



Trabajo Fin de Grado

Configuración, instalación e integración de un sistema RTLS para un espacio interactivo

Project JUGUEMOS: setting and integrating an RTLS system in an interactive space aimed to pervasive games

Autor

Alejandro Dieste Cortés

Director

Dr. Javier Marco Rubio

Ponente

Dra. Eva Cerezo Bagdasari

Escuela de Ingeniería y Arquitectura
2016

Configuración, instalación e integración de un sistema RTLS para un espacio interactivo

RESUMEN

Este trabajo, que está dentro del proyecto JUGUEMOS del grupo Affective Lab, consiste en establecer un Sistema de Localización en Tiempo Real de la empresa Ubisense mediante el cual poder monitorizar las posiciones de unos objetos denominados Tags a través de unos sensores de localización dentro de un espacio definido. Concretamente este espacio está situado en una habitación del tercer piso del edificio Etopia. Este sistema dispone de cuatro sensores, once dispositivos Tag y necesita de un servidor que controle todo el sistema.

El desarrollo del proyecto se ha dividido en distintas fases, las cuales se han ido completando durante el trabajo para poner en funcionamiento el sistema.

Se ha realizado una planificación de la instalación del sistema en el espacio. Se ha establecido la posición del servidor y las posiciones en las que colocar los cuatro sensores del sistema y se han tomado medidas exactas de éstas. También se ha calculado cuántos metros de cable Ethernet son necesarios para crear la red.

Tras la planificación, se ha configurado la red y el sistema de Ubisense. También se ha hecho una instalación de prueba para comprobar que el sistema funciona antes de colocarlos en su posición real dentro del espacio. Esta comprobación ha consistido en observar que los códigos de error de Ubisense en los sensores indiquen que funciona correctamente, así como comprobar que la información de los Tags se envía al servidor.

Al comprobar que el sistema funciona, se ha realizado la instalación final en el espacio con la ayuda de los técnicos de Etopia. Además, se ha vuelto a hacer una comprobación de que el sistema funciona correctamente en la posición final.

El sistema debe estar integrado con el resto de la red JUGUEMOS, por lo que se ha desarrollado una aplicación en C# mediante la cual se conecta el servidor del sistema de Ubisense con el servidor de JUGUEMOS. De esta manera y mediante comunicación OSC la información de los Tags que se almacena en el servidor se envía automáticamente a la red JUGUEMOS.

Por último, para mejorar el rendimiento del sistema se han realizado una serie de optimizaciones y calibraciones que hacen la tasa de refresco de los datos sea en tiempo real y tenga un error en la precisión menor a 30 centímetros.

Agradecimientos

En primer lugar, quiero dar las gracias al director del proyecto, Javier, sin cuya inestimable ayuda no habría sido posible. Ha estado desde el principio dispuesto a resolver cualquier pregunta y prestarme ayuda cuando lo necesitaba. Ha sido un verdadero placer trabajar con él.

En segundo lugar, quiero dar las gracias a Eva, sin la cual nunca habría llegado a realizar este proyecto. Cuando le fui a preguntar por el proyecto no dudó en ofrecérmelo.

En tercer lugar, quiero dar las gracias a Juan Carlos Peñas, de la empresa Iberware, por haber atendido las dudas y peticiones que se le han hecho en este trabajo.

Por último, quiero dar las gracias a Cristina y Carla, dos grandes amigas que me hablaron del proyecto cuando aún no sabía qué hacer.



DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD

(Este documento debe acompañar al Trabajo Fin de Grado (TFG)/Trabajo Fin de Máster (TFM) cuando sea depositado para su evaluación).

D./Dª. Alejandro Dieste Cortés,

con nº de DNI 77218440B en aplicación de lo dispuesto en el art.

14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de (Grado/Máster) Grado _____, (Título del Trabajo)

Configuración, instalación e integración de un sistema RTLS para un espacio interactivo

es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada debidamente.

Zaragoza, 29 de agosto de 2016

Fdo: Alejandro Dieste Cortés

ÍNDICE GENERAL

1.	Introducción	10
1.1	Contexto de desarrollo	10
1.2	Objetivos	12
1.3	Requisitos no funcionales	13
1.4	RTLS	13
1.5	Estructura de la memoria	15
2.	Planificación	16
2.1	Investigación	16
2.2	Principios de diseño	17
2.3	Plan de instalación	19
3.	Configuración del sistema	23
3.1	Configuración de la red y del servidor	23
3.2	Instalación del software de Ubisense	30
3.3	Comprobación de funcionamiento	32
4.	Instalación	35
4.1	Preparación de la instalación	35
4.2	Instalación final	38
5.	Desarrollo de aplicación	41
5.1	Aplicación para mostrar resultados	41
5.2	Integración del sistema con la red JUGUEMOS	42
6.	Optimización	45
6.1	Datos en tiempo real	45
6.2	Precisión del sistema	46
7.	Gestión del proyecto	50
7.1	Planificación	50
7.2	Tiempo dedicado	50
7.3	Gestión	51
8.	Conclusiones	52
8.1	Resultados	52
8.2	Conclusiones personales	52
9.	Anexo A	54
10.	Anexo B	59
11.	Bibliografía	60

