“*Smart keys*” como aplicación ilustrativa del IoT

[[1]](#footnote-1)Pérez Díaz, Isabel Victoria*, Universidad Distrital Francisco José de Caldas*.

***Resumen*—En el siguiente artículo se explicará de forma detallada los conceptos básicos de las redes de comunicaciones y el IoT con el fin de desarrollar el diseño lógico del montaje de unas llaves inteligentes, implementando sensores de movimiento, alarmas, routers, servidores, tarjetas programables de Arduino y switches. Para su elaboración se tomó como base distintos artículos publicados por la IEEE, los cuales serán explicados más adelante. Todo con el fín de ver una sencilla aplicación del internet de las cosas, teniendo en cuenta la distancia que se encuentran entre los dispositivos (aproximadamente 8m), el alcance máximo del sensor (7m) y el mejor y peor tiempo de respuesta obtenido tras varias mediciones (5ms y 1610ms respectivamente) se llega a la conclusión a nivel cuantitativo que la velocidad es de 1 m/s, si bien se llega a la conclusión que esta velocidad no es óptima, se ve la forma de optimizarla mediante la aplicación de metodologías como AboT y Hamma las cuales con la aplicación de nodos logran un alcance de 15m de distancia por sensor con un radio de detección respectivo a 7m y una velocidad de respuesta de 10 m/s.**

***Abstract*— The following article will explain in detail the basic concepts of communication networks and the IoT in order to develop the logical design of the assembly of intelligent keys, implementing motion sensors, alarms, routers, servers, Arduino programmable cards and switches. For its elaboration, several articles published by the IEEE were taken as a basis, which will be explained later.** **All in order to see a simple application of the internet of things, reaching an average speed of 1 m / s taking into account the distance between the devices (approximately 8m), the maximum range of the sensor (7 meters) and the best and worst response time (5ms and 1610ms respectively) although it is concluded that this speed is not optimal, we can see how to optimize it by applying methodologies such as AboT and Hamma, which with the application of nodes achieve a range of 15m distance per sensor with a respective detection radius of 7m and a response speed of 10 m / s.**

***Palabras clave*— IoT, router, switch, servidor, Arduino, redes de comunicaciones, sensor.**

# **INTRODUCCIÓN**

Actualmente la sociedad avanza a gran velocidad, todo se convierte en una constante carrera, y es común olvidar distintos implementos que son fundamentales para la vida cotidiana, entre ellos se encuentran las llaves del hogar, trayendo consigo inconvenientes posteriores.

Es por esto que ha surgido la idea de crear una alarma que se active cuando se hayan olvidados las llaves al momento de estar en el portón de la casa; para que esto se pueda llevar a cabo, deberá existir una comunicación entre la puerta de la casa, y las llaves, esta se realizará mediante el uso de sensores, los cuales serán definidos a lo largo del diseño, para posteriormente enviar este resultado de comunicación a la alarma.

Para la realización de este trabajo se tomó como base los siguientes textos: HAMA: A Three-Layered Architecture for Integrating Object Tracking and Location Management in Wireless Sensor Networks[1], del autor Chao-Chun Chen; Tracking irregularly moving objects based on alert-enabling sensor model in sensor networks[2], del autor Chung y An object tracking technique in Wireless sensor network based on prediction[3] del autor Yang Hai-bo, entre otros, los cuales serán explicados detalladamente más adelante.

# **ARTÍCULOS RELACIONADOS**

A continuación, se presentan artículos relacionados con el proyecto de los cuales se pude extraer distintos elementos para aplicarlos posteriormente en la red.

## **HAMA: A Three-Layered Architecture for Integrating Object Tracking and Location Management in Wireless Sensor Networks.**

Se realiza la aplicación de la estructura HAMA para minimizar el número de nodos que en las actividades de seguimiento y administrar de manera eficiente toda la información de ubicación de objetos en movimiento. [1]

En la simulación, un campo de detección se construye en un tamaño de 1600 × 1600m2, y la red de sensores se divide en cuadrículas con un tamaño de 30 × 30m2, y selecciona aleatoriamente un nodo sensor como líder de celda para cada cuadrícula. Colocamos la OET en la esquina superior izquierda del archivo de detección para administrar las ubicaciones de todos los objetos. Observe que la OTS colocada en el centro del campo de detección puede salvar la mayoría de las comunicaciones entre OTS y TAM. Por lo tanto, nuestra configuración en el experimento es rigurosa para cada esquema, por lo que podemos observar claramente la diferencia entre varios esquemas. En Sensing Alert Model, el radio de detección de cada nodo sensor es de 20 metros, y el radio del círculo de alerta es de 8 metros. Cuando ocurre un evento M, cada nodo activo detecta una vez por segundo para rastrear el objeto. En el Modelo de seguimiento de alertas, cada celda se establece en 200 × 200m2, y el ancho de la celda M se establece en 40 metros. Para la red de sensores inalámbricos, se adopta el codicioso protocolo de enrutamiento sin estado perimetral (GPSR) [7] en la entrega de consultas e información de objetos de informes. En nuestra simulación, la velocidad del objeto se establece aleatoriamente entre [0], [401] km / h.

El objetivo de la arquitectura de modelo de alerta jerárquica propuesta es minimizar el consumo de energía de los nodos sensores para el seguimiento y la gestión de objetos en movimiento. Comparamos HAMA con otros dos métodos relacionados similares, EASE [10] y AbOT [3], en los experimentos. Las métricas de rendimiento en nuestros experimentos incluyen el número total de nodos de sensores, el número total de mensajes transmitidos y el número total de transferencia de células.

Si se implementara este artículo en el proyecto se lograría un mayor alcance en la detección de movimiento en el sensor a menor distancia (maximización de velocidad).

