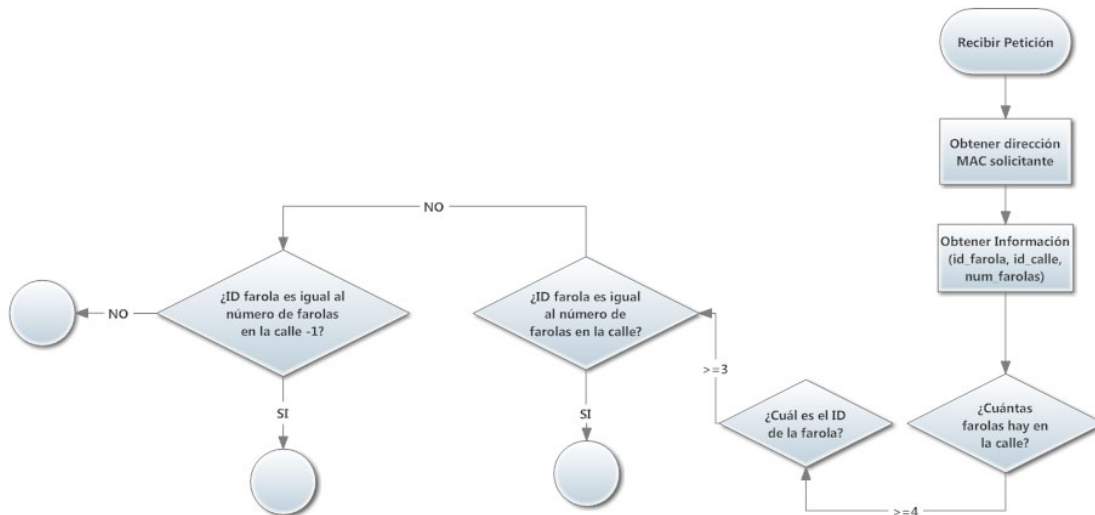


Figura 67. Algoritmo de selección de vecinos

- **ID Farola = 1 o 2:** Se buscarán las confluencias con calles por la parte baja, y en las cuales, se extraerá, en caso de haber una única farola, esa misma, o en caso de haber dos, ambas farolas. En caso de haber más farolas en la calle vecina, se cogerán las de ID=1 e ID=2. Además se añadirán las farolas con ID=2 e ID=3 de su propia calle si su ID es 1, o las farolas con ID=1 e ID=3 si su ID es 2. A continuación el diagrama explicativo:

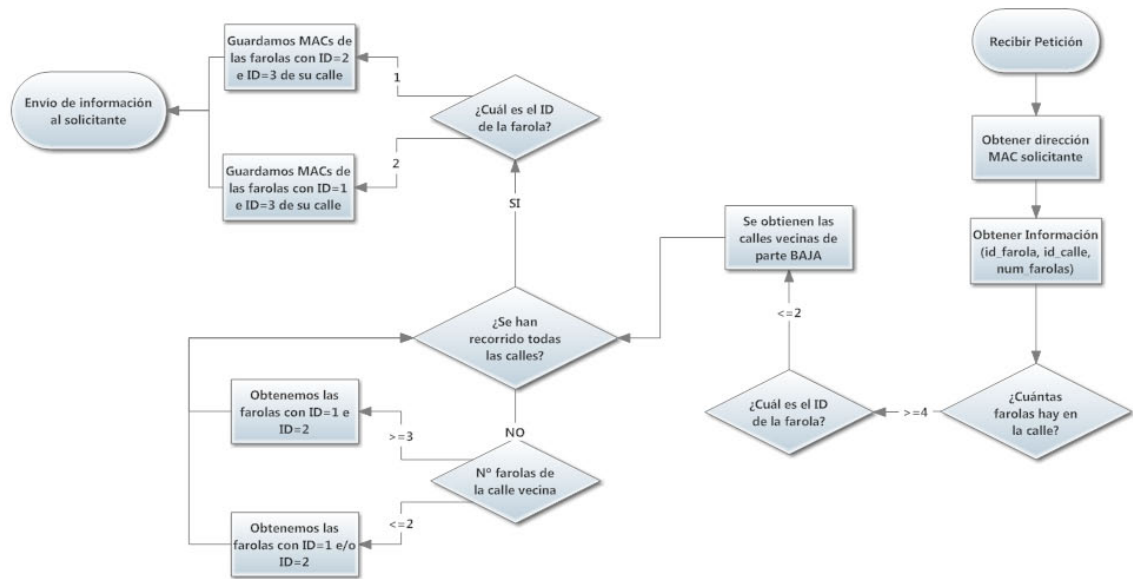


Figura 68. Algoritmo de selección de vecinos

A continuación se muestra el código correspondiente a este caso:

```

default:
corresponda
switch(id_farola) //vemos el ID para operar según
{
case 1:
id_calle_vecina,alto_bajo from res = stmt.executeQuery ("SELECT
(id_calle="+id_calle+") AND (alto_bajo=0)"); "+TABLA_VECINDADES_CALLES+" WHERE
while (res.next())
{
res.getInt("id_calle_vecina"); id_calle_vecina =
tes =stmt2.executeQuery
("SELECT num_farolas FROM "+ TABLA_CALLES + " WHERE id="+ id_calle_vecina);
while (tes.next())
{
tes.getInt("num_farolas"); aux_farolas_calle =
stmt3.executeQuery ("SELECT mac FROM " + TABLA_FAROLAS + " WHERE
id_calle="+id_calle_vecina+" ORDER BY mac");
switch
(aux_farolas_calle) //Número de farolas en la calle vecina
{
case 1: //Si en
la calle vecina hay una o dos farolas, se cogen directamente

```



```
case 2:
    while
    {
(ses.next())
aux_macs[contador_vecinos]=ses.getString("mac");
contador_vecinos++;
    }
    break;
    default:
        les = stmt4.executeQuery ("SELECT mac FROM " + TABLA_FAROLAS + " WHERE
id_calle="+id_calle_vecina+" AND (id_farola=1 OR id_farola=2) ORDER BY mac");
        while
        {
(les.next())
aux_macs[contador_vecinos]=les.getString("mac");
contador_vecinos++;
        }
        break;
    }
}
}
}
les = stmt4.executeQuery
("SELECT mac FROM " + TABLA_FAROLAS + " WHERE id_calle="+id_calle+" AND
(id_farola=2 OR id_farola=3) ORDER BY mac");
while (les.next())
{
aux_macs[contador_vecinos]=les.getString("mac");
contador_vecinos++;
}
break;
case 2:
res = stmt.executeQuery ("SELECT
id_calle_vecina,alto_bajo from "+TABLA_VECINDADES_CALLES+" WHERE
(id_calle="+id_calle+") AND (alto_bajo=0)");
while (res.next())
{
id_calle_vecina =
res.getInt("id_calle_vecina");
```



```
tes =stmt2.executeQuery
("SELECT num_farolas FROM "+ TABLA_CALLES + " WHERE id=" + id_calle_vecina);
ses = stmt3.executeQuery
("SELECT mac FROM " + TABLA_FAROLAS + " WHERE id_calle="+id_calle_vecina+" ORDER
BY mac");
while (tes.next())
{
    aux_farolas_calle =
tes.getInt("num_farolas");
    switch
(aux_farolas_calle) //Número de farolas en la calle vecina
    {
        case 1: //si en
la calle vecina hay una o dos farolas, se cogen directamente
        case 2:
            while
(ses.next())
            {
                aux_macs[contador_vecinos]=ses.getString("mac");
                contador_vecinos++;
            }
            break;
        default:
            les =
stmt4.executeQuery ("SELECT mac FROM " + TABLA_FAROLAS + " WHERE
id_calle="+id_calle_vecina+" AND (id_farola=1 OR id_farola=2) ORDER BY mac");
            while
(les.next())
            {
                aux_macs[contador_vecinos]=les.getString("mac");
                contador_vecinos++;
            }
            break;
    }
}
les = stmt4.executeQuery
("SELECT mac FROM " + TABLA_FAROLAS + " WHERE id_calle="+id_calle+" AND
(id_farola=1 OR id_farola=3 OR id_farola=4) ORDER BY mac");
while (les.next())
{
    aux_macs[contador_vecinos]=les.getString("mac");
    contador_vecinos++;
}
break;
```


- **ID Farola ≥ 3** : Este caso se subdivide en otros 3:
 - **ID Farola = nº farolas en la calle**: Se buscarán las confluencias con calles por la parte alta, y en las cuales, se extraerá, en caso de haber una única farola, esa misma, o en caso de haber dos, ambas farolas. En caso de haber más farolas en la calle vecina, se cogerán las 2 con ID más alta. Además, se obtendrán también las 2 farolas siguientes en sentido descendente. A continuación el diagrama explicativo:

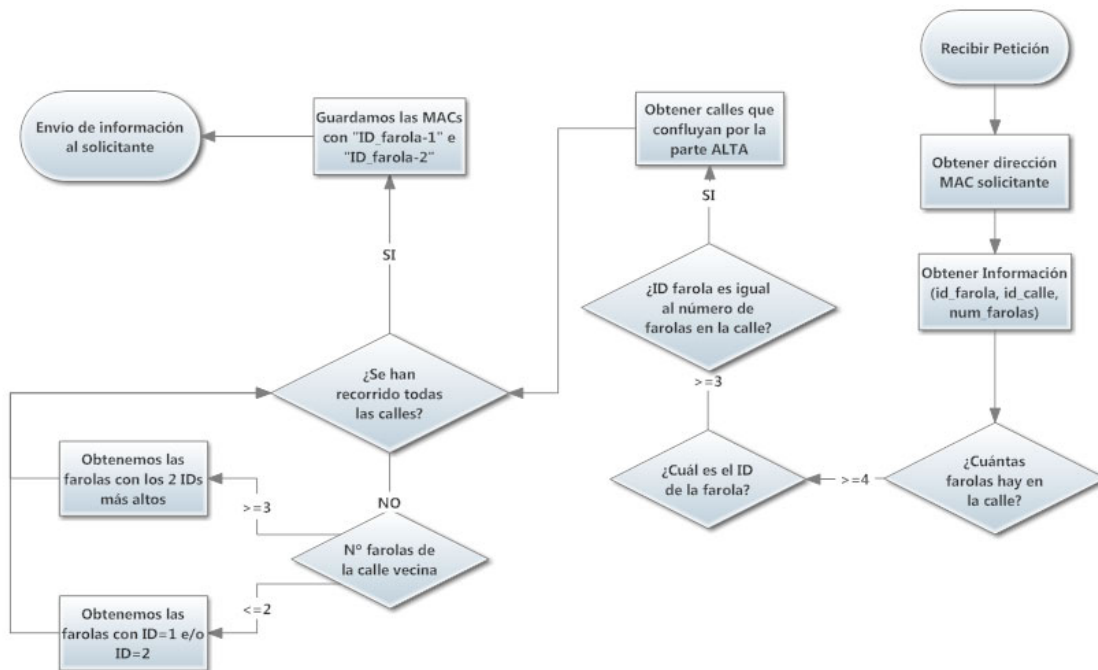


Figura 69. Algoritmo de selección de vecinos

El código correspondiente a esta parte es el siguiente:

```

default:
    if (id_farola==num_farolas_calle)
    {
        System.out.println("DEBUG      SELECT
id_calle_vecina,alto_bajo      from      "+TABLA_VECINDADES_CALLES+"      WHERE
(id_calle="+id_calle+") AND (alto_bajo=1)");
        res = stmt.executeQuery ("SELECT
id_calle_vecina,alto_bajo      from      "+TABLA_VECINDADES_CALLES+"      WHERE
(id_calle="+id_calle+") AND (alto_bajo=1)");
        while (res.next())
        {
            id_calle_vecina      =
res.getInt("id_calle_vecina");
    }
    }
    
```



```
num_farolas FROM "+ TABLA_CALLES + " WHERE id=" + id_calle_vecina);
tes =stmt2.executeQuery ("SELECT
("SELECT mac FROM " + TABLA_FAROLAS + " WHERE id_calle="+id_calle_vecina + "
ORDER BY mac");
ses = stmt3.executeQuery
while (tes.next())
{
aux_farolas_calle =
switch (aux_farolas_calle)
{
case 1: //Si en la calle
case 2:
while (ses.next())
{
aux_macs[contador_vecinos]=ses.getString("mac");
contador_vecinos++;
}
break;
default:
les =
stmt4.executeQuery ("SELECT mac FROM " + TABLA_FAROLAS + " WHERE
id_calle="+id_calle_vecina+" AND (id_farola="+aux_farolas_calle-1)+" OR
id_farola="+aux_farolas_calle+") ORDER BY mac");
while (les.next())
{
aux_macs[contador_vecinos]=les.getString("mac");
contador_vecinos++;
}
break;
}
}
}
les = stmt4.executeQuery ("SELECT
mac FROM " + TABLA_FAROLAS + " WHERE id_calle="+id_calle+" AND
(id_farola="+id_farola-1)+" OR id_farola="+id_farola-2)") ORDER BY mac");
while (les.next())
{
aux_macs[contador_vecinos]=les.getString("mac");
contador_vecinos++;
}
}
```

- **ID Farola = nº farolas en la calle-1:** Se buscarán las confluencias con calles por la parte alta, y en las cuales, se extraerá, en caso de haber una única farola, esa misma, o en caso de haber dos, ambas farolas. En caso de haber más farolas en la calle vecina, se cogerán las 2 con ID más alta. Además, se obtendrán también las 2 farolas siguientes en sentido descendente, así como la siguiente en sentido ascendente (ID_FAROLA-2, ID_FAROLA-1, ID_FAROLA+1).
- Si su ID no se encuentra dentro de este supuesto, significa que no está cerca de una esquina, y por tanto, solo tendría de vecinos a los 2 inmediatamente anteriores, y a los inmediatamente posteriores. A continuación el diagrama explicativo de estos dos últimos casos:

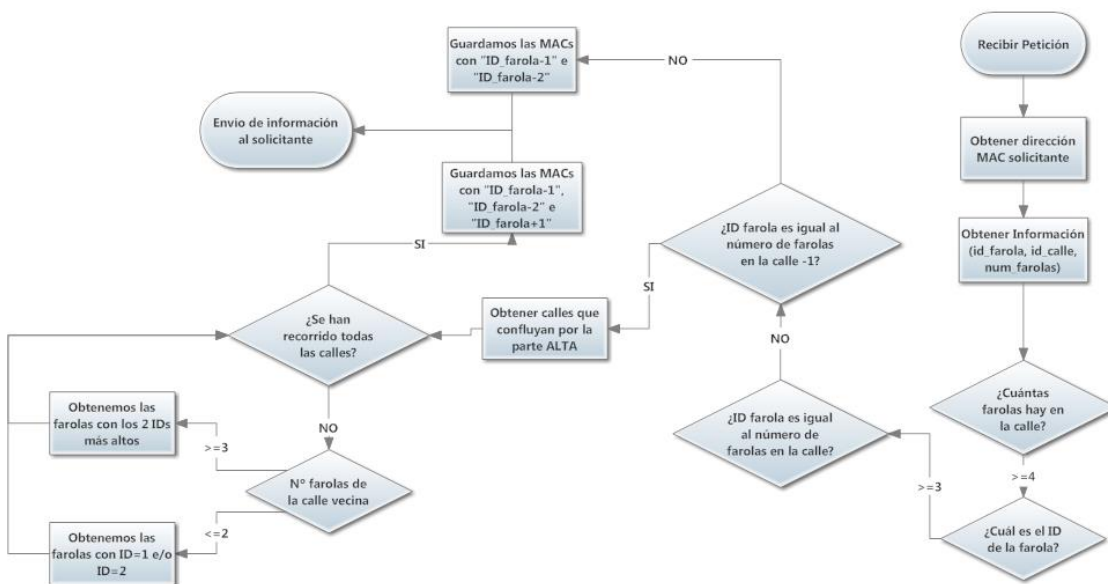


Figura 70. Algoritmo de selección de vecinos

A continuación, el código explicativo:



```
else
    {
        if (id_farola == (num_farolas_calle-
1))
        {
            res = stmt.executeQuery ("SELECT
id_calle_vecina,alto_bajo from "+TABLA_VECINDADES_CALLES+" WHERE
(id_calle="+id_calle+") AND (alto_bajo=1)");
            while (res.next())
            {
                id_calle_vecina =
res.getInt("id_calle_vecina");
                tes =stmt2.executeQuery ("SELECT
num_farolas FROM "+ TABLA_CALLES + " WHERE id=" + id_calle_vecina);
                ses = stmt3.executeQuery
("SELECT mac FROM " + TABLA_FAROLAS + " WHERE id_calle="+id_calle_vecina+" ORDER
BY mac");
                while (tes.next())
                {
                    aux_farolas_calle =
tes.getInt("num_farolas");
                    switch (aux_farolas_calle)
//Número de farolas en la calle vecina
                    {
                        case 1: //Si en la calle
                        case 2:
                            while (ses.next())
                            {
                                aux_macs[contador_vecinos]=ses.getString("mac");
                                contador_vecinos++;
                            }
                            break;
                            default:
                                les =
stmt4.executeQuery ("SELECT mac FROM " + TABLA_FAROLAS + " WHERE
id_calle="+id_calle_vecina+" AND (id_farola="+aux_farolas_calle-1)+" OR
id_farola="+aux_farolas_calle+") ORDER BY mac");
                                while (les.next())
                                {
                                    aux_macs[contador_vecinos]=les.getString("mac");
                                    contador_vecinos++;
                                }
                                break;
                            }
                    }
                }
            }
        }
    }
```



```
        }
        les = stmt4.executeQuery ("SELECT
mac FROM " + TABLA_FAROLAS + " WHERE id_calle="+id_calle+" AND
(id_farola="+(id_farola-1)+" OR id_farola="+(id_farola-2)+" OR
id_farola="+(id_farola+1)+" ) ORDER BY mac");
        while (les.next())
        {
aux_macs[contador_vecinos]=les.getString("mac");
contador_vecinos++;
        }
    }
    else
    {
        les = stmt4.executeQuery
("SELECT mac FROM " + TABLA_FAROLAS + " WHERE id_calle="+id_calle+" AND
(id_farola="+(id_farola-1)+" OR id_farola="+(id_farola-2)+" OR
id_farola="+(id_farola+1)+" OR id_farola="+(id_farola+2)+" ) ORDER BY mac");
        while (les.next())
        {
aux_macs[contador_vecinos]=les.getString("mac");
contador_vecinos++;
        }
    }
}
break;
}
break;
```

5.4.3. Operativa de las SunSPOT

En este apartado, se va a proceder a explicar la operativa de las SunSPOTs, una vez ya se han inicializado siguiendo el proceso descrito en los apartados anteriores.