ÍNDICE DE FIGURAS

Figura 1 Juguetes con marcadores	10
Figura 2 Foto del espacio JUGUEMOS.....	11
Figura 3 Sensor Ubisense	13
Figura 4 Tag grande Figura 5 Tag pequeño.....	14
Figura 6 Diagrama sistema	15
Figura 7 Sketch Up del espacio.....	17
Figura 8 Sensores en un espacio cuadrilátero	18
Figura 9 Ángulos de los sensores	18
Figura 10 Sensores a distintas alturas	19
Figura 11 Información. Métodos de localización y visión de Tags.....	19
Figura 12 Sensor maestro en el sistema	20
Figura 13 Medidas con puntero láser	20
Figura 14 Coordenadas y origen del espacio	21
Figura 15 Esquema sistema.....	22
Figura 16 Switch POE D-Link.....	23
Figura 17 Diagrama de red del sistema.....	24
Figura 18 Dirección IP fija del servidor	25
Figura 19 Deshabilitar el Firewall	25
Figura 20 Comprobación de Firewall.....	26
Figura 21 Paso 1: DHCP	27
Figura 22 Paso2: DHCP	27
Figura 23 Paso3: DHCP	28
Figura 24 Paso4: DHCP	28
Figura 25 Registro IGMP Level de Windows	29
Figura 26 Paquetes de instalación	30
Figura 27 PlatformControl	31
Figura 28 Servicios de Ubisense	31
Figura 29 ServiceManager	32
Figura 30 Sensores reconocidos	32
Figura 31 Códigos de error de los sensores.....	33
Figura 32 Porción de cable Ethernet al lado de un sensor	35
Figura 33 Herramienta crimpadora.....	36
Figura 34 Cable Ethernet tras retirar la protección	36
Figura 35 Estándar T568B	37
Figura 36 Cable Ethernet crimpado.....	37
Figura 37 Marca en el suelo de sensor	38
Figura 38 Instalación final - 1.....	39
Figura 39 Instalación final - 2.....	39
Figura 40 Medidor horizontal.....	40
Figura 41 Propiedades sensor maestro.....	40
Figura 42 Tabla con la información de los Tags	42
Figura 43 Diagrama de red JUGUEMOS	43
Figura 44 Propiedades de los Tags.....	45

Figura 45 Coordenadas de un sensor	47
Figura 46 Eliminación del ruido	48
Figura 47 Calibraciones del sistema.....	49
Figura 48 Diagrama de Gantt	50
Figura 49 Tiempo dedicado por fase en horas	51
Figura 50 Frecuencias (imagen proveniente de la documentación de Ubisense)	54
Figura 51 Tag Ubisense	55
Figura 52 Sensor Ubisense	55
Figura 53 Señal directa y reflexiones (imagen proveniente de la documentación de Ubisense)	55
Figura 54 Señal con radiofrecuencia tradicional y UWB (imagen proveniente de la documentación de Ubisense)	56
Figura 55 Control dinámico de Tags (imagen proveniente de la documentación de Ubisense)	56
Figura 56 Cálculo de la posición (imagen proveniente de la documentación de Ubisense)	57
Figura 57 Espacio con paredes obstruyendo la visión de algún sensor (imagen proveniente de la documentación de Ubisense).....	58

ÍNDICE DE TABLAS

Tabla 1 Coordenadas de los sensores en el espacio	21
Tabla 2 Medidas de los cables necesarios	22
Tabla 3 Campos de los paquetes OSC	44
Tabla 4 Intervalos de tiempo entre datos	46

1. Introducción

En este apartado se presenta el contexto en el que se realiza este Trabajo de Fin de Grado y se explicarán los objetivos a cumplir del mismo. Además, se explica la estructura de esta memoria.

1.1 Contexto de desarrollo

El grupo de investigación GIGA Affective Lab lleva años trabajando en proyectos de juegos híbridos, resultado de esa investigación surge NIKVision, un prototipo **tabletop** o superficie horizontal activa [MCB09]. Los juegos creados para NIKVision se basan en la interacción de los usuarios, en este caso niños pequeños, con el computador a través de la manipulación de juguetes. Dichos juguetes, que se pueden ver en la Figura 1, no necesitan ninguna modificación electrónica para poder ser usados, simplemente se le adhiere un marcador impreso en su base, lo que permite al hardware de **tabletop** NIKVision identificar y localizar objetos en la superficie del **tabletop**. Hasta el momento, las aplicaciones y juegos de NIKVision, se han aplicado en entornos educativos con niños de educación infantil y especial y más recientemente con adultos con problemas cognitivos y ancianos.



Figura 1 Juguetes con marcadores

En este momento el GIGA Affective Lab se encuentra involucrado en el proyecto JUGUEMOS: Juegos Pervasivos basados en Interfaces multimodales emocionales y agentes sociales, proyecto TIN-2015-67149-C3-1R financiado por el Ministerio de Educación y Ciencia, que tiene como objetivo el desarrollo de una plataforma hardware y software para el prototipado de juegos pervasivos en espacios interactivos. Además, el GIGA Affective Lab participa en el proyecto Cesar-Etopía, mediante el cual, el BIFI de la universidad de Zaragoza colabora con el ayuntamiento de Zaragoza para la utilización de espacios y equipamiento con fines de investigación y difusión ciudadana de la investigación científica. Gracias a esta colaboración, el GIGA Affective Lab, dispone de un espacio en Etopía y de equipamiento hardware para la implementación del espacio interactivo de Juegos pervasivos necesario para el proyecto JUGUEMOS.

En octubre del año 2015 la Universidad de Zaragoza y el Ayuntamiento de Zaragoza firmaron una adenda al convenio de colaboración sobre el programa Ciencia Ciudadana y Visualización perteneciente al Centro de Supercomputación en Aragón (CeSar). Los objetivos de esta modificación responden a la necesidad de situar a Etopia, como centro de referencia en la investigación y la divulgación científica, al mismo tiempo que esta sea compartida por la ciudadanía. Para ello, se aprobó la puesta en funcionamiento de una serie de laboratorios y el desarrollo de una programación de actividades públicas y de uso para estos espacios.

El espacio del Laboratorio de Programación y Ciencia Ciudadana, en el cual se desarrolla este trabajo, se encuentra en el edificio Etopia de Zaragoza, en la tercera planta y ocupa unos 70 m². El nombre de este espacio es JUGUEMOS. En la Figura 2 se puede ver una foto de dicho espacio.



Figura 2 Foto del espacio JUGUEMOS

El objetivo es ampliar el espacio interactivo de los juegos híbridos más allá de la superficie del **tabletop**, y abarcar toda una habitación, la cual se convierte en un espacio de juegos aumentado digitalmente con proyecciones en sus paredes y en las mesas NIKVision. Así se quiere crear un entorno para el diseño, desarrollo y evaluación de juegos híbridos para grupos de niños en los que todo el espacio y los objetos en él formen parte de la actividad de juego.

Este espacio está compuesto por 4 mesas NIKVision, sensores Kinect, **smart toys**, micrófonos y proyección de imágenes en las paredes.