## **Tracking irregularly moving objects based on alert-enabling sensor model in sensor networks.**

En la simulación, construimos una red de cuadrícula que incluye nodos de sensores lógicos 900 en 300 × 300 metros2. La distancia entre dos vecinos verticales (u horizontales) es de diez metros. Para cada dispositivo sensor, se generan dos lecturas por segundo, y el rango de detección es de 15 metros. El radio de la zona C se establece en 7,5 metros. En el caso de que el objeto se mueva de la zona C a la zona M (es decir, se haya producido el evento M), se utiliza la Política de cobertura del rango para determinar qué sensores están activados. La energía de cada fase en un sensor está dentro del rango razonable. Los modos de consumo de energía, incluidos el modo "Reposo", el modo "Procesamiento", el modo "Procesamiento y recuperación", el modo "Detección" y el modo "Transmisión" se configuran en 5 μA, 8 mA, 8 mA, 4 mA y 25 mA, respectivamente. Dado que nuestros experimentos se centran en estudiar el rendimiento de los métodos de seguimiento de objetos, algunas técnicas subyacentes adoptan los algoritmos existentes. La transmisión de datos entre los sensores al servidor OT sigue el algoritmo de enrutamiento de múltiples saltos de ruta más corta [2]. Cada trayectoria de un objeto en movimiento se genera siguiendo el trabajo de Sey-dims. Finalmente, la velocidad de un objeto en movimiento es de diez metros por segundo.

Esta investigación de ser aplicada en el proyecto logra gestionar un enrutamiento dinámico (de mínimo camino) entre los elementos IoT para así maximizar la velocidad de comunicación teniendo una mayor distancia entre cada uno de los elementos y sensores.

## **An object tracking techinque in Wireless sensor network based on prediction**

Dependiendo de la topología y la aplicación del sensor, varios algoritmos de predicción diferentes para rastrear objetos móviles han sido reportados en la literatura. Obviamente, diferentes algoritmos de predicción dan como resultado una precisión de seguimiento diferente. Por lo tanto, la relación entre la probabilidad perdida y el intervalo de seguimiento varía con los algoritmos de predicción. Además, las probabilidades que faltan de un algoritmo de predicción específico con los mismos intervalos de seguimiento en un objeto móvil con un patrón de movimiento diferente también son diferentes. Sobre la base de las simulaciones, observamos que la relación entre la probabilidad perdida y el intervalo de seguimiento, en el tramo correspondiente, puede ajustarse en una función cuadrática P (s) = as2 + bs + c, donde s es el intervalo de seguimiento, a, "b yc son las constantes, ya que la probabilidad, P (s), es positiva para cualquier s (> 0), la parte constante c debe ser mayor que cero. En este documento, para la conveniencia de la descripción, creamos un modelo de predicción simple, como MEDIO Heurístico, como ejemplo, en el que la ubicación y la velocidad actuales del objeto móvil se derivan del promedio del historial de movimiento del objeto. La Fig. 1 muestra el resultado de una simulación, en la cual una persona que camina, cuya velocidad promedio es de 1 milla / hora rastreada por una red de sensores con 100 nodos de sensores bajo el algoritmo de predicción MEDIA Heurística. Los valores de los intervalos de seguimiento superiores a 10 segundos se filtran fuera del ajuste de la curva, ya que están más allá de la tolerancia precisa y deben ignorarse. A partir de la adaptación de la curva en la Fig. 1, la función aproximada de probabilidad perdida se puede escribir como P (s) = 0.0013s2 + 0.025s + 0.062 en función de la adaptación cuadrática que se muestra con la línea continua. [3]

Esto puede ser implementado en los sensores, ya que no sería necesario mantenerlo activo todo el tiempo, debido a que se puede predecir en qué momento se activaría según el comportamiento que ha venido llevando.

## **A scheme for low path loss in object monitoring with sensor network**

Se realiza la aplicación de un nuevo esquema de energía que transmite eficientemente la información del objetivo al servidor con la red del sensor. [4]

En esta sección evaluamos el rendimiento y la eficacia del esquema propuesto mediante simulación por computadora. En la simulación construimos una red de red que consta de 900 nodos de sensores en un área de 300 × 300m2. La distancia entre dos vecinos verticales (u horizontales) es, por lo tanto, de 10 m. Para cada nodo sensor, el radio de detección es de 15 m. El objeto se mueve aleatoriamente con una velocidad, v = 10 m / s. El programa de simulación es un simulador dirigido por eventos desarrollado utilizando el lenguaje C ++. Localizamos al azar un objetivo y luego recopilamos datos sobre la pérdida de la ruta. 10,000 objetivos son probados y los datos son promediados.

En el proyecto se puede incorporar al sensor de movimiento para el ahorro de energía y que, aun así, sea eficiente la comunicación con el servidor.

## **Alert, monitoring and tracking for electronic device prototype.**

Se elabora un prototipo para monitorear un dispositivo electrónico y recibir una alerta cuando el dispositivo se aleja de una distancia establecida. [5]

Mediante una mejora a las llaves se puede utilizar este proyecto, para que ya no sea necesario obturar un botón, sino que sea mediante uso de geolocalización la alerta.

## **Power conservation and quality of surveillance in target tracking sensor networks.**

El seguimiento de objetivos es una aplicación importante de las redes de sensores inalámbricos. En esta aplicación, los nodos supervisan y rastrean colectivamente el movimiento de un evento u objeto objetivo. Las operaciones de red tienen dos estados: el estado de vigilancia durante la ausencia de cualquier evento de interés, y el estado de seguimiento que es en respuesta a cualquier objetivo en movimiento. Por lo tanto, las operaciones de ahorro de energía, que son de importancia crítica para extender la vida útil de la red, también deberían estar operativas en dos modos diferentes. En este documento, estudiamos las operaciones de ahorro de energía en ambos estados de las operaciones de red. Durante el estado de vigilancia, se propone un conjunto de métricas novedosas para la calidad de la vigilancia específicamente para detectar objetos en movimiento. En el estado de seguimiento, proponemos un esquema de mensajería colaborativa que despierta y cierra los nodos del sensor con precisión espacial y temporal. Este estudio, que es una combinación de análisis teóricos y evaluaciones simuladas, cuantifica la relación entre la conservación de energía y la calidad de la vigilancia, al tiempo que presenta las pautas para la implementación eficiente de nodos de sensores para la aplicación de seguimiento de objetivos. [6]

Puede ser aplicado en el proyecto reduciendo el gasto de energía en los distintos sensores inalámbricos que se poseen en la red, además si se desea realizar posteriormente el seguimiento de las llaves mediante uso de sensores de rastreo, esto podrá reducir el costo de su implementación.