Una vez la SunSPOT conoce todos los vecinos con los cuales va a comunicarse, **la SunSPOT permanecerá en reposo esperando:**

- Que una **SunSPOT vecina le avise que ha detectado presencia**, y que por lo tanto, en caso de ser necesario, modifique la intensidad con la que está trabajando la farola.



- Que **la propia SunSPOT detecte movimiento**, y sea ella la que tenga que avisar al resto de vecinos.

Además de avisar a los vecinos, cuando se produce alguno de estos dos eventos, la SunSPOT enviará a la Basestation la información necesaria con datos como la potencia a la que está trabajando, tiempo transcurrido desde la última medición, etc.

A continuación se va a detallar todo este proceso.

5.4.3.1. Detección de presencia

Tal y como se ha comentado anteriormente, la SunSPOT se encontrará esperando a que, o bien se reciba un aviso de una SunSPOT vecina, o se detecte movimiento en el rango de acción del sensor. En este sub-apartado se presenta el proceso que se sigue en la SunSPOT.

Para ello, se expone a continuación un diagrama explicativo:

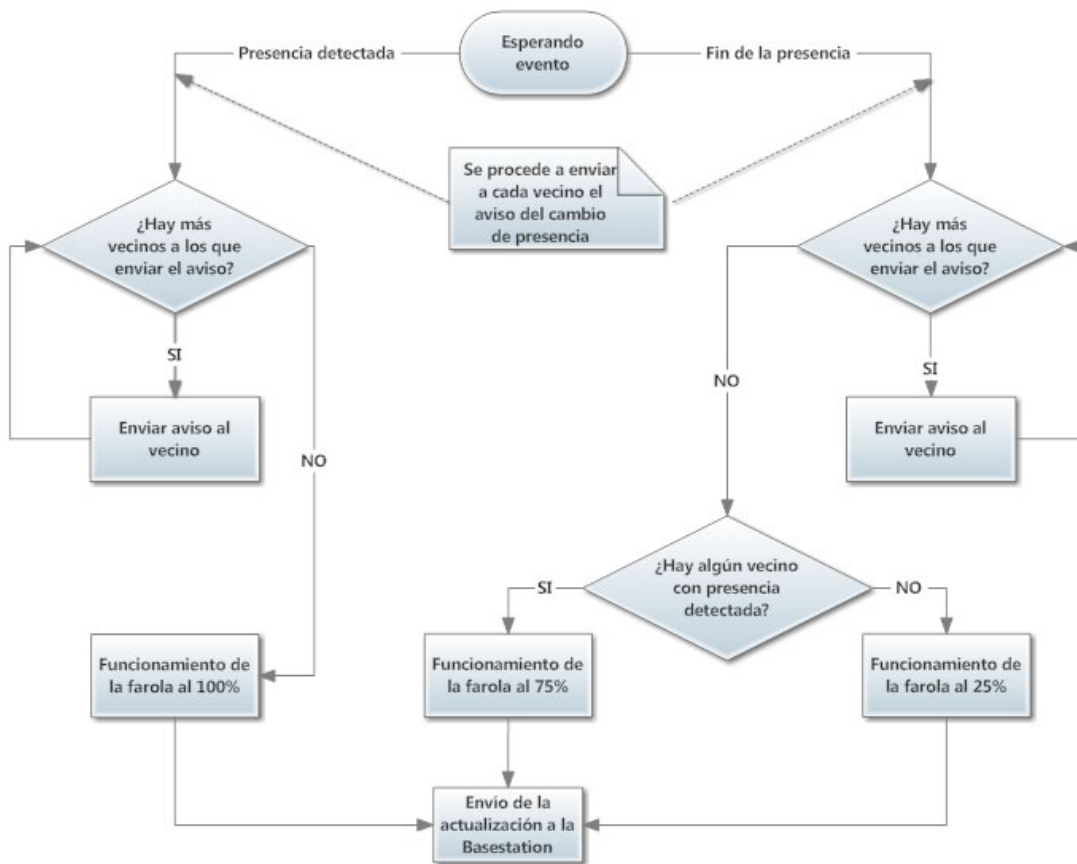


Figura 71. Diagrama del proceso operativo del nodo

- Cuando se detecte una nueva presencia, se procederá a enviar a cada uno de los vecinos un mensaje, avisando de dicho evento. El formato del mensaje será el ya visto en la Figura 51.
- Una vez enviado el aviso a todos los vecinos, la farola comenzará a funcionar al 100% (simulado en este proyecto con la iluminación de todos los LED), y se procederá a enviar una actualización del estado del funcionamiento a la Basestation.
- En caso de que el evento sea que el sensor ha dejado de detectar presencia, el mensaje que se enviará a cada uno de los vecinos será el mostrado en la Figura 50.
- Una vez enviado el aviso a todos los vecinos, la SunSPOT consultará el registro actualizado del número de vecinos de los cuales ha recibido aviso de detección de presencia. El objetivo



de esta consulta es que el funcionamiento sea del 75% si existe algún vecino en el cual actualmente exista presencia, o por el contrario, del 25% si no se da este hecho en ninguno de ellos.

El código correspondiente a esta parte es el siguiente:

```
for (int i=0; i<num_vecinos;i++)
{
    a = null;
    da = null;
    a = (RadiogramConnection) Connector.open
("radiogram://" + listaVecinos[i] + ":100");
    a.setTimeout(1500);
    da = (Radiogram)a.newDatagram(20);

    if (presencia==0) // Presencia detectada
    {
        try
        {
            da.reset();
            da.writeInt(1);
            a.send(da);
            a.close();
        }
        catch(IOException ioe)
        {
            System.out.println("Error envio a "+
listaVecinos[i]);
            a.close();
            da.reset();
            a = null;
            da = null;
        }
    }
    else //Fin de la detección de presencia
    {
        try
        {
            da.reset();
            da.writeInt(0);
            a.send(da);
            a.close();
        }
        catch(IOException ioe)
        {
```




```
listaVecinos[i]);
    System.out.println("Error envio2 a "+
        a.close();
        da.reset();
        a = null;
        da = null;
    }
}
} //Determinar el % de funcionamiento según el evento
detectado
if (presencia==0)
{
    presencia=1;
    if (lum_alcanzada==1)
    {
        enciendeLEDS(8,VERDE);
    }
    else
    {
        enciendeLEDS(8,AZUL);
    }
}
else
{
    presencia=0;
    if (lum_alcanzada==1)
    {
        if (vecinos_ON==0)
        {
            enciendeLEDS(2,VERDE);
        }
        else
        {
            enciendeLEDS(6,VERDE);
        }
    }
    else
    {
        enciendeLEDS(8,AZUL);
    }
}
}
```

5.4.3.2. Envío de actualizaciones a la Basestation

Tal y como se describía en el apartado anterior, cuando se detecte un evento, se enviará la actualización con el estado del funcionamiento de la farola a la Basestation. A continuación se va a exponer el proceso que se sigue por un lado en la SunSPOT, y por otro en la Basestation.

5.4.3.2.1. Proceso seguido en el nodo

La información que se enviará será la siguiente:

datagram				
int	string	int	int	long
0	Fecha y hora	% func. anterior	% func. actual	tiempo última medida

Figura 72. Formato del Datagram enviado a la Basestation

- **Tipo petición:** entero que valdrá 0 para que la Basestation reconozca que se trata de un envío de datos.
- **Timestamp:** String con la fecha y hora en formato YYYY-MM-DD HH:MM:SS
- **% funcionamiento anterior:** Porcentaje al cual se encontraba trabajando la farola antes del cambio que se anuncia en este mensaje.
- **% funcionamiento actual:** Porcentaje al que pasa a funcionar la farola, y el cual será determinado por el algoritmo explicado en el apartado anterior.
- **Tiempo desde la última medida:** Tiempo que ha pasado, en ms, desde la medida anterior.



El código que se ejecuta en la clase Mota es el siguiente:

```
rCon = (DatagramConnection) Connector.open(
    "radiogram://broadcast:" + DATA_SINK_PORT);
dg = rCon.newDatagram(rCon.getMaximumLength());
dg.reset();
now = System.currentTimeMillis();
dif = now-start;
start = now;
dg.writeInt(0);
cal.setTime(new Date(now));
ts = cal.get(Calendar.YEAR) + "-" +
    (1 + cal.get(Calendar.MONTH)) + "-" +
    cal.get(Calendar.DAY_OF_MONTH) + " " +
    cal.get(Calendar.HOUR_OF_DAY) + ":" +
    cal.get(Calendar.MINUTE) + ":" +
    cal.get(Calendar.SECOND);
dg.writeUTF(ts);
if (presencia==1)
{
funcionamiento anterior    dg.writeInt(porcentaje_funcionamiento);    //%
    if (lum_alcanzada==1)
    {
        porcentaje_funcionamiento=100;
    }
    else
    {
        porcentaje_funcionamiento=0;
    }
funcionamiento nuevo    dg.writeInt(porcentaje_funcionamiento);    //%
    dg.writeLong(dif);
}
else
{
    if (vecinos_ON==0)
    {
funcionamiento anterior    dg.writeInt(porcentaje_funcionamiento);    //%
        if (lum_alcanzada==1)
        {
            porcentaje_funcionamiento=25;
        }
        else
        {
```



```
                porcentaje_funcionamiento=0;
            }
            dg.writeInt(porcentaje_funcionamiento);    //%
funcionamiento nuevo
                dg.writeLong(dif);
            }
            else
            {
funcionamiento anterior
                dg.writeInt(porcentaje_funcionamiento);    //%
                if (lum_alcanzada==1)
                {
                    porcentaje_funcionamiento=75;
                }
                else
                {
                    porcentaje_funcionamiento=0;
                }
funcionamiento nuevo
                dg.writeInt(porcentaje_funcionamiento);    //%
                dg.writeLong(dif);
            }
        }
        rCon.send(dg);
        rCon.close();
```

5.4.3.2.2. Proceso seguido en la Basestation

El mensaje que se recibirá seguirá el formato expuesto en el punto anterior. La Basestation, al reconocer que se trata de un envío de datos por parte de la SunSPOT, comenzará a leer, uno por uno, cada uno de los campos que componen el datagram, para después formar la query que ejecutará sobre la Base de Datos y así actualizar la información con la actualización recibida desde la SunSPOT.