Para poder expandir el espacio de identificación y seguimiento de objetos más allá del **tabletop**, e incluir seguimiento de personas en todo el espacio JUGUEMOS, se adquirió un Sistema de Localización en Tiempo Real, o Real Time Localization System (RTLS). La obtención de dicho material se adquirió mediante concurso público, en el que se buscaba la dotación de materiales para el proyecto CESAR. Se explicitó que se buscaba un sistema que pudiera realizar

localización de objetos en tiempo real, en 3 dimensiones y con una precisión con un error máximo de 50 centímetros. Dicho concurso lo ganó la empresa Ubisense [UBS16]. Ubisense es un fabricante francés de sistemas RTLS, líder global en soluciones de localización. Su sistema se utiliza en manufacturación, transporte, entrenamiento militar, cuidados de salud, logística, entretenimiento, comunicación y empresas de servicio público. Venden sistemas RTLS desde el 2002. [UBI16].

Integrando este sistema con el resto de dispositivos del espacio JUGUEMOS se permitirá ampliar las posibilidades de interacción de los usuarios con el resto de elementos del entorno. En concreto, el sistema RTLS permitirá identificar objetos y usuarios y obtener su posición en tiempo real en el espacio 3D durante el juego.

En el momento en el que se empezó el trabajo, ya se contaba con el sistema de la empresa Ubisense, entregado por el distribuidor oficial en España de Iberware. Todo el material ya se había comprado y estaba disponible en el espacio para proceder a su instalación, configuración e integración. Ese era el trabajo a realizar.

1.2 Objetivos

En base al contexto planteado en el apartado anterior y para conseguir que este sistema esté funcionando dentro del espacio interactivo hay que cumplir varios objetivos. El objetivo general de este Trabajo de Fin de Grado ha sido instalar, configurar e integrar el RTLS de Ubisense en el espacio JUGEMOS, de forma que los juegos diseñados para este espacio puedan hacer uso de la información de la posición de objetos y personas participantes de la actividad.

Para ello, se plantean los siguientes sub-objetivos:

- Configurar dicho RTLS y comprobar el correcto funcionamiento de manera independiente y separada del resto del espacio interactivo.
- Instalar físicamente el sistema en el espacio JUGEMOS.
- Integrar mediante software el RTLS con el resto del espacio interactivo.
- Crear una demostración del funcionamiento correcto tras los 3 primeros puntos.

1.3 Requisitos no funcionales

Además de cumplir los objetivos, se han establecido una serie de requisitos no funcionales que el sistema debe cumplir para que el funcionamiento sea lo mejor posible:

- El envío de datos del sistema debe ser en tiempo real.
- La precisión de localización debe tener un error máximo de 30 centímetros.

1.4 RTLS

Parte esencial de este trabajo es el RTLS (Real Time Location System) de Ubisense. Este sistema es una solución tecnológica que proporciona una solución fiable para la localización precisa de objetos en tiempo real. La empresa Ubisense proporcionó un paquete con el hardware necesario para montar el sistema. Este paquete está compuesto por 4 sensores de posición y por 11 dispositivos denominados Tags. El objetivo de este sistema consiste en la localización espacial de los Tags en un espacio designado y acotado a través de los sensores.



Figura 3 Sensor Ubisense

Un sensor es un dispositivo Ethernet estándar que utiliza ondas de radiofrecuencia para comunicarse con los Tags y de esta forma calcular su posición. En la Figura 3 se puede ver una imagen de un sensor. Un Tag es otro dispositivo pequeño que también utiliza ondas de radiofrecuencia para comunicarse con los sensores. Ubisense proporciona dos tipos de Tags con

mismo funcionamiento, pero distinta forma y tamaño. Uno de los tipos, el grande, tiene forma rectangular y mide 2 centímetros de ancho y 5 de largo. El otro tipo es más pequeño, tiene forma de cuadrado y mide 2 centímetros de ancho y 2 de largo. El grosor de ambos es menor de un centímetro. En la Figura 4 y 5 se pueden ver imágenes de los dos tipos de Tags.



Figura 4 Tag grande



Figura 5 Tag pequeño

El sistema se basa en las señales de radiofrecuencia de los Tags para calcular sus posiciones y se garantiza que el error de la precisión sea menor de 30 centímetros y que la tasa de refresco de los datos sea en tiempo real. El cálculo de las posiciones se realiza en función del movimiento de los Tags y va por turnos ya que solo se puede calcular la posición de un Tag a la vez. El sistema divide el tiempo en **time slots** (27.023 milisegundos) y si un Tag se está moviendo la tasa de refresco es mayor y se actualiza su posición cada 16 **time slots** y si está quieto se actualiza cada 32 **time slots**, aunque se puede configurar de otras formas. Esto se denomina dentro del sistema como **Faster Quality of Service** o **Slower Quality of Service**.

El sistema necesita de un servidor en el cual almacenar la información de las posiciones de los Tags que han calculado los sensores. El servidor se comunica con los sensores mediante el envío y recepción de paquetes UDP unicast y multicast.

Para tener la máxima robustez en los cálculos se designa uno de los sensores del sistema como Maestro, lo que significa que este sensor además de tener las funciones normales, es el encargado de recibir la información del resto y realizar el cálculo de las posiciones en función de todos los cálculos de los sensores. En la Figura 6 se muestra un diagrama de cómo es el sistema.