## **Deploying multiple interconnected gateways in heterogeneous wireless sensor networks: An optimization approach.**

Los datos recolectados por los sensores a menudo tienen que ser entregados remotamente a través de rutas inalámbricas de múltiples saltos a los receptores de datos conectados a los servidores de aplicaciones para el procesamiento de la información. La posición de estos sumideros tiene un gran impacto en la calidad de la Red de sensores inalámbricos (WSN) específica. De hecho, puede crear cuellos de botella artificiales que afectan la eficiencia energética y la vida útil de WSN. Este documento considera un escenario de red heterogéneo donde los sensores inalámbricos entregan datos a pasarelas intermedias adaptadas a una tecnología inalámbrica diversa e interconectadas entre sí y al receptor. Se desarrolló un marco de optimización basado en Integer Linear Programming (ILP) para localizar gateways inalámbricos que minimizan el costo total de instalación y el consumo de energía en la WSN, mientras se toma en cuenta la cobertura de múltiples saltos entre sensores y gateways, y la conectividad entre gateways inalámbricos. También se considera un escenario de variable de tráfico, donde la red puede pasar por puntos de operación de tráfico alto y bajo, y la topología se optimiza en consecuencia. Las formulaciones de ILP propuestas se resuelven para optimizar las instancias de tamaño mediano para analizar la calidad de las redes diseñadas, y también se proponen algoritmos heurísticos para abordar escenarios heterogéneos a gran escala. [7]

Para la optimización de la comunicación entre los distintos dispositivos inalámbricos, se utilizaría este proyecto, ya que con estas mejoras se realizaría una comunicación en tiempo real mejorando también el uso de energía.

## **A multi-objective approach for data collection in wireless sensor networks**

Las redes de sensores inalámbricos (WSN) se implementan para recopilar grandes cantidades de datos del entorno. Esta información producida debe ser entregada a través de la interfaz inalámbrica del sensor utilizando comunicaciones de varios saltos hacia un receptor. La posición del fregadero afecta el rendimiento de la red de sensores inalámbricos con respecto a la demora y el consumo de energía, especialmente para los sensores de retransmisión. La optimización del proceso de recopilación de datos en las redes de sensores inalámbricos multisalto es, por lo tanto, un problema clave. Este artículo aborda el problema de la recopilación de datos utilizando receptores móviles en una WSN. Proporcionamos un marco que estudia el equilibrio entre el consumo de energía y el retraso en la recopilación de datos. Este marco proporciona soluciones que permiten a los responsables de la toma de decisiones diseñar de manera óptima el plan de recopilación de datos en redes de sensores inalámbricos con receptores móviles. [8]

Se implementará en el proyecto recolectando los datos de las veces en las que se encontraron las llaves y las veces en las que no, para así llevar un estudio estadístico con dicha información.

* 1. ***Heap and parameter-based load balanced clustering algorithms for wireless sensor networks***

La agrupación en clústeres es una técnica eficiente para mejorar la escalabilidad y el tiempo de vida de redes de sensores inalámbricos WSN. Sin embargo, la eficiencia energética de los nodos del sensor y el equilibrio de carga de los cabezales del clúster CH son los dos problemas más importantes que deben abordarse al diseñar el algoritmo de agrupamiento para WSN. En este documento, primero presentamos un algoritmo de agrupamiento llamado EELBCA algoritmo de agrupamiento equilibrado de carga eficiente de energía que se ocupa de la eficiencia energética de los nodos del sensor y el equilibrio de carga de los CH en términos de cardinalidad de los clusters. EELBCA es un algoritmo de agrupamiento basado en min-heap que se muestra para ejecutarse en tiempo de log m para n nodos de sensor y m CHs. [9]

Para alargar la vida y escabilidad de la red, se utilizará este proyecto, pudiéndose agregar más sensores y mayores funcionalidades posteriormente.

## **Deploying multiple interconnected gateways in heterogeneous wireless sensor networks: An optimization approach.**

La sobrecarga de comunicación es uno de los grandes problemas en la agrupación distribuida para la cual se deben introducir protocolos que puedan garantizar un flujo limitado de paquetes de control. Teniendo en cuenta que la mayoría de los protocolos basados ​​en fuzzy son vulnerables a la sobrecarga de paquetes de control en la fase de configuración del clúster, se ha introducido un nuevo enfoque para reducir la sobrecarga de comunicación y, en consecuencia, optimizar la formación de clúster distribuido. Los clústeres se forman usando lógica difusa y las rutas óptimas hacia la estación base se establecen en base al algoritmo ACO de optimización de colonias de hormigas. El problema de optimización consiste en encontrar rutas hacia la estación base de manera que la duración del recorrido se minimice y el equilibrio del consumo de energía se mantenga entre las rutas en la ruta. La lógica difusa y las cajas de herramientas de ACO están desarrolladas en C ++ e integradas con la plataforma de simulación OMNeT ++ para implementar el protocolo. Los resultados experimentales demuestran que el protocolo propuesto extiende la vida útil de la red en comparación con los protocolos LEACH y CHEF. [10]

Esto se puede utilizar para facilitar la comunicación de los dispositivos y evitar la sobrecarga para así poder realizar una alerta en tiempo real.