El código correspondiente a esta parte en la Basestation es el siguiente:

```
while (true) {
    try {
        rCon.receive(dg);
        tipo_pet = dg.readInt();
```



```
        if (tipo_pet == 1) //solicitud vecinos
    {
        .
        .
        .
    }
    else //tipo_pet= 0 --> Envio de datos
        {
            mac = dg.getAddress();
            horafecha = dg.readUTF();
            medida_anterior = dg.readInt();
            porcentaje = dg.readInt();
            dif_ult_medida = dg.readLong();
            query_insercion.executeUpdate("INSERT INTO " +
TABLA_MEDIDAS +
"(id_farola,timestamp,medida,medida_anterior,tiempo)" +
" VALUES(\'" + mac + "\",\'\' + horafecha + "\",\" +
+ porcentaje + "\",\"+medida_anterior+\", \"+dif_ult_medida+\")");
        }
```

5.4.3.3. Recepción de avisos de presencia

En este apartado se explica el proceso a seguir cuándo se recibe un aviso, de una SunSPOT vecina, de la presencia o no en su rango de detección.

La SunSPOT recibirá un datagram, con formato similar al expuesto en la Figura 50. Al recibirlo, realizará un análisis basado en lo siguiente:

- Comprobará si actualmente se está detectando alguna presencia, ya que en ese caso, la intensidad con la que trabaje la farola seguirá siendo del 100%.
- En caso de no haber presencia en ese momento, actualizará el contador de vecinos que actualmente están detectando presencia (restará 1 en caso de que el aviso sea que ha dejado de detectar presencia; sumará 1 en caso de que el aviso sea



que ha detectado una presencia) para decidir así el porcentaje al que trabajará la farola:

- Si hay algún vecino en el que actualmente se esté detectando presencia, el porcentaje de trabajo de la farola será del 75%.
- Si no existe ningún vecino en el que actualmente se esté detectando presencia, el porcentaje de la farola será del 25%.
- Tras recibir el mensaje, y actualizar su estado, se procederá a enviar la actualización a la Basestation para ser almacenada.

A continuación se muestra el esquema del proceso:

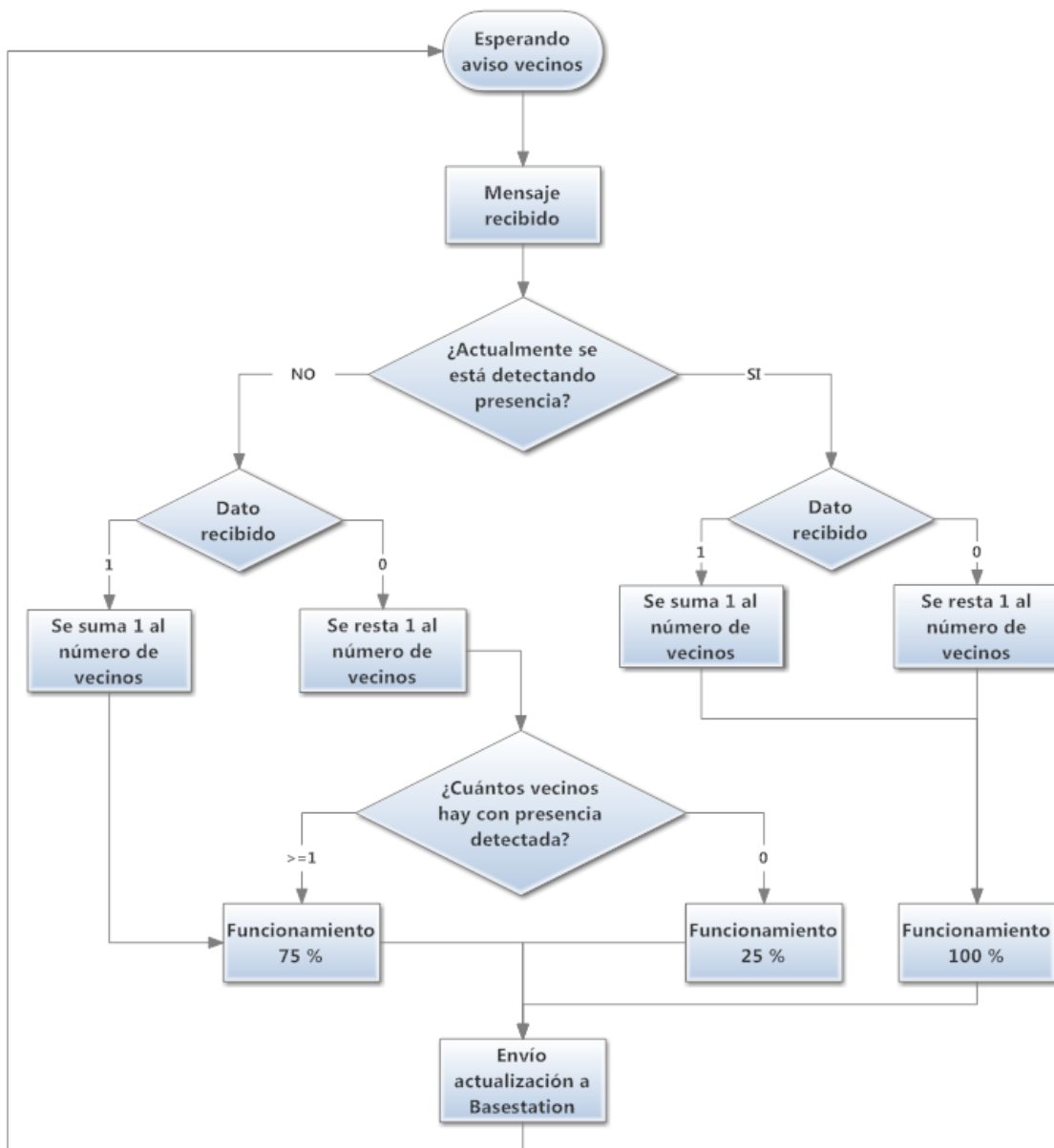


Figura 73. Diagrama del proceso de recepción de avisos de los nodos vecinos

El código ejecutado en esta fase es el siguiente:

```
public void iniciaroperativa()
{
    while (true)
    {
        try
        {
            try
```



```
{  
  
    radiogramConnectionVecinos = (RadiogramConnection)  
Connector.open("radiogram://:100");  
    radiogramConnectionVecinos.setTimeout(-1);  
    datagramVecinos =  
radiogramConnectionVecinos.newDatagram(radiogramConnectionVecinos.getMaximumLength());  
    datagramVecinos.reset();  
    radiogramConnectionVecinos.receive(datagramVecinos);  
    aux = datagramVecinos.readInt();  
    analizar(aux,0);  
    System.out.println("Dato Recibido de "+  
datagramVecinos.getAddress() + ": " + aux);  
    radiogramConnectionVecinos.close();  
    datagramVecinos.reset();  
  
    //Envio a la base del estado.  
  
    rCon = (DatagramConnection) Connector.open(  
        "radiogram://broadcast:" + DATA_SINK_PORT);  
    dg = rCon.newDatagram(rCon.getMaximumLength());  
    dg.reset();  
    now = System.currentTimeMillis();  
    dif = now-start;  
    start = now;  
    dg.writeInt(0);  
    cal.setTime(new Date(now));  
    ts = cal.get(Calendar.YEAR) + "-" +  
        (1 + cal.get(Calendar.MONTH)) + "-" +  
        cal.get(Calendar.DAY_OF_MONTH) + " " +  
        cal.get(Calendar.HOUR_OF_DAY) + ":" +  
        cal.get(Calendar.MINUTE) + ":" +  
        cal.get(Calendar.SECOND);  
    dg.writeUTF(ts);  
  
    if (presencia==1)  
    {  
        dg.writeInt(porcentaje_funcionamiento);  
        if (lum_alcanzada==1)  
        {  
            porcentaje_funcionamiento=100;  
        }  
    }  
    else
```




```
        {
            porcentaje_funcionamiento=0;
        }
        dg.writeInt(porcentaje_funcionamiento); ///  
funcionamiento nuevo
        dg.writeLong(dif);
    }
    else
    {
        if (vecinos_ON==0)
        {
            dg.writeInt(porcentaje_funcionamiento);
            if (lum_alcanzada==1)
            {
                porcentaje_funcionamiento=25;
            }
            else
            {
                porcentaje_funcionamiento=0;
            }
            dg.writeInt(porcentaje_funcionamiento); ///  
funcionamiento nuevo
            dg.writeLong(dif);
        }
        else
        {
            dg.writeInt(porcentaje_funcionamiento);
            if (lum_alcanzada==1)
            {
                porcentaje_funcionamiento=75;
            }
            else
            {
                porcentaje_funcionamiento=0;
            }
            dg.writeInt(porcentaje_funcionamiento);
            ///  
funcionamiento nuevo
            dg.writeLong(dif);
        }
    }
    rCon.send(dg);
    rCon.close();
    if (lum_alcanzada==0)
    {
        enciendeLEDS(8,AZUL);
    }
}
```



```
    }
    System.out.println("Enviada info");
    }

    catch (com.sun.spot.peripheral.TimeoutException te)
    {
        System.out.println("Tiempo agotado");
    }
}

catch (IOException ex)
{
    ex.printStackTrace();
}
}

public void analizar (int pres, int modo)
{
    if (pres==0)
    {
        if (modo==0)
        {
            vecinos_ON--;
        }
        else
        {
        }
    }
    if (presencia==1)
    {
        enciendeLEDS(8,VERDE); //100%
    }
    else
    {
        if (vecinos_ON==0)
        {
            enciendeLEDS(2,VERDE); //20%
        }
        else
        {
            enciendeLEDS(6,VERDE); //75%
        }
    }
}
}
else
```



```
{
    if (modo==0)
    {
        vecinos_ON++;
    }
    else
    {
    }
    if (presencia==1)
    {
        enciendeLEDS(8,VERDE); //100%
    }
    else
    {
        enciendeLEDS(6,VERDE); //75%
    }
}
}
```