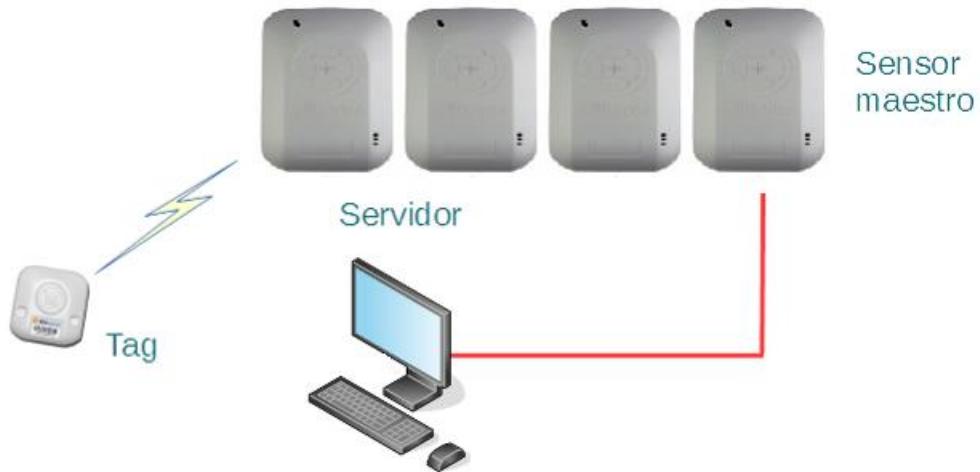


Figura 6 Diagrama sistema

En resumen, el sistema se encarga de realizar un seguimiento preciso, en tres dimensiones y en tiempo real de los dispositivos Tag. En el anexo A se explica cómo funcionan los dispositivos y el sistema de manera más detallada.

1.5 Estructura de la memoria

Esta memoria consta de los siguientes apartados:

- Planificación. Se explica el proceso de planificación del sistema, cómo se ha diseñado el sistema, y las decisiones que se han tomado.
- Configuración del sistema. Se explica el proceso de configuración del sistema, las decisiones que se han tomado y los problemas que han surgido.
- Instalación. Se explica el proceso de instalación, realizada conjuntamente con los técnicos del edificio Etopia.
- Desarrollo de aplicación. Se explica el proceso de integración del sistema mediante software con el espacio interactivo.
- Optimización. Se explica que se ha realizado para mejorar el rendimiento del sistema.
- Gestión del proyecto. Se explica cómo se ha organizado el proyecto y se presenta un resumen de las horas invertidas en el trabajo.
- Conclusiones. Se resumen la consecución de objetivos y se expone una valoración personal.
- Anexos.

2. Planificación

En este apartado se explica cómo ha sido la planificación para la colocación de los sensores, los aspectos que se han tenido en cuenta para ello y la elaboración de un plan de instalación de los sensores en el espacio interactivo.

2.1 Investigación

Antes de empezar a trabajar la planificación hubo un periodo de investigación del sistema de Ubisense. Se estudió cuáles eran los componentes del sistema y cómo funcionaban para poder realizar el trabajo de manera óptima.

Como se ha comentado, el paquete de hardware que entrega Ubisense al comprar su sistema contiene los siguientes elementos. Cuatro sensores, once objetos Tag, soportes, tornillos y tuercas para poder enganchar los sensores en el techo o la pared y la documentación del sistema. El paquete software está formado por la documentación del sistema en PDF, los instaladores de las aplicaciones Ubisense para la gestión del sistema y sus manuales de aplicación correspondiente

La documentación se divide en dos tipos de documentos. Documentos de contextualización y documentos de ayuda a la instalación y configuración del sistema. Los primeros sirven para conocer el sistema y saber de manera general cual es el funcionamiento. Los segundos siguen una estructura por capítulos ordenados de manera que sirven de guía para la configuración e instalación del sistema. Los capítulos utilizados en este trabajo son los siguientes:

- Planificación. En este capítulo se explican los pasos a seguir para realizar la instalación física del sistema. Detalla los principios de diseño que debe cumplir el sistema, como la colocación y ángulo de los sensores y distancia entre cada uno de ellos. Es decir, explica cómo realizar un plan para la posición de los sensores y un plan para el cableado.
- Configuración del sistema. En este capítulo se detallan cuáles son las características que se deben configurar en el servidor para que comience a funcionar el sistema. Se explica cuáles son los programas de Ubisense y cómo utilizarlos para realizar una configuración correcta del sistema y de la red.
- Desarrollo de aplicaciones. En este capítulo se describe la arquitectura de las aplicaciones de Ubisense, se describe también el modelo de programación para que se puedan crear aplicaciones que utilicen el sistema. Por último, se explica el programa de una aplicación ejemplo de Ubisense.

- Ajustes y soluciones de problemas. Este no es un capítulo de guía, sino que resuelve las dudas más frecuentes que surgen al estar trabajando con este sistema.

Siguiendo la estructura de la documentación, se ha organizado el contenido de esta memoria conforme a esta estructura. De la lectura de esta documentación se ha extraído la información para realizar la configuración e instalación del sistema. Además, se contaba con el apoyo del soporte técnico de la empresa Iberware al que se acudió cuando fue necesario.