# **METODOLOGÍA**

Para la realización de este montaje es necesario contar con los siguientes elementos:

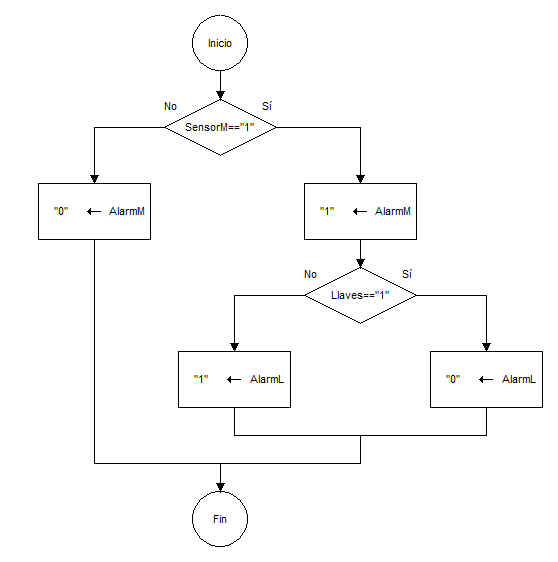
* 1 Switch 2950-24
* 2 Sirenas de MTBF 26280
* 1 Servidor PT
* 1 Sensor de movimiento
* 1 tarjeta programable SBC-PT
* 1 push toggle button

Teniendo estos elementos se puede proceder a realizar el montaje el cual se tiene planeado de la siguiente manera:

1. Conexión al switch de las dos sirenas, el sensor de movimiento, el servidor y la tarjeta programable.
2. Configuración la dirección IP y la máscara del servidor.
3. Configuración del servicio DHCP del servidor para generación automática de direcciones IP.
4. Activación del servicio IoT en el servidor.
5. Configuración de los hosts para asignación de dirección IP mediante DHCP.
6. Configuración de IoT monitor del servidor.
7. Conexión de servidor remoto a cada uno de los dispositivos inteligentes.
8. Creación de condición de activación de sirena de movimiento según el sensor de movimiento, con el monitor IoT.
9. Conexión de sirenas y botón de las llaves a tarjeta programable.
10. Programación de la tarjeta para que se active la alerta de las llaves al prenderse la sirena de movimiento y no encontrarse activo el botón de las llaves.

Básicamente, la lógica del montaje funciona como el diagrama de flujo que se muestra a continuación (fig. 1), suponiendo que:

* SensorM hace referencia al sensor de movimiento.
* AlarmM hace referencia a la alarma de movimiento.
* AlarmL hace referencia a la alarma de las llaves.
* Llaves hace referencia al botón anexo a las llaves.



*Imagen 1. Diagrama de flujo lógica de la red. Fuente: autor*

En el diagrama de flujo mostrado anteriormente se observa que, al tener un sensor de movimiento, si este se activa la alarma de movimiento se activará, si este no se activa, la alarma de movimiento tampoco se activará.

De ser activada la alarma de movimiento se mirará si ha sido o no obturado el botón de las llaves, de ser obturado la alarma de las llaves no se activará, de no haber sido obturado la alarma de las llaves será activada hasta que se obture el botón.

# **DISEÑO**

Para el diseño del montaje se realizó una sencilla simulación en Packet Tracer, la cual será explicada a continuación.

* + - 1. Ubicar el switch en la parte de la puerta.



*Imagen 2. Switch ubicado en la puerta. Fuente: autor*

* + - 1. Ubicar las sirenas y el sensor de movimiento en una zona cercana a la puerta.



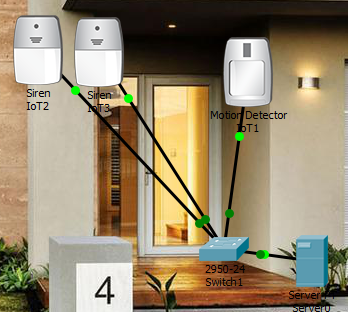
*Imagen 3. Sirenas ubicadas. Fuente: autor.*

* + - 1. Se debe tener un servidor el cual servirá como monitor de los elementos IoT. Este será ubicado a un costado de la casa.



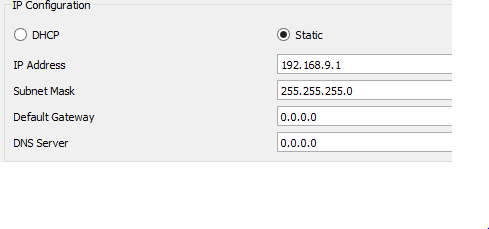
*Imagen 4 Servidor IoT ubicado. Fuente: autor*

* + - 1. Ahora se deben conectar todos los elementos al Switch mediante conexión de cable Copper Straight-Through, esta debe darse mediante puertos FasthEthernet.



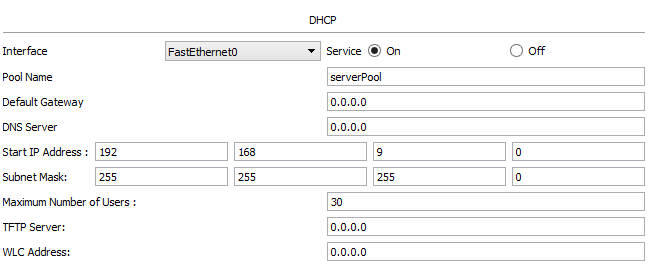
*Imagen 5. Elementos conectados al Switch. Fuente: autor*

* + - 1. Se procederá a configurar la dirección IP del servidor.



*Imagen 6. configuración IP servidor. fuente: autor*

* + - 1. A continuación, se configura el servicio DHCP del servidor para poder generar direcciones IP de forma automática.



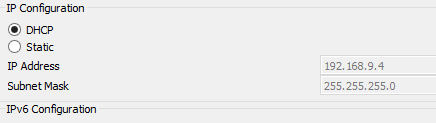
*Imagen 7 Configuración servicio DHCP. Fuente: autor*

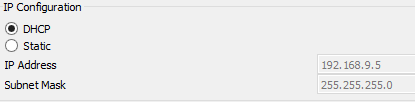
* + - 1. Ahora se debe activar el servico IoT en el servidor.