5.4.4. Herramienta de monitorización

Para mostrar toda la información acerca del Sistema, se ha creado una página web en PHP, que permite conocer:

- % de funcionamiento actual de las farolas.
- Situación geográfica de cada farola.
- **Evolución del consumo eléctrico**, así como filtrado por fechas.

Gracias a esta herramienta, se podrá tener acceso a la situación actual del sistema para poder llevar un control de todo lo que concierne a su funcionamiento.

Debido a la utilización de la información facilitada por el Excelentísimo Ayuntamiento de Pedro Muñoz, el diseño ha sido planificado como un servicio ofrecido en su página web. Por ello, y para la utilización legal del escudo, se solicitó la correspondiente autorización al mismo, pudiéndose ser consultada en la sección 8. Anexos.

La ficticia página web quedaría de esta forma estructurada en 2 partes:

- **Página principal** donde se puede observar un resumen de la situación actual de las farolas y localización geográfica.
- **Página específica de cada farola**, con información detallada acerca de:
 - Consumo
 - Funcionamiento actual
 - Localización
 - Filtrados por fechas.

A continuación se puede observar la página web principal y la específica de cada farola:

CALLE	FAROLA	ULTIMO CAMBIO	% FUNCIONAMIENTO
1	1	2013-08-13 04:19:59	100 %
1	2	2013-08-13 04:19:59	75 %
1	3	2013-08-13 04:35:24	75 %
1	4	2013-08-12 14:08:45	75 %
2	1	2013-08-12 11:10:08	25 %
2	2	2013-08-12 12:04:29	25 %

Figura 74. Captura de la página web principal

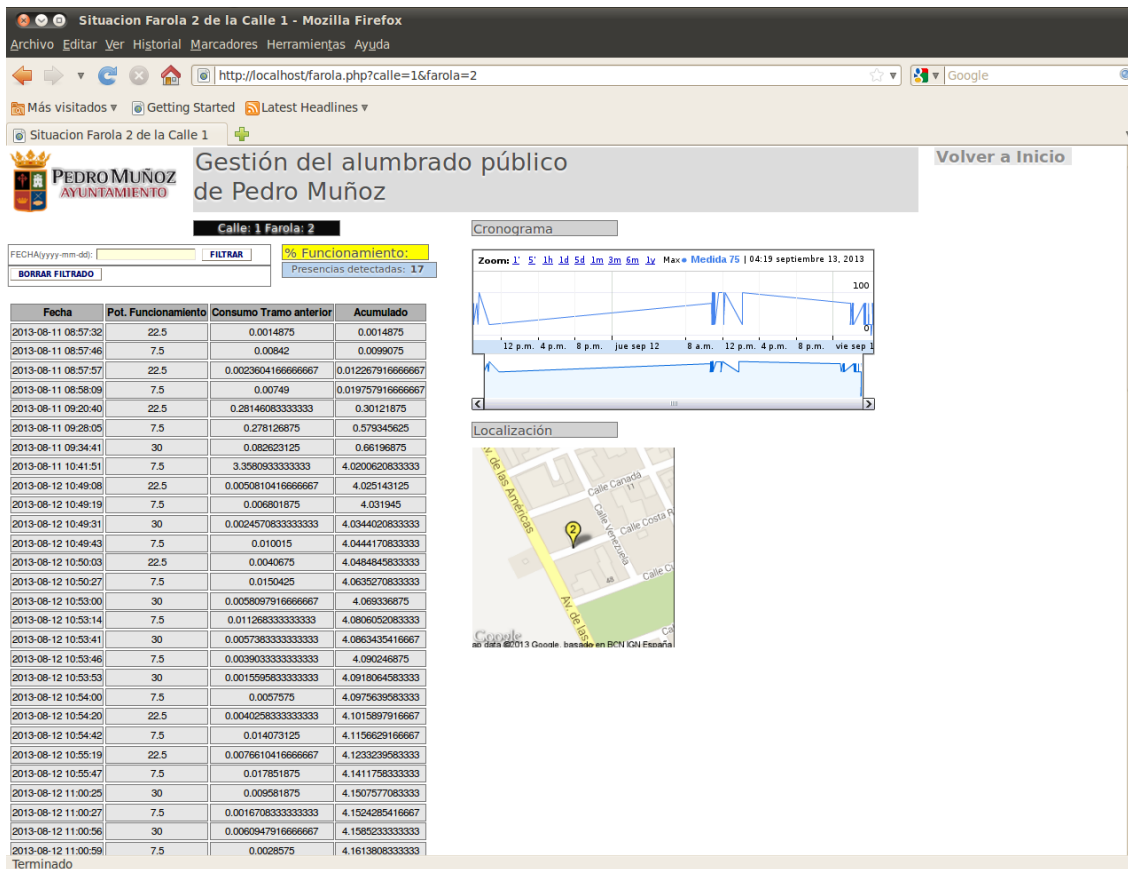


Figura 75. Captura de la página web específica de cada farola

En dicha herramienta de gestión se irían implementando nuevas funciones. De algunas de ellas se habla en el apartado 6. Conclusiones y desarrollos futuros.

5.5. Pruebas realizadas

En este apartado se expondrán algunas de las pruebas realizadas para mostrar el correcto funcionamiento del sistema.

Más concretamente, se ofrecen pruebas de los siguientes procesos:

- Puesta en marcha de la SunSPOT.
- Algoritmo descubrimiento de vecinos.
- Influencia del sensor de Iluminación.
- Envío y Recepción a/de las SunSPOTs vecinas tras detectar una modificación de la presencia.

- Actualización de la herramienta de gestión con el funcionamiento actual de la farola.

A continuación se mostrará, a través de mensajes mostrados, tanto en la consola de la SunSPOT, como en la consola de la basestation, el proceso de puesta en marcha de la SunSPOT. Dicho proceso se encuentra procedimentado en el punto “5.4.2. Puesta en marcha”

Además, los resultados de las pruebas se apoyarán también en diferentes capturas realizadas de las herramientas de simulación.

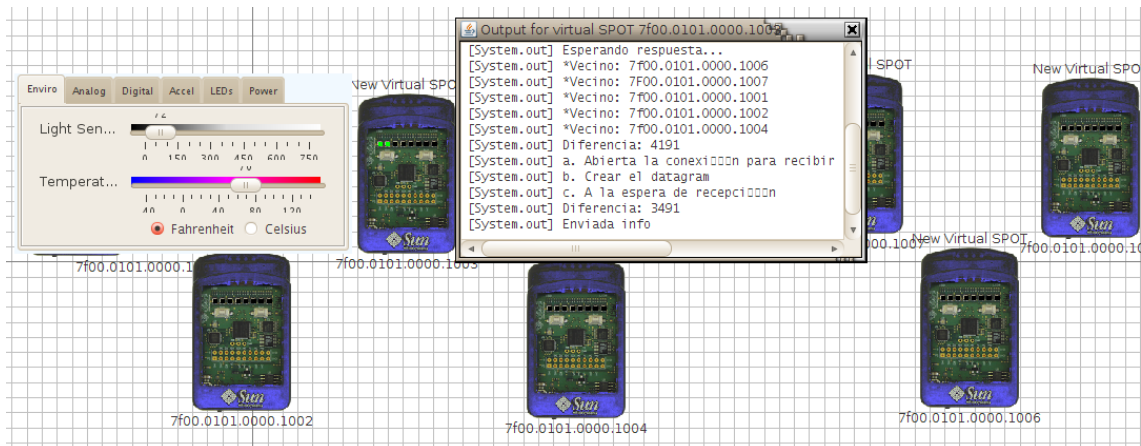
El escenario será el descrito en el apartado “5.3.2. Escenario”, y más concretamente, en la Figura 48. A continuación podemos ver el despliegue con SunSPOTs virtuales realizado en Solarium:



Figura 76. Despliegue de Virtual SPOTs en Solarium

A continuación se exponen las pruebas realizadas:

- **Prueba 1:** Se inicializará la SunSPOT con MAC “7f00.0101.0000.1003”, con el sensor de iluminación por debajo del valor mínimo, y por tanto, haciendo que la farola se ponga a funcionar a un 25%. Tras ello, enviará la primera la medida a la base:



```
Archivo Editar Ver Terminal Ayuda
[java] MAC Emisora: 7F00.0101.0000.1003
[java] [radiogram] Adding: Output to 7F00.0101.0000.1003 on port 99
[java] [radiogram] Adding: Input from 7F00.0101.0000.1003 on port 99
[java] **Número de vecinos encontrados5
[java] ->VECINO: 7f00.0101.0000.1006
[java] ->VECINO: 7f00.0101.0000.1007
[java] ->VECINO: 7f00.0101.0000.1001
[java] ->VECINO: 7f00.0101.0000.1002
[java] ->VECINO: 7f00.0101.0000.1004
[java] [radiogram]Removing: Input from 7F00.0101.0000.1003 on port 99
[java] [radiogram]Removing: Output to 7F00.0101.0000.1003 on port 99
```

```
Archivo Editar Ver Terminal Ayuda
[java] Medida recibida:
[java] MAC: 7F00.0101.0000.1003
[java] Fecha y hora: 2013-8-21 7:26:33
[java] % Funcionamiento 25
[java] Consumo último tramo 3491
```

Figura 77. Resultados de la Prueba 1

- **Prueba 2:** Se subirá el valor del sensor de iluminación por encima del máximo para funcionar, observando como la farola deja de funcionar (representado por los LEDs en color azul), y su posterior envío de información a la Basestation:

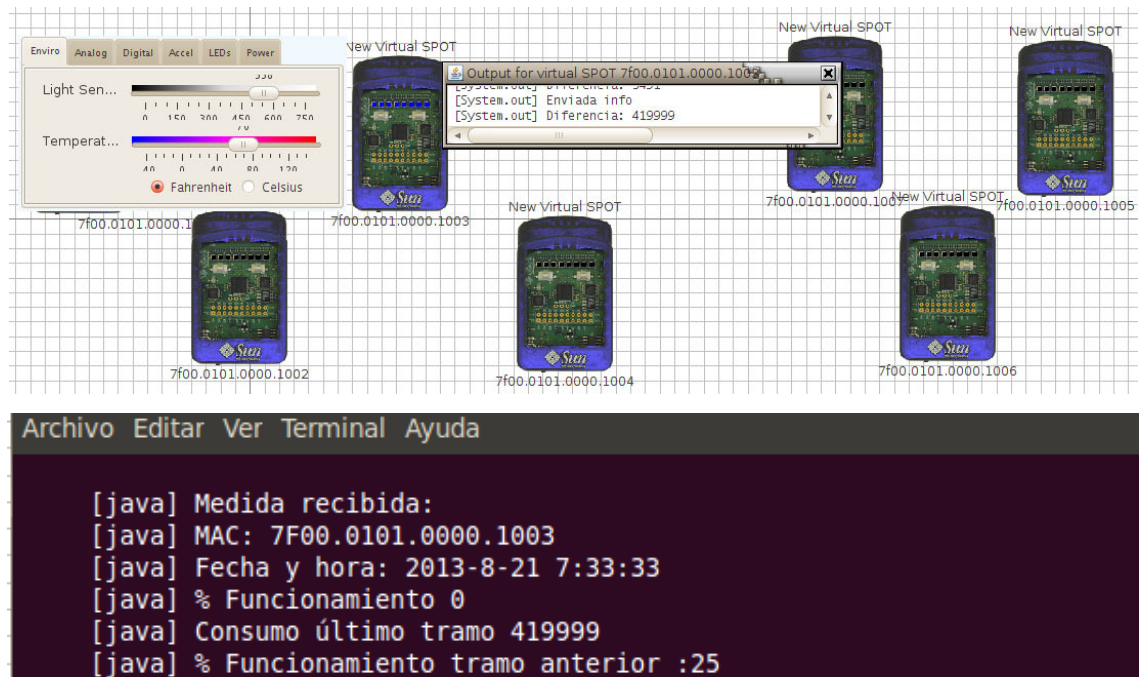


Figura 78. Resultados de la Prueba 2

- **Prueba 3.** En este caso se detectará presencia en la SunSPOT "7f00.0101.0000.1003", pasando a operar al 100%, avisando ésta a sus vecinos (5 en este caso), que pasarán al 75%:

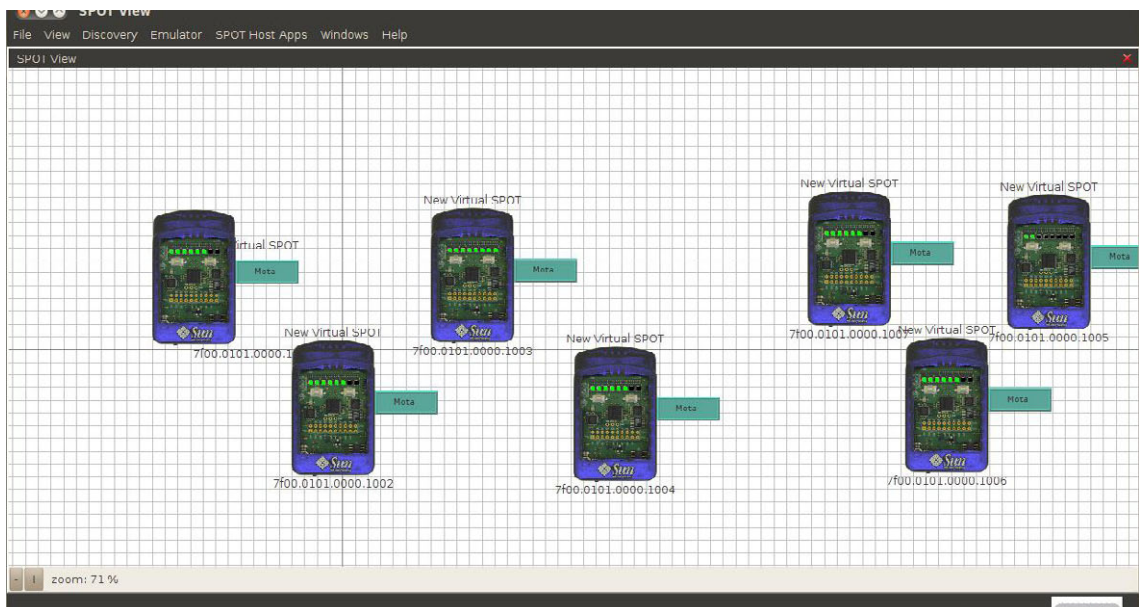


Figura 79. Resultados de la Prueba 3

- **Prueba 4:** Se muestra una combinación de estados de las farolas, para mostrar su reflejo en el sistema de gestión:

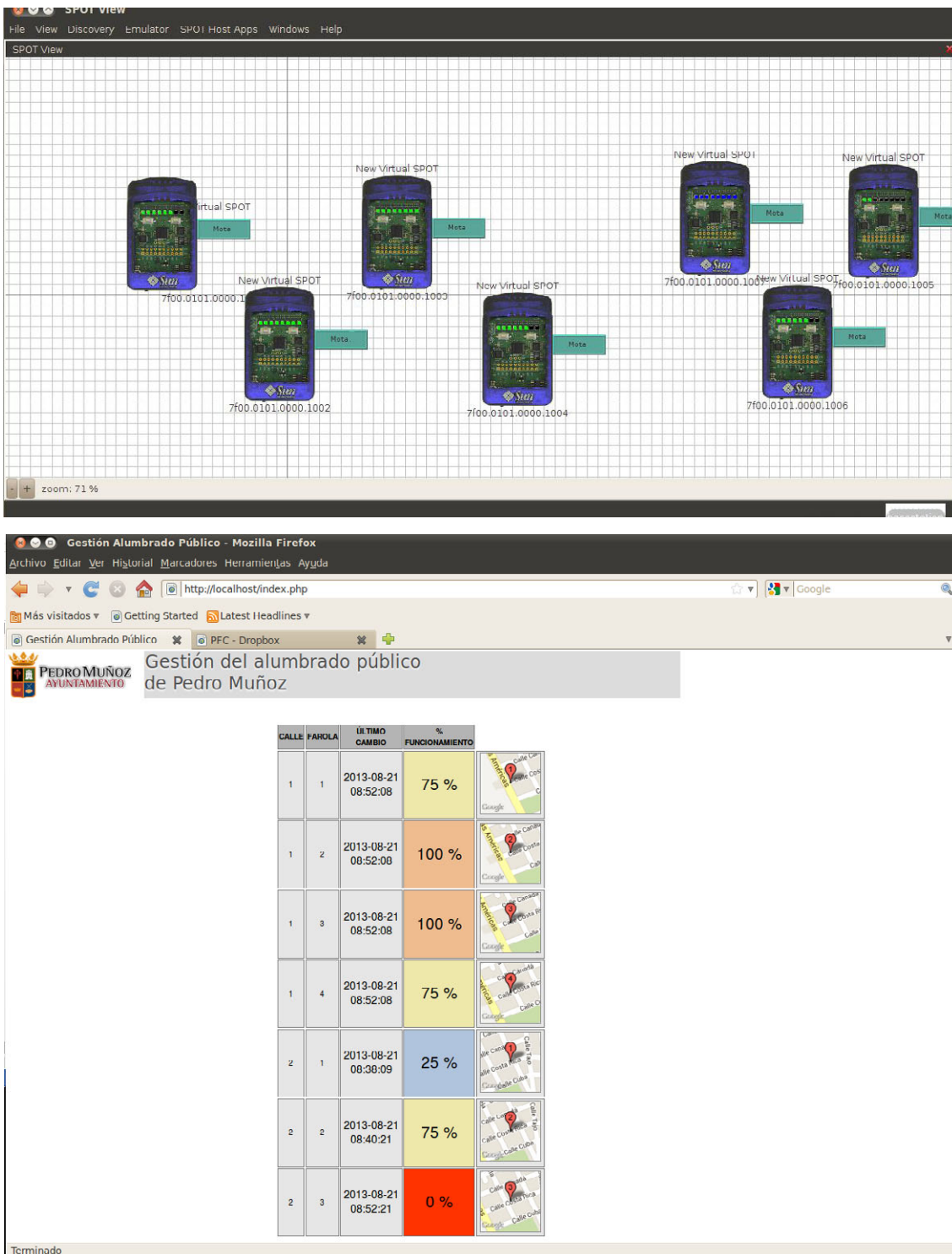


Figura 80. Resultados de la prueba 4

- **Prueba 5:** Se observará cómo, una farola que se encuentra apagada, aun recibiendo notificaciones de vecindades de las farolas contiguas, no verá reflejada modificación alguna en su funcionamiento:

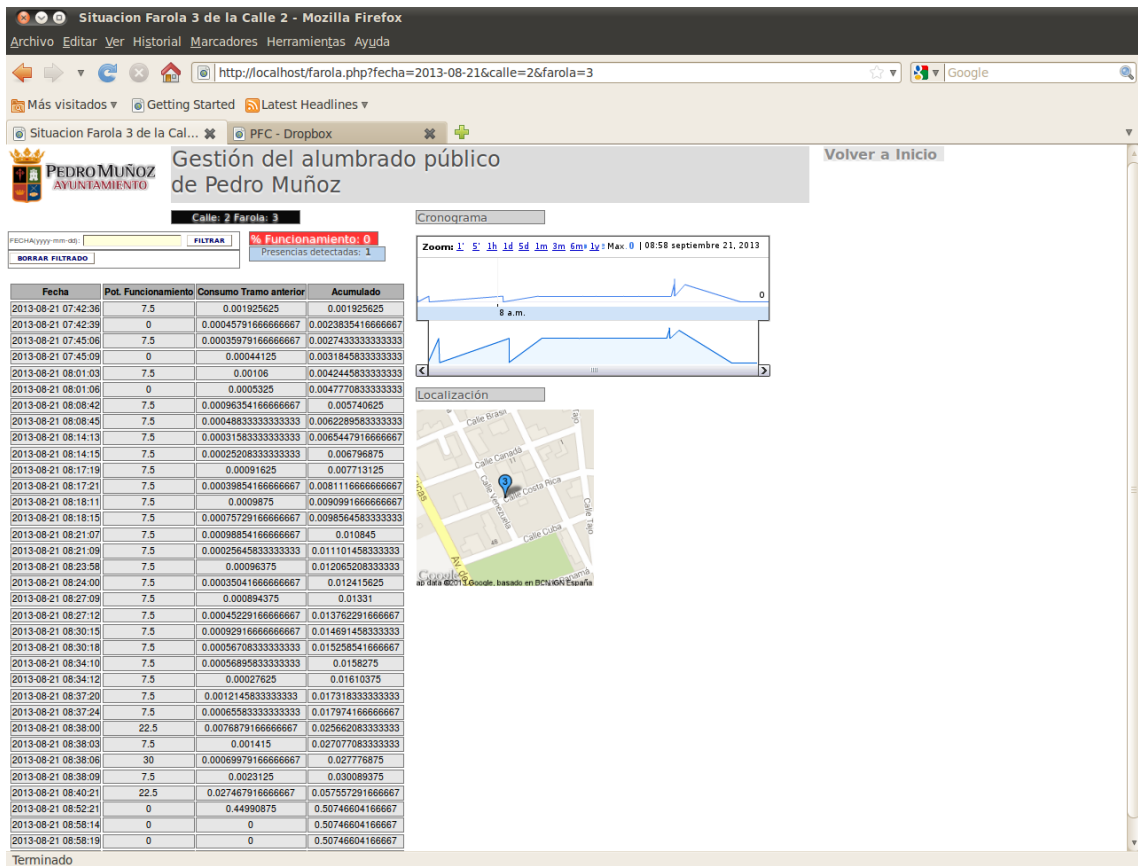


Figura 81. Resultados de la prueba 5



Jorge Gálvez Alfaro

Desarrollo de sistema de gestión de alumbrado público mediante WSN



POLITÉCNICA



6

Conclusiones y desarrollos futuros



Jorge Gálvez Alfaro

Desarrollo de sistema de gestión de alumbrado público mediante WSN



POLITÉCNICA



Con la realización de este Proyecto Fin de Carrera, he buscado aunar gran parte de los conocimientos y destrezas adquiridos durante el transcurso de la Ingeniería Técnica de tal forma que quedaran plasmados en el desarrollo de este proyecto.

El objetivo primordial era la búsqueda de un sistema que ofreciera ventajas reales y que tuviera posibilidades de ser llevado a producción. Esta fue la razón por la cual se eligió desarrollar el sistema de alumbrado público inteligente explicado en este documento.

Debido a los beneficios que la implementación de este sistema podría tener en cualquier localidad, mantuve diversas reuniones con el Alcalde del Excelentísimo Ayuntamiento de Pedro Muñoz para presentarle dicha propuesta, la cual en buena parte se pudo realizar gracias a la documentación facilitada por el consistorio. Pese a que la predisposición de los representantes del equipo de gobierno de la localidad siempre fue muy positiva, con intención clara de querer desplegar una fase piloto, la situación económica actual, impidió la implementación real del proyecto, objetivo marcado desde el comienzo.

No obstante, se ha dejado la puerta abierta a una futura implantación del sistema debido a que gracias al prototipo realizado, con las herramientas de Sun, se ha podido demostrar la utilidad y ventajas del sistema. Un despliegue de lo especificado en este proyecto supondría para la localidad un gran avance tecnológico, consiguiendo un gran ahorro tanto energético como económico y del cual se han dado cuentas en diversos apartados de esta memoria.

La tecnología WSN empleada no se plantea como solución única para resolver el problema de control de iluminación planteado dado que puede integrarse con soluciones cableadas (Power Line Communication) o WiFi. En general dependerá de los costes de instalación tanto en el apartado de costes de los componentes como de la mano de obra



necesaria para su ejecución, que en general dependerá del tipo de instalación a automatizar.

Otro de los grandes beneficios aportados por una WSN como la desarrollada en este proyecto es la escalabilidad, ya que, por un lado, aumentar el número de puntos de la red resulta una tarea sencilla. Por otro lado, la adición de funcionalidades extras es igualmente viable, porque si bien la red expuesta en este PFC se centra únicamente en el control de la iluminación, en un futuro podría extenderse para soportar otro tipo de funcionalidades, las cuales podrían desplegarse haciendo uso de los sensores ya instalados en las farolas, y/o añadiendo nuevos sensores ubicados fuera de la infraestructura de iluminación, pero estando todos ellos interrelacionados.

6.1 Trabajos futuros

Este Proyecto Fin de Carrera se deja abierto a posibles mejoras en el futuro, ya que poseer una WSN, y por tanto, nodos en cada una de las farolas, permite implementar multitud de posibilidades:

- Obtención de valores ambientales más detallados.
- Poder detectar el vandalismo.
- Poseer puntos de información actualizados para, por ejemplo, descargar guías de la localidad o las últimas noticias.
- Ofrecer una página web con más detalles y un diseño más actual.

En definitiva, el mundo de las WSN está en constante desarrollo, y una buena utilización de ellas, puede conllevar a hacer de la vida diario algo mejor.