2.2 Principios de diseño

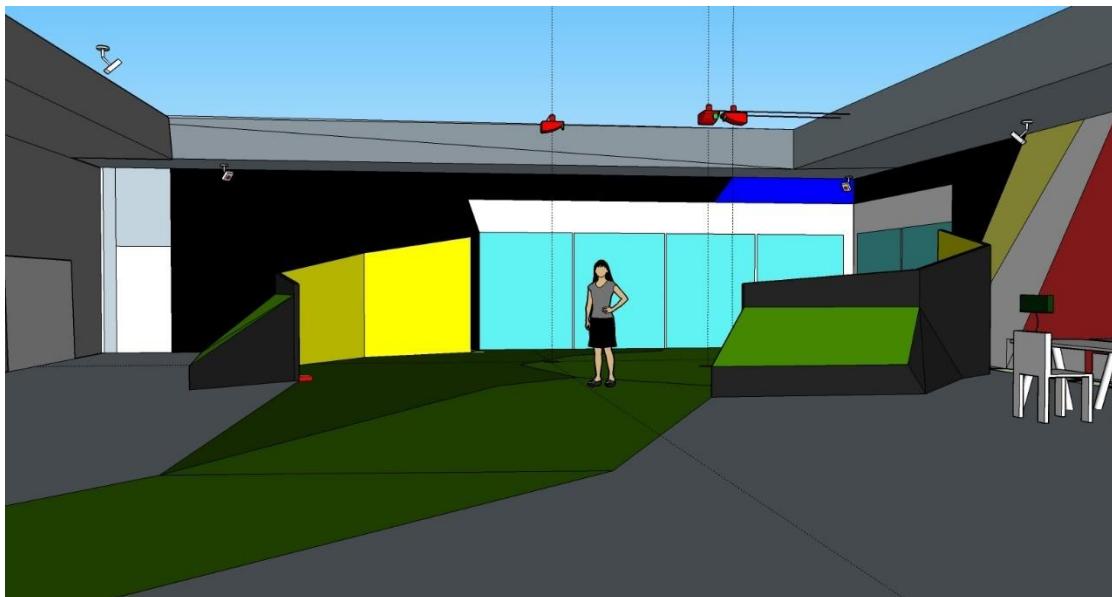


Figura 7 Sketch Up del espacio

Como se ha comentado en la introducción el espacio tiene unos 70 m² de superficie dentro de una habitación cuadrangular. Al tener cuatro sensores en total y ser un espacio cuadrangular, se decidió que los sensores se colocarían en forma de cuadrilátero para que todos los sensores tuvieran visión entre sí. En la Figura 7 se observa cómo queda el espacio JUGUEMOS tras colocar los sensores. Se ha utilizado el programa Sketch Up [SKE16] para crear el diseño de esta Figura y de las siguientes de este punto. La influencia de las paredes obstaculizando la visión tiene importancia y se explica por qué en el anexo A y

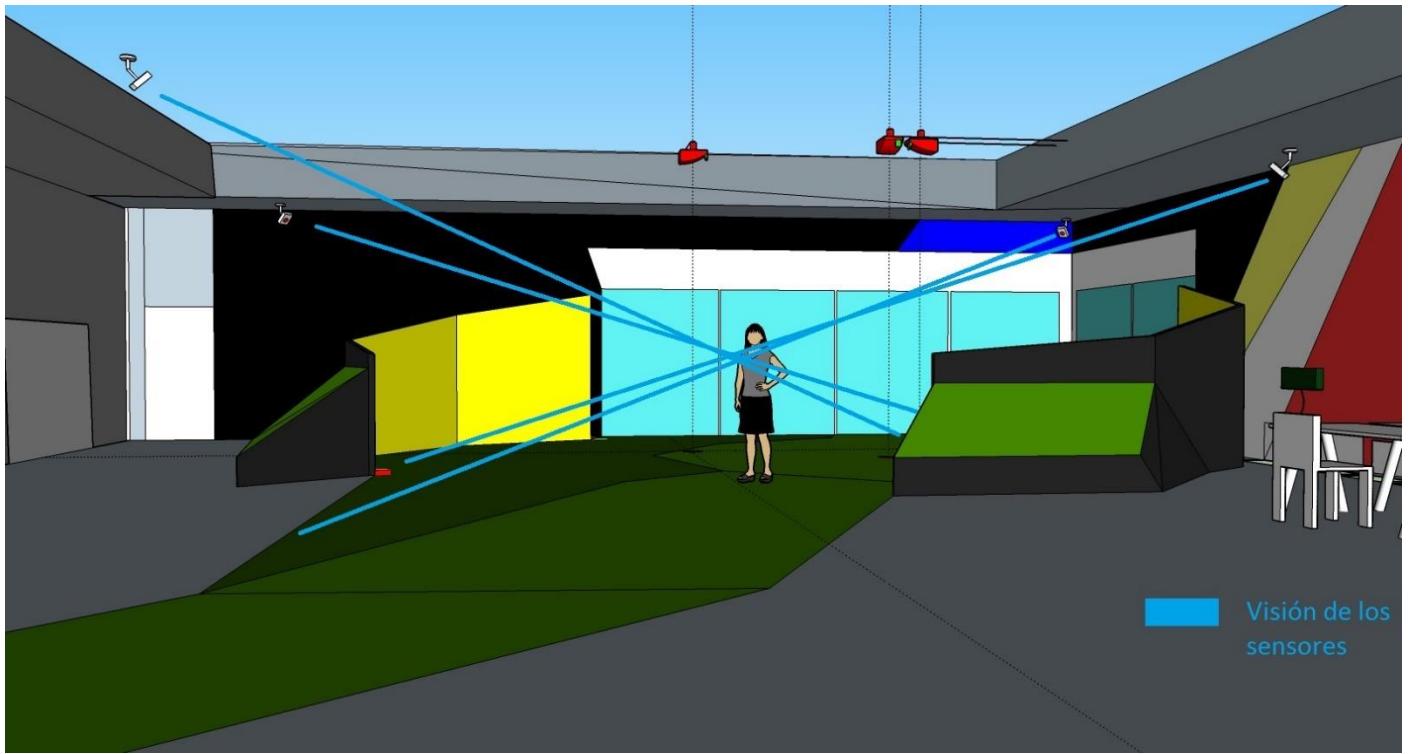


Figura 8 Sensores en un espacio cuadrilátero

el punto 2.3. Como se observa en la Figura 8 en el espacio no hay paredes altas que dificulten la visión de los sensores.

Un aspecto importante a la hora de la colocación de los sensores es que estos tienen un campo de visión limitado de 120° en horizontal y 90° en vertical. Por ello se decidió que el ángulo del sensor en su posición fuera de 45° en vertical respecto del techo y 60° en horizontal respecto a la pared. En la Figura 9 se muestra un esquema de los ángulos de los sensores.

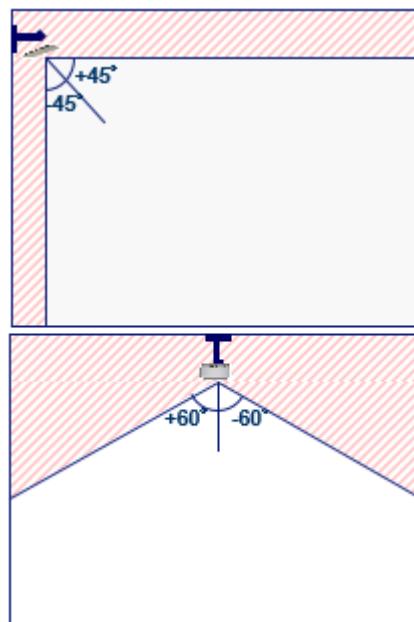


Figura 9 Ángulos de los sensores

2.3 Plan de instalación

Se decidió que la altura a la que se colocarían los sensores fuera la máxima posible, es decir, en el techo ya que de esta manera se tiene una visión amplia de este espacio y no se convierten en un estorbo para las personas que estén jugando en él. El techo de la habitación no tiene la misma altura en todos los puntos de esta, por lo que uno de los sensores estaría a distinta altura del resto. En la Figura 10 se observa como los sensores no están a la misma altura en el espacio. La altura no influye en el rendimiento de los sensores, pero en el momento de la instalación final se calculó la altura real para que se tengan en cuenta en los cálculos de las posiciones de los Tags. En el punto 4 se habla detalladamente de esto y de toda la instalación final.