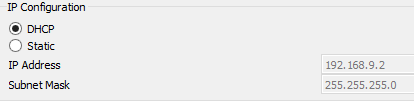


*Imagen 8 Activación servicio IoT. Fuente: autor*

* + - 1. Teniendo esto se procede a configurar la dirección IP de cada uno de los dispositivos inteligentes.

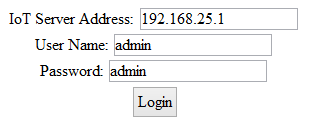






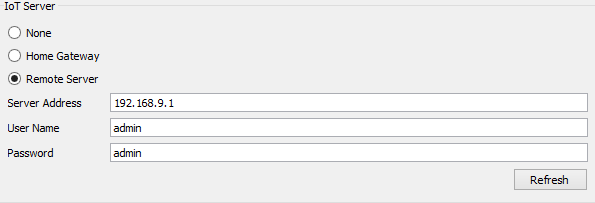
*Imagen 9 Configuración IP dispositivos inteligentes. Fuente: autor*

* + - 1. A continuación, se accede al monitor IoT desde el servidor. Esto se realiza ingresando la dirección IP del servidor y un usuario y contraseña del servidor IoT.



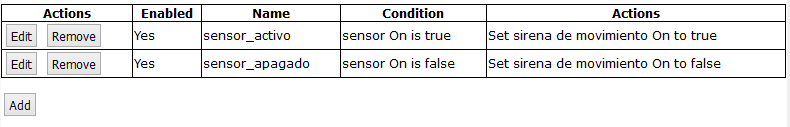
*Imagen 10 Monitor IoT. Fuente: autor*

* + - 1. Ahora se debe configurar el servidor IoT al que se conectan los dispositivos inteligentes como un servidor remoto, ingresando la IP, el usuario y la contraseña.



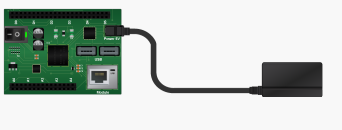
*Imagen 11 Configuración servidor IoT en dispositivos. Fuente: autor*

1. A continuación, se realiza la condición de encender la alarma de movimiento si se activa el sensor de movimiento y que de lo contrario se mantenga apagada. Esto se realiza mediante el monitor IoT del servidor.



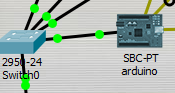
*Imagen 14 Condiciones IoT. Fuente: autor*

1. Ahora se procederá a realizar la conexión de las llaves con el switch, para esto se utiliza la tarjeta programable SBC-PT. Para esto primero se le debe incorporar el componente T-IOT-NM-1CE para tener un puerto Ethernet.



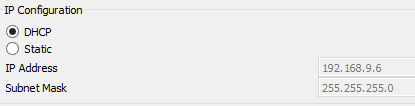
*Imagen 15 Tarjeta programable con puerto ethernet. Fuente: autor*

1. Ya es posible conectar la tarjeta programable con el switch mediante Ethernet.



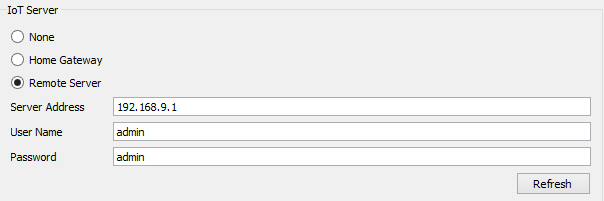
*Imagen 16 Conexión Switch con tarjeta programable. Fuente: autor*

1. Se debe configurar la dirección IP de la tarjeta programable.



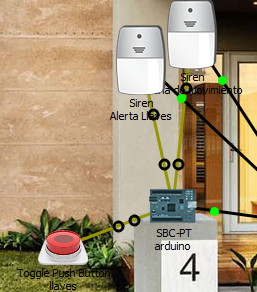
*Imagen 12 Configuración IP tarjeta programable. Fuente: autor*

1. Se debe configurar el servidor IoT al que se conecta la tarjeta programable como un servidor remoto, ingresando la IP, el usuario y la contraseña.



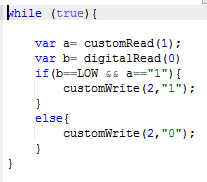
*Imagen 13 Configuración servidor IoT de tarjeta programable. Fuente: autor*

1. Se debe conectar a la tarjeta programable el toggle push button de las llaves y las dos alarmas, teniendo muy presente el puerto al que se conectó cada dispositivo.



*Imagen 17 Conexión alarmas y botón a tarjeta programable. Fuente: autor*

1. A continuación, se procede a programar la tarjeta programable, para que la alarma de las llaves sea activada cuando no sea obturado el botón y se haya activado la alarma de movimiento.



*Imagen 18. Código tarjeta programable. Fuente: autor*

Con estos pasos ya se posee montada la red para el manejo de las llaves inteligentes.

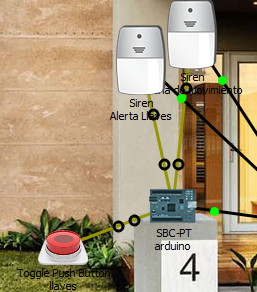
1. **IMPLEMENTACIÓN**

Para la implementación de este montaje, mediante la simulación en Packet Tracer se obtuvo una gran ayuda con la utilización de herramientas IoT.

Sin embargo, se debieron realizar ciertas adaptaciones que no son contempladas en el simulador Packet Tracer, todo esto solucionado mediante el uso de una tarjeta programable SBC-PT.

El problema presentado en la implementación, fue la imposibilidad de conectar directamente el botón incorporado a las llaves con el servidor, por ende, fue necesario conectar este botón con la tarjeta programable, al igual que las alarmas.