7

Referencias Bibliográficas



Jorge Gálvez Alfaro

Desarrollo de sistema de gestión de alumbrado público mediante WSN



POLITÉCNICA



Referencias Bibliográficas

- [1] I. W. S. C. M. Sudip Misra, *Guide to Wireless Sensor Networks*, London: Springer, 2009.
- [2] L. G. Muñoz, «Estudio de Consumo en Redes de Sensores INalámbricos para la detección de ondas características en ECG,» 2008.
- [3] «Nike,» [En línea]. Available: http://store.nike.com/es/es_es/?sitesrc=eslp&l=shop,search,searchList-sensor#l=shop,pdp,ctr-inline/cid-300/pid-323819.
- [4] «Adidas,» [En línea]. Available: <http://www.adidas.com/es/miCoach/#GearLanding/sdf/modgr>.
- [5] D. C. J. P. R. S. a. J. A. A. Mainwaring, «Wireless sensor networks for habitat monitoring,» de *WSNA'02*, Atlanta, USA, 2002.
- [6] «STMicroelectronics,» [En línea]. Available: <http://www.st.com/internet/analog/product/250938.jsp>.
- [7] «Monnit,» [En línea]. Available: <http://www.monnit.com/products/wireless-sensors/humidity-HA.php>.
- [8] «sensors,» [En línea]. Available: <http://www.sensormag.com/product/wireless-temperature-transmitter-oleumtech>.
- [9] «Monnit,» [En línea]. Available: <http://www.monnit.com/products/wireless-sensors/lux.php>.
- [10] «Tadd,» [En línea]. Available: http://tandd.com/about_tandd/newsrelease/20100906001.html.
- [11] «libelium,» [En línea]. Available: <http://www.libelium.com/products/waspmote/sensors>.
- [12] Q. L. Y. Z. Jiaming Zhang, «A Tire Pressure Monitoring System Based onWireless Sensor Networks Technology,» de *International Conference on MultiMedia and Information Technology*, 2008.
- [13] «Cátedra COITT,» [En línea]. Available: http://catedra-coitt.euitt.upm.es/web_HD/inicio.htm.
- [14] «watermetercorp,» [En línea]. Available: <http://www.watermetercorp.com/medinalambrico.html>.
- [15] «SmartGrids,» [En línea]. Available: <http://www.smartgrids.eu>.
- [16] N.-b. Z. Z.-x. F. J.-h. X. W.-w. G. Y. S. Qing-quan Li, «A proposed Intelligent Vehicle Telematics Service System (IVTS),» 2006.



- [17] «ERCIM news,» [En línea]. Available: <http://ercim-news.ercim.eu/coopers-automotive-visions-beyond-in-car-driver-assistance>.
- [18] J.-W. Lee, Y.-B. Yoo, J.-J. Rho y S.-S. Choi, «An enhanced parking lot service model using wireless sensor network,» 2008.
- [19] «StreetLine,» [En línea]. Available: <http://www.streetlinenetworks.com/smart-parking-solutions/how-our-sensor-technology-works>.
- [20] «ISAACS & Associates,» [En línea]. Available: <http://www.isaacstech.com/SoilMoisture.htm>.
- [21] D. S. a. D. G. Chunguo Jing, «Design of Streetlight Monitoring and Control System Based on Wireless Sensor Networks,» de *Industrial Electronics and Applications, 2007. ICIEA 2007. 2nd IEEE Conference on* , 2007.
- [22] «Philips,» [En línea]. Available: http://www.simplicityevent.philips.com/global/tomorrow/light_blossom/.
- [23] A. W. Holger Karl, *Protocols and Architectures for Wireless Sensor Networks*, Wiley, 2007.
- [24] «Coalesenses,» [En línea]. Available: <http://www.coalesenses.com/index.php?page=isense-hardware>.
- [25] «Sayme,» [En línea]. Available: <http://www.sayme.es/productos/sensbee-3/>.
- [26] «Arduino,» [En línea]. Available: <http://www.arduino.cc>.
- [27] «Arduino UNO,» [En línea]. Available: <http://arduino.cc/en/Main/ArduinoBoardUno>.
- [28] «Arduino FIO,» [En línea]. Available: <http://arduino.cc/en/Main/ArduinoBoardFio>.
- [29] «Arduino Xbee Shield,» [En línea]. Available: <http://arduino.cc/es/Main/ArduinoXbeeShield>.
- [30] «Arduino Ethernet Shield,» [En línea]. Available: <http://arduino.cc/es/Main/ArduinoEthernetShield>.
- [31] «Oracle Labs,» [En línea]. Available: <http://www.sunspotworld.com/index.html>.
- [32] «Libellium,» [En línea]. Available: <http://www.libellium.com/company/about>.
- [33] «Sensinode,» [En línea]. Available: <http://www.sensinode.com/EN/company.html>.
- [34] «ZigBee Alliance,» [En línea]. Available: <http://www.zigbee.org>.
- [35] Z. Alliance, *ZigBee Specification*, January 17, 2008 11:09 am.
- [36] IEEE, «IEEE Std 802.15.4-2003,» 2003.



- [37] «<http://sx-de-tx.wikispaces.com/ZIGBEE>,» [En línea]. [Último acceso: 22 08 2013].
- [38] Bluetooth, «Bluetooth,» [En línea]. Available: <https://www.bluetooth.org/Building/HowTechnologyWorks/CoreSpecifications.htm>. [Último acceso: 28 septiembre 2011].
- [39] N. B. A.-G. W. H. S. Tilak, «Collaborative storage management in sensor networks,» 2005.
- [40] H. O. Y. W. M. M. L.-S. P. a. D. R. Philo Juang, «Energy-Efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences with ZebraNet,» San Jose, CA, 2002.
- [41] Excmo. Ayto. de Pedro Muñoz, «Auditoría para la mejora de la eficiencia y ahorro energético en el alumbrado público exterior, en Pedro Muñoz (C. Real)».
- [42] Oracle, «<http://www.sunspotworld.com/docs/Yellow/SunSPOT-Programmers-Manual.pdf>,» [En línea].
- [43] Wikipedia, «http://en.wikipedia.org/wiki/Squawk_virtual_machine,» [En línea].
- [44] Oracle, «<http://docs.oracle.com/javase/tutorial/jdbc/overview/index.html>,» [En línea].
- [45] MySQL, «<http://www.mysql.com/about>,» [En línea].
- [46] Apache, «http://httpd.apache.org/ABOUT_APACHE.html,» [En línea].
- [47] Google, «<https://developers.google.com/>,» [En línea].
- [48] Sun SPOT World, «<http://www.sunspotworld.com/docs/Yellow/SunSPOT-Programmers-Manual.pdf> - Pág. 58,» [En línea].
- [49] Wikipedia, «http://es.wikipedia.org/wiki/Luz_diurna,» [En línea].
- [50] «Shimmer,» [En línea]. Available: <http://www.shimmer-research.com/p/sensor-and-modules>.
- [51] D. Gislason, ZigBee Wireless Networking, Newnes.



8

Anexos



Jorge Gálvez Alfaro

M^a del Prado Peinado Marchante, Secretaria Acctal del Excmo. Ayuntamiento de Pedro Muñoz, le hace saber, le participa del Acuerdo adoptado, entre otros, por la Junta de Gobierno Local en sesión extraordinaria celebrada el día 14 de agosto de 2013, a expensas de la aprobación del acta:

1. INSTANCIAS

«*** Visto la petición de trece de agosto de 2013 (r.e. 8406) solicitada por D. Jorge Gálvez Alfaro sobre utilización del escudo del Ayuntamiento de Pedro Muñoz dentro del proyecto fin de carrera que está elaborando.

Se trataría de una simulación de página web y mostraría la gestión del funcionamiento y consumo de farolas.

Visto el artículo 188 del Real Decreto 2568/1986 de 28 de noviembre por el que se aprueba el Reglamento de Organización, Funcionamiento y Régimen Jurídico de las Entidades Locales que dispone que «cada Corporación Local, sin perjuicio de poder usar en las comunicaciones oficiales el sello constituido por los emblemas del escudo nacional, usará el que privativamente corresponda a la Entidad Local respectiva, ya porque estuviera consagrado por la Historia y el uso, ya en virtud de expresa rehabilitación o adopción a tenor de los artículos anteriores»

Considerando que según consta en la solicitud presentada por el Sr. Gálvez la página web donde se va a mostrar el escudo no va a ser pública sino de carácter interno

La Junta de Gobierno, con el voto favorable de unanimidad de los miembros presentes (4de 5),
ACUERDA:

PRIMERO: Autorizar la utilización del escudo del Ayuntamiento de Pedro Muñoz a D. Jorge Gálvez Alfaro en la página web elaborada para su proyecto fin de carrera.

SEGUNDO: Dar traslado del presente acuerdo al interesado, haciéndole saber que esta autorización no supone un uso indiscriminado del escudo municipal, sino que únicamente la autorización comprende al titular y para el acto autorizado. »

Contra la presente Resolución, que pone fin a la vía administrativa, puede interponer alternativamente o recurso de reposición potestativo, en el plazo de un mes a contar desde el día siguiente a la recepción de esta notificación, ante el Alcalde de este Ayuntamiento de Pedro Muñoz, de conformidad con los artículos 116 y 117 de Ley de Régimen Jurídico y Procedimiento Administrativo Común, o recurso contencioso-administrativo, ante el Juzgado de lo Contencioso-Administrativo con sede en Ciudad Real, en el plazo de dos meses a contar desde el día siguiente a la recepción de la presente notificación, de conformidad con el artículo 46 de la Ley de la Jurisdicción Contencioso-Administrativa. Si se optara por interponer el recurso de reposición potestativo, no podrá interponer recurso contencioso-administrativo hasta que aquel sea resuelto expresamente o se haya producido su desestimación por silencio. Todo ello sin perjuicio de que pueda ejercitar cualquier otro recurso que estime pertinente.

En Pedro Muñoz, a 16 de agosto de 2013

La Secretaria Acctal



Fdo.: M^a. Del Prado Peinado Marchante.

Área de Secretaría

Plaza de España, 1 – 13620 Pedro Muñoz (Ciudad Real)

Tfno.: 926 586 001 – FAX: 926 586 798 – email: secretaria@pedro-munoz.com – www.pedro-munoz.es.