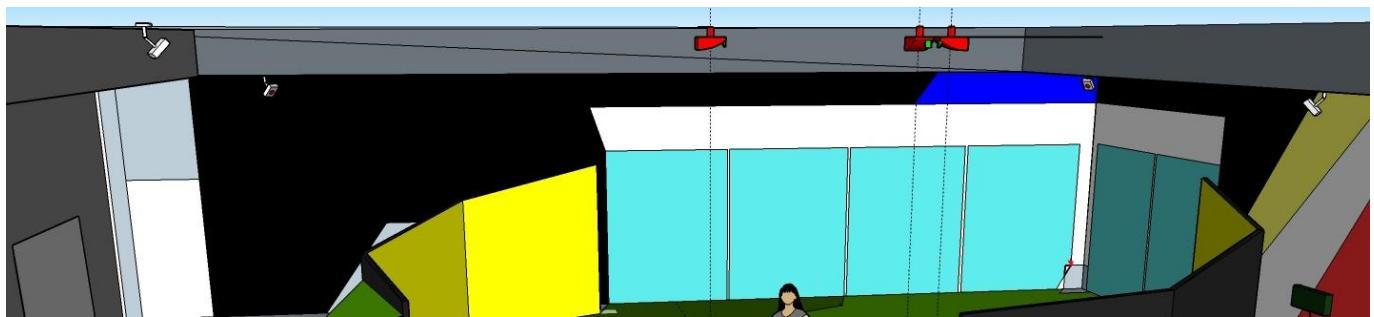


Figura 10 Sensores a distintas alturas

También se decidió que se colocaría un cable **timing** o de sincronización para tener información de diferencia de tiempos de llegada (TDAO) y tener mayor precisión. En la Figura 11 se muestra una tabla extraída de la documentación de Ubisense que muestra la influencia del método de localización, del número de sensores detectando un Tag y del conocimiento de la altura del Tag en el resultado de la posición. En el método de localización se incluye el TDAO y el AOA, que es el ángulo de llegada de la señal desde el Tag al sensor. Tras decidir tener cable de **timing** se designó uno de los sensores como maestro, concretamente el sensor que está más cerca de la ubicación donde se colocaría el servidor. Además, también todos los sensores deben estar conectados al servidor mediante cable Ethernet. Y como se ha comentado, debe haber un cable Ethernet que une el maestro con el resto de sensores para la sincronización. En la Figura 12 se observa el estado del espacio tras haber puesto los cables Ethernet.

Método de localización	Número de sensores detectando el tag	Información adicional	Resultado
Sensor único AOA	1	Altura del tag conocida	Posición horizontal 2D + altura conocida
AOA	2 o más	Ninguna	Posición 3D
TDOA + AOA	2 o más	Ninguna	Posición 3D precisa
TDOA solo	4 o más	Ninguna	Posición 3D