A continuación, se muestra dicha conexión:



*Imagen 19 Conexión alarmas y botón a tarjeta programable. Fuente: autor*

Fue necesario implementar un código en JavaScript para la tarjeta programable, este código se muestra a continuación (fuente: autor):

while (true){

var a=customRead(1);

var b=digitalRead(0);

if(b==LOW && a=="1"){

customWrite(2,"1");

}

else{

customWrite(2,"0");

}

}

En la primera línea de código decimos que mientras se encuentre funcionando la red, se ejecutarán las instrucciones, posteriormente se toma lo enviado por la sirena de movimiento (var a) y lo enviado por el botón (var b) y se realiza la condición de que si el botón se encuentra apagado (LOW) y la alarma activa (1), se prenderá la alarma de las llaves (estado 1), de lo contrario se mantendrá apagado.

Con esta implementación se logra tener una conectividad entre el botón de las llaves y el servidor, permitiendo configurar los casos de encendido y apagado de la alarma de las llaves, que de otra forma no habrían podido ser configuradas.

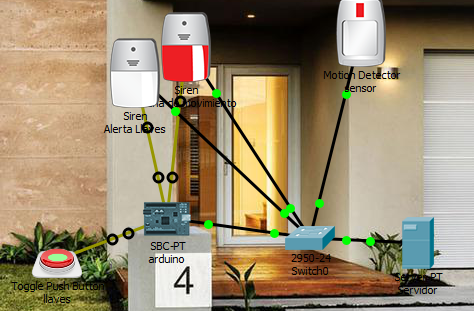
# **VALIDACIÓN DEL DISEÑO**

Ya teniendo la red montada se procede a validar que funcione como se espera para los distintos casos:

## **SE ACTIVA EL SENSOR Y LAS LLAVES SON ACTIVADAS.**

En el caso que el sensor de movimiento sea activado y el botón de las llaves también, debe sólo encender la alarma de movimiento, quedando apagada la alarma de las llaves ya que implica que no han sido olvidadas.

A continuación, se muestra el resultado:

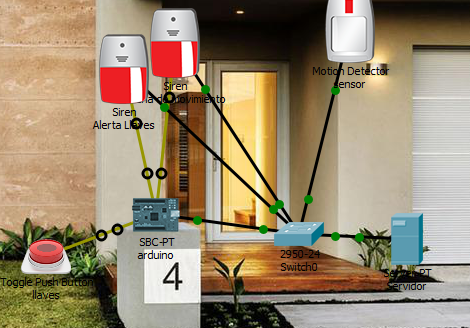


*Imagen 20. Validación con botón activo. Fuente: autor*

## **SE ACTIVA EL SENSOR Y LAS LLAVES NO SON ACTIVADAS.**

En el caso que el sensor de movimiento sea activado y el botón de las llaves no, se debe encender la alarma de movimiento, y posteriormente encender la alarma de las llaves, ya que estas han sido olvidadas.

A continuación, se muestra el resultado:



*Imagen 21. Validación con botón no activo Fuente: autor*

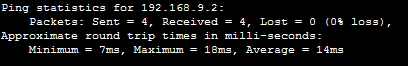
1. **OBTENCIÓN DE RESULTADOS**

Observando el resultado de la implementación, y comprobando que funciona el montaje, se procede a realizar una toma de datos cuantitativos respecto al montaje.

Teniendo en cuenta que toda comunicación se hace vía el servidor, se mide el tiempo de respuesta de cada dispositivo desde el servidor. Viéndose lo siguiente:

* 1. ***TIEMPO SIN ENCONTRARSE ACTIVO EL SENSOR DE MOVIMIENTO***

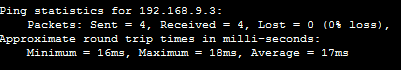
Tiempo de respuesta desde el servidor al sensor de movimiento:



*Imagen 22. ping servidor a sensor de movimiento Fuente: autor*

Se puede observar que la comunicación es exitosa teniendo un tiempo mínimo de respuesta de 7 ms y tiempo máximo de respuesta de 18 ms, siendo en promedio un tiempo de 14 ms, que se encuentra en el rango permitido.

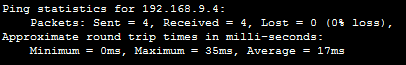
Tiempo de respuesta desde el servidor a la alarma de movimiento:



*Imagen 23. ping servidor a alarma de movimiento Fuente: autor*

Se obtiene como resultado una comunicación exitosa, teniendo un tiempo mínimo de respuesta de 16 ms y tiempo máximo de respuesta de 18 ms, siendo en promedio un tiempo de 17 ms, que se encuentra en el rango permitido.

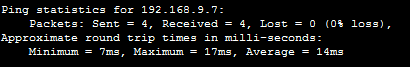
Tiempo de respuesta desde el servidor a la alarma de las llaves:



*Imagen 24. ping servidor a alarma de las llaves Fuente: autor*

Se realiza una comunicación exitosa, teniendo un tiempo mínimo de respuesta de 0 ms y tiempo máximo de respuesta de 35 ms, siendo en promedio un tiempo de 17 ms, que se encuentra en el rango permitido.

Tiempo de respuesta desde el servidor a la tarjeta programable:



*Imagen 25. ping servidor a tarjeta programable Fuente: autor*

Se comunican de forma exitosa ambos dispositivos, teniendo un tiempo mínimo de respuesta de 7 ms y tiempo máximo de respuesta de 17 ms, siendo en promedio un tiempo de 14 ms, que se encuentra en el rango permitido.

* 1. ***TIEMPO ENCONTRANDOSE ACTIVO EL SENSOR DE MOVIMENTO Y SI EL BOTÓN***

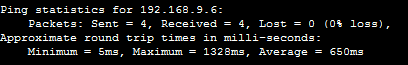
Tiempo de respuesta desde el servidor al sensor de movimiento:



*Imagen 26. ping servidor a sensor de movimiento Fuente: autor*

Se da una comunicación exitosa, teniendo un tiempo mínimo de respuesta de 0 ms y tiempo máximo de respuesta de 365 ms, siendo en promedio un tiempo de 91 ms, que se encuentra en el rango permitido.

Tiempo de respuesta desde el servidor a la alarma de movimiento:



*Imagen 27. ping servidor a alarma de movimiento Fuente: autor*

Se presenta con éxito la comunicación, teniendo un tiempo mínimo de respuesta de 5 ms y tiempo máximo de respuesta de 1328 ms, siendo en promedio un tiempo de 650 ms, que se encuentra en el rango permitido.

Tiempo de respuesta desde el servidor a la alarma de las llaves:



*Imagen 28. ping servidor a alarma de las llaves Fuente: autor*

Se observa una comunicación exitosa, teniendo un tiempo mínimo de respuesta de 0 ms y tiempo máximo de respuesta de 1424 ms, siendo en promedio un tiempo de 365 ms, que se encuentra en el rango permitido.

Tiempo de respuesta desde el servidor a la tarjeta programable:

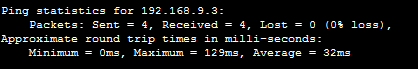


*Imagen 29. ping servidor a tarjeta programable Fuente: autor*

Se realiza exitosamente la comunicación, teniendo un tiempo mínimo de respuesta de 0 ms y tiempo máximo de respuesta de 694 ms, siendo en promedio un tiempo de 338 ms, que se encuentra en el rango permitido.

* 1. ***TIEMPO ENCONTRANDOSE ACTIVO EL SENSOR DE MOVIMENTO Y NO EL BOTÓN***

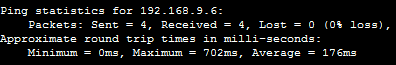
Tiempo de respuesta desde el servidor al sensor de movimiento:



*Imagen 30. ping servidor a sensor de movimiento Fuente: autor*

Se da la comunicación entre ambos dispositivos de forma exitosa, teniendo un tiempo mínimo de respuesta de 0 ms y tiempo máximo de respuesta de 129 ms, siendo en promedio un tiempo de 32 ms, que se encuentra en el rango permitido.

Tiempo de respuesta desde el servidor a la alarma de movimiento:



*Imagen 31. ping servidor a alarma de movimiento Fuente: autor*

Se presenta una comunicación exitosa, teniendo un tiempo mínimo de respuesta de 0 ms y tiempo máximo de respuesta de 702 ms, siendo en promedio un tiempo de 176 ms, que se encuentra en el rango permitido.

Tiempo de respuesta desde el servidor a la alarma de las llaves:



*Imagen 32. ping servidor a alarma de las llaves Fuente: autor*

Se consigue una comunicación exitosa, teniendo un tiempo mínimo de respuesta de 0 ms y tiempo máximo de respuesta de 1610 ms, siendo en promedio un tiempo de 403 ms, que se encuentra en el rango permitido

Tiempo de respuesta desde el servidor a la tarjeta programable:



*Imagen 33. ping servidor a tarjeta programable Fuente: autor*

Se puede visualizar una comunicación exitosa, teniendo un tiempo mínimo de respuesta de 1 ms y tiempo máximo de respuesta de 660 ms, siendo en promedio un tiempo de 166 ms, que se encuentra en el rango permitido.

1. **ANÁLISIS DE RESULTADOS**

A continuación, se muestran en tablas y gráficas los resultados obtenidos:

|  |  |  |  |
| --- | --- | --- | --- |
| **TIEMPOS SENSOR DE MOVIMIENTO** | | | |
| **Estado sensores** | **t.mínimo (ms)** | **t.promedio (ms)** | **t.maximo (ms)** |
| Sensor apagado | 7 | 14 | 18 |
| Sensor prendido con botón | 0 | 91 | 365 |
| Sensor prendido sin botón | 0 | 32 | 129 |

*Tabla 1. Tiempos sensor de movimiento Fuente: autor*

*Imagen 34. gráfica tiempos sensor de movimiento Fuente: autor*

Se observa que en todos los casos el mayor consumo de tiempo de la comunicación entre el servidor y el sensor de movimiento, se presenta cuando tanto el sensor como el botón de las llaves se encuentra prendido, con un tiempo promedio de 91ms.

|  |  |  |  |
| --- | --- | --- | --- |
| **TIEMPOS ALARMA DE MOVIMIENTO** | | | |
| **Estado sensores** | **t.mínimo (ms)** | **t.promedio (ms)** | **t.maximo (ms)** |
| Sensor apagado | 16 | 17 | 18 |
| Sensor prendido con botón | 5 | 650 | 1328 |
| Sensor prendido sin botón | 0 | 176 | 702 |

*Tabla 2. Tiempos alarma de movimiento Fuente: autor*

*Imagen 35. gráfica tiempos alarma de movimiento Fuente: autor*

Se puede evidenciar que en todos los casos el mayor consumo de tiempo de la comunicación entre el servidor y la alarma de movimiento, se presenta cuando tanto el sensor como el botón de las llaves se encuentra prendido, con un tiempo promedio de 650ms.

|  |  |  |  |
| --- | --- | --- | --- |
| **TIEMPOS ALARMA DE LLAVES** | | | |
| **Estado sensores** | **t.mínimo (ms)** | **t.promedio (ms)** | **t.maximo (ms)** |
| Sensor apagado | 0 | 17 | 35 |
| Sensor prendido con botón | 0 | 365 | 1424 |
| Sensor prendido sin botón | 0 | 403 | 1610 |

*Tabla 3. Tiempos alarma de llaves Fuente: autor*

*Imagen 36. gráfica tiempos alarma de llaves Fuente: autor*

Se observa que en todos los casos el mayor consumo de tiempo de la comunicación entre el servidor y la alarma de las llaves se presenta cuando el sensor se encuentra encendido y no el botón de las llaves, con un tiempo promedio de 403ms.

|  |  |  |  |
| --- | --- | --- | --- |
| **TIEMPOS TARJETA PROGRAMABLE** | | | |
| **Estado sensores** | **t.mínimo (ms)** | **t.promedio (ms)** | **t.maximo (ms)** |
| Sensor apagado | 0 | 17 | 35 |
| Sensor prendido con botón | 0 | 338 | 694 |
| Sensor prendido sin botón | 1 | 166 | 660 |

*Tabla 4. Tiempos tarjeta programable Fuente: autor*

*Imagen 37. gráfica tiempos tarjeta programable Fuente: autor*

Se evidencia que en todos los casos el mayor consumo de tiempo de la comunicación entre el servidor y la tarjeta programable, se presenta cuando tanto el sensor como el botón de las llaves se encuentra prendido, con un tiempo promedio de 338ms.