Figura 11 Información. Métodos de localización y visión de Tags

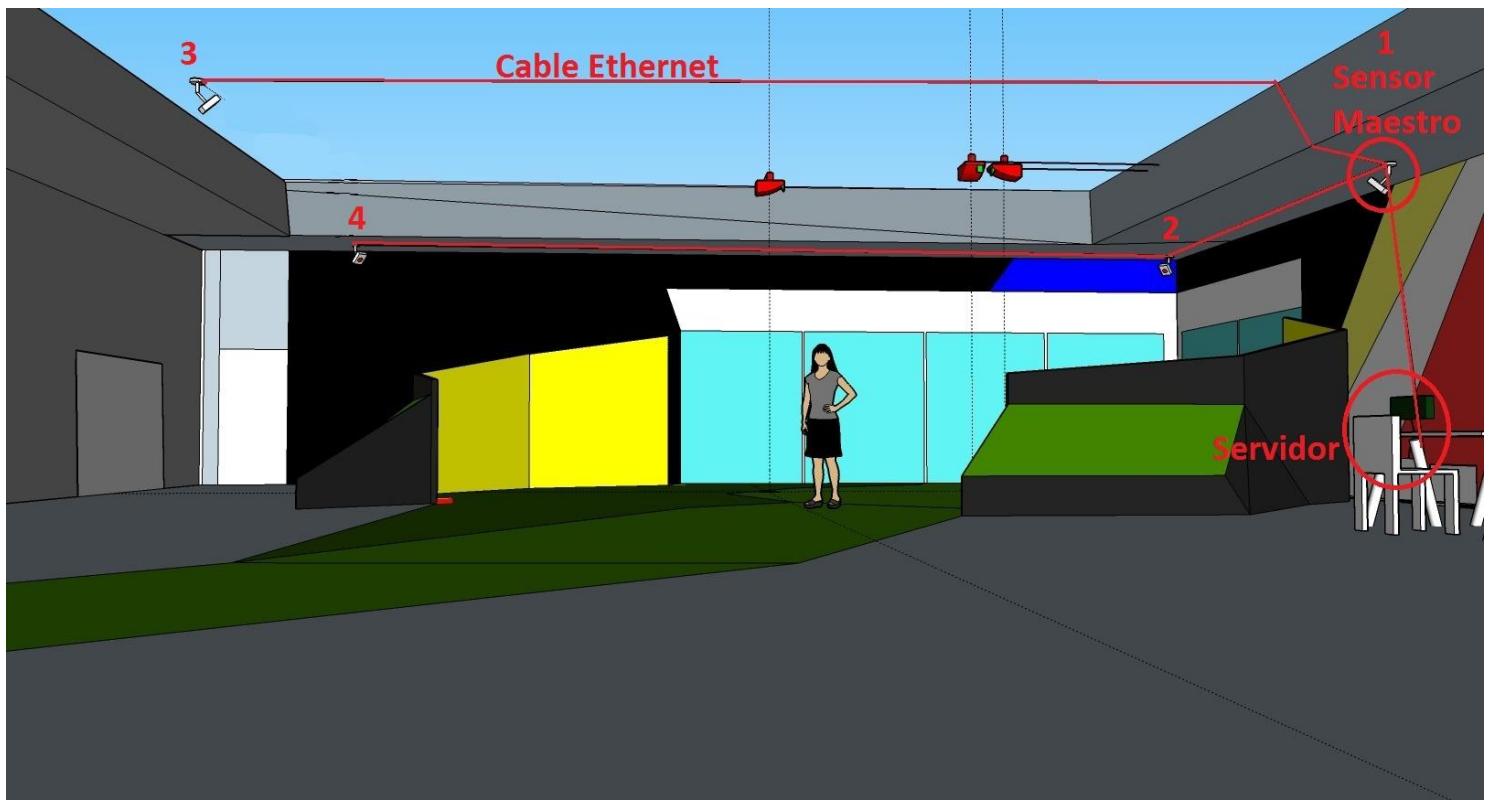


Figura 12 Sensor maestro en el sistema

Para realizar un plan de instalación correcto, se midieron las posiciones concretas en las que debían estar los sensores con un puntero láser, como se observa en la Figura 13. Un puntero láser tiene una precisión de milímetros. Además, se numeraron los sensores para que fuera más sencillo el manejo a la hora de la instalación.

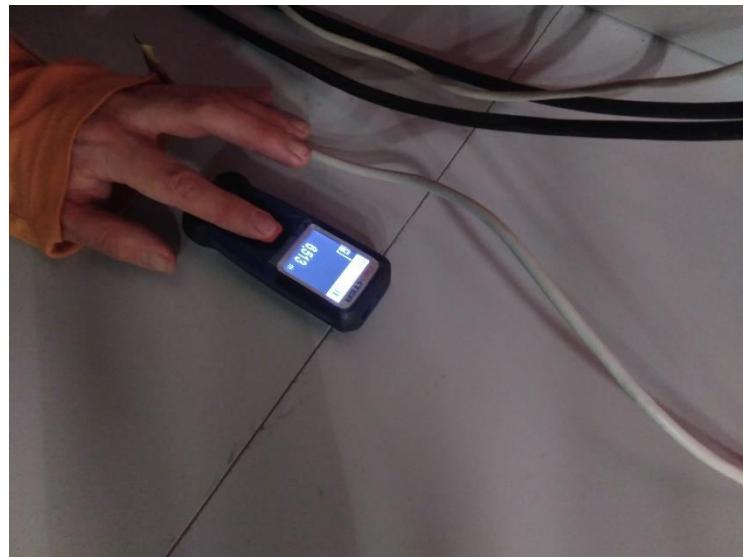


Figura 13 Medidas con puntero láser

Tomando de referencia la Figura 14, se tomó como el punto (0,0,0) del espacio del sistema la esquina inferior del fondo derecho, la coordenada **x** se tomó en dirección de la pared del fondo, la coordenada **y** se tomó en dirección de la pared de la derecha y la coordenada **z** como la altura de la habitación. Tras las mediciones en la Tabla 1 se muestran cuáles son las coordenadas exactas en las que se debían colocar los cuatro sensores.

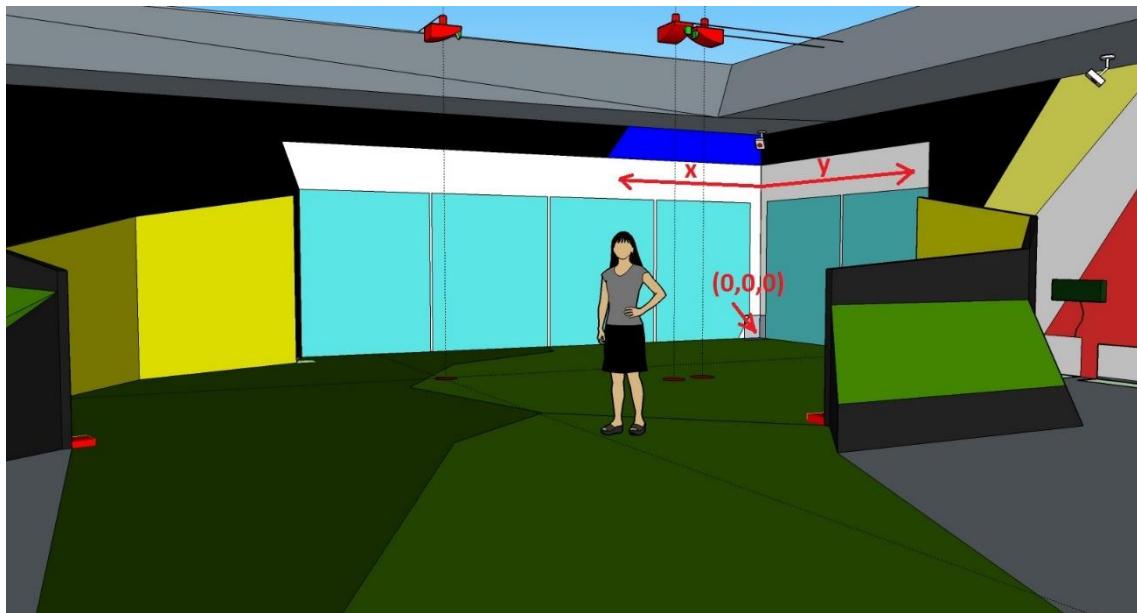


Figura 14 Coordenadas y origen del espacio

Número de sensor	Coordenada x (en metros)	Coordenada y (en metros)	Coordenada z (en metros)
1 (Maestro)	0.82	8.3	3.55
2	0.74	0.208	3.55
3	10,91	7.72	4.23
4	10.91	1.09	3.55

Tabla 1 Coordenadas de los sensores en el espacio

Después, se hicieron los cálculos para saber cuánto debía medir cada cable, además se dejó un metro de sobra para que no quedaría el cable demasiado tenso. En total 6 cables midiendo en total 80,17 metros. Finalmente se compraron 100 metros para tener margen en caso de necesitarlos. En la Tabla 2 se muestran las medidas de los cables, si son de **timing** y los sensores de origen y destino.