1. **CONCLUSIONES**

## Se observa en la investigación HAMA: A Three-Layered Architecture for Integrating Object Tracking and Location Management in Wireless Sensor Networks[1] que con la aplicación de la metodología HAMA se logra una distancia entre nodos de 20 m, teniendo un radio de sensor de 8 metros, alcanzando a recorrer 42 km/h al poseer una gran red, si se contrasta con el proyecto realizado donde se cuenta con un único sensor de movimiento con un alcance igual a únicamente el radio de un nodo de la investigación con un tiempo de respuesta en el mejor de los casos de 7ms, se observa un retardo y poco alcance en el proyecto.

Para trabajos futuros se podrá realizar una implementación HAMA en la red, logrando incorporar mayores sensores de movimiento, maximizando su distancia para así lograr más alcance del sistema con una mayor velocidad, ya que actualmente la velocidad es de 1 m/s y el objetivo es superarla.

En el artículoTracking irregularly moving objects based on alert-enabling sensor model in sensor network [2] se observa un enrutamiento dinámico, mejorando según el contexto el tiempo de comunicación entre los dispositivos IoT, todo esto se logra implementando la metodología AboT, alcanzando una velocidad de 10 m/s.

El objetivo está en lograr alcanzar dicha velocidad mediante la implementación de rutas dinámicas entre los elementos inteligentes junto con la metodología AboT para lograr alcanzar una mayor distancia con una menor velocidad que la actual.

La investigación An object tracking technique in wireless sensor network based on prediction [3] ya se ve mucho más avanzada logrando una predicción del comportamiento del sistema siguiendo la fórmula P(s)=as2+bs+c, aunque el proyecto posee sensores, aún no se cuenta con resultados tales que se logre alcanzar una predicción, se cuenta con la tendencia de valores del tiempo de comunicación según su estado, dando una correlación directamente proporcional lineal entre el estado (apagado, prendido) y su tiempo de respuesta.

El paper A scheme for low path loss in object monitoring with sensor network[4] se observa una distribución de nodos de 10 m con un radio de 15m, probándose objetos con velocidad de 10m/s siendo eficiente el programa aún con reducción de energía.

Si se contrasta con el resultado del proyecto, se puede observar una posibilidad de optimización de energía contando con la misma calidad del servicio de la red, incluso mejorándola a 10 m/s.

En el trabajo Power conservation and quality of surveillance in target tracking sensor networks[6] se obtiene una comunicación constante entre los dispositivos, alertando el movimiento de un objeto, actualmente no se cuentan con datos comparables en el proyecto, ya que no se realiza seguimiento constante al dispositivo, sólo importa su posición en un momento establecido.

Finalmente, teniendo en cuenta la distancia que se encuentran entre los dispositivos (aproximadamente 8m), el alcance máximo del sensor (7m) y el mejor y peor tiempo de respuesta obtenido tras varias mediciones (7ms y 1610ms respectivamente) se llega a la conclusión a nivel cuantitativo que la velocidad es de 1 m/s, si bien esta velocidad no es óptima, se ve la forma de optimizarla mediante la aplicación de metodologías como AboT y Hamma las cuales con la aplicación de nodos logran un alcance de 20m de distancia por sensor con un radio de detección respectivo a 15m y una velocidad de respuesta de 10 m/s.

Cabe resaltar que es de tomar como prioridad la optimización del sistema en el caso donde se encuentra el sensor prendido junto con el botón ya que el sistema tiende a volverse volátil con un tiempo promedio de 166 ms en tarjeta programable, 650 ms en la alarma de movimiento, y 91 ms en el sensor de movimiento.

# **REFERENCIAS**

|  |  |
| --- | --- |
| [1] | J.-M. H. C.-H. L. Chao-Chun Chen, «Hama: A three-layered architecture for integrating object tracking and location management in wireless sensor networks,» *IEEE,* pp. 268-275, 2009. |
| [2] | C.-C. C. a. Y.-C. Chung, «Tracking irregularly moving objects based on alert-enabling sensor model in sensor networks,» *IEEE ,* vol. 1, pp. 571-577, 2005. |
| [3] | Q. N. C. Y.-r. Yang Hai-bo, «An object tracking technique in wireless sensor network based on prediction,» *IEEE,* pp. 3-8, 2009. |
| [4] | J. H. K. a. H. Y. Y. T. H Trinh, «A scheme for low path loss in object monitoring with sensor network,» *IEEE,* pp. 749-753, 2007. |
| [5] | M. I. S. A. A. G. A. F. R. a. A. W. R. Handayani, «Alert, monitoring and tracking for electronic device prototype,» *IEEE,* pp. 1-4, 2017. |
| [6] | P. M. Chao Gui, «Power conservation and quality of surveillance in target tracking sensor networks,» *ACM,* 2004. |
| [7] | C. Antonio, «Deploying multiple interconnected gateways in heterogeneous wireless sensor networks: An optimization approach,» *ACM,* 2010. |
| [8] | «A multi-objective approach for data collection in wireless sensor networks,» *Christelle Cailloute,* pp. 220-223, 2011. |
| [9] | P. Kuila, «Heap and parameter-based load balanced clustering algorithms for wireless sensor networks,» *ACM,* vol. 4, pp. 413-432, 2015. |
| [10] | «Intelligent clustering protocol design and optimisation in wireless sensor networks,» *ACM,* vol. 20, nº 2, pp. 190-213, 2018. |

1. Isabel Victoria Pérez Díaz. Universidad Distrital Francisco José de Caldas, última modificación realizada el 26 de febrero del 2019. [↑](#footnote-ref-1)