Número de cable	Timing	Sensores	Distancia en metros
1	No	1-2	9.09 + 3.55
2	No	1-3	10.03 + 3.55
3	No	2-4	11.14 + 3.55
4	Sí	1-2	9.09
5	Sí	1-4	20,14
6	Sí	1-3	10.03

Tabla 2 Medidas de los cables necesarios

A continuación, en la Figura 15 se muestra un esquema del plan instalación en el que se visualiza la posición y ángulo de los sensores que deberán tener en la instalación final, así como el cable Ethernet necesario (también el de **timing**).

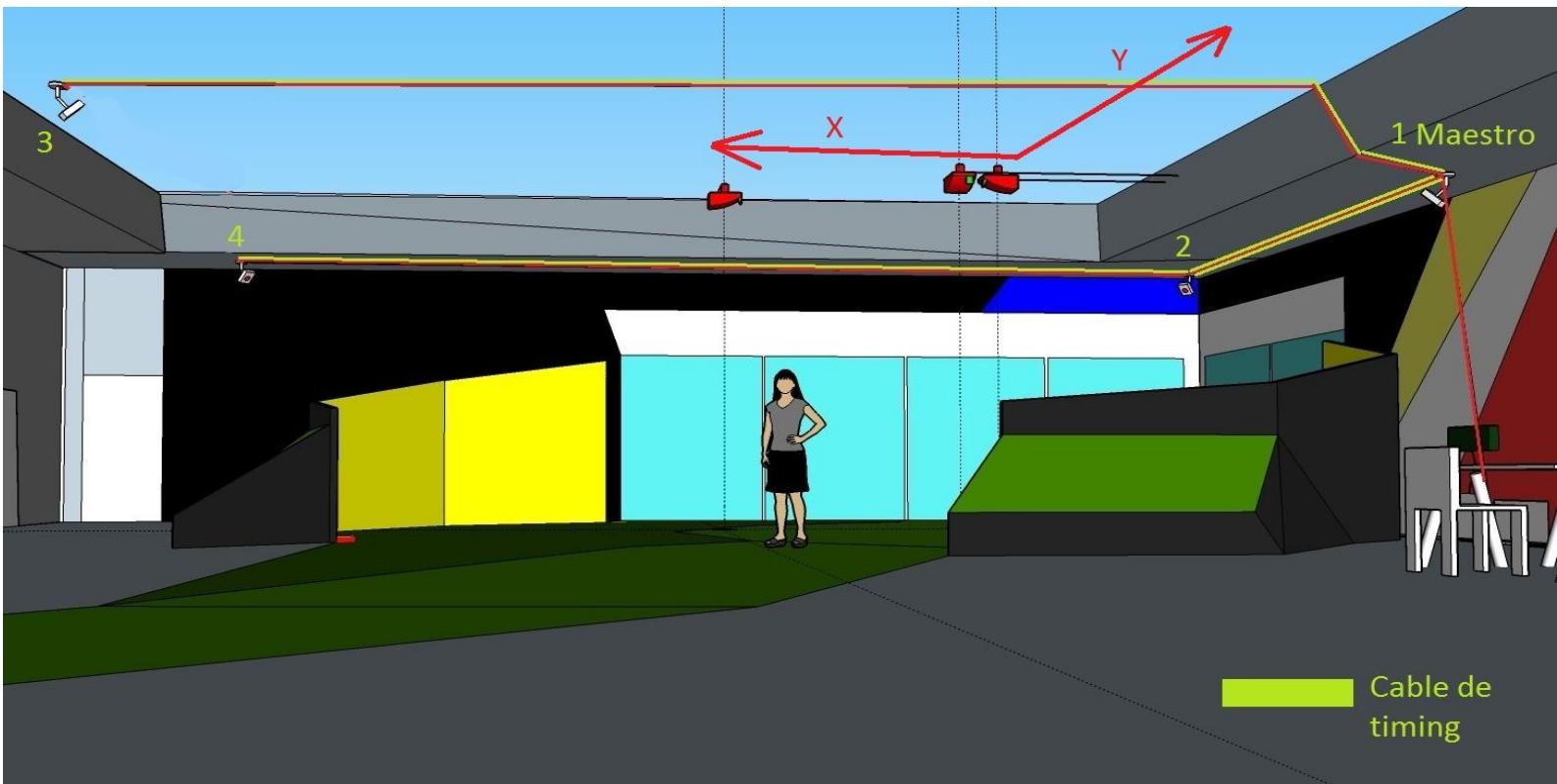


Figura 15 Esquema sistema

3. Configuración del sistema

En este apartado se explica cuál ha sido el procedimiento paso a paso para configurar el RTLS, los problemas que han surgido y cómo se han solucionado para configurar de manera correcta dicho sistema.

3.1 Configuración de la red y del servidor

La carga de trabajo que el servidor, ya mencionado en la Figura 6, de este sistema debe soportar es poca. Esto se debe a que es un sistema pequeño de 4 sensores y que gran parte del trabajo se procesa en los sensores. Debido a esto, se decidió que la máquina fuera un ordenador de medianas prestaciones y que cumpliera los requisitos del sistema. Así, la elección fue un ordenador de mesa de 64 bits con un procesador i5 y sistema operativo Windows 8 Pro. Previamente se había comprobado que el sistema era compatible con este sistema operativo.

Los sensores son dispositivos que necesitan alimentación de 12 voltios para estar encendidos y funcionar. Esto se puede hacer con cables de alimentación, lo que llevaría a instalar cable de instalación para cada sensor en el techo, o mediante cables Ethernet conectados a un switch POE (Power Over Ethernet). POE es una tecnología que incorpora alimentación eléctrica a una infraestructura LAN estándar a través de cable Ethernet [POE16]. Finalmente, se decidió seguir la segunda opción y comprar un switch POE. Tras realizar una búsqueda de switches se eligió uno de la empresa D-Link, como se observa en la Figura 16. Tiene 8 puertos de Ethernet, 4 de ellos POE y no se necesita configuración para funcionar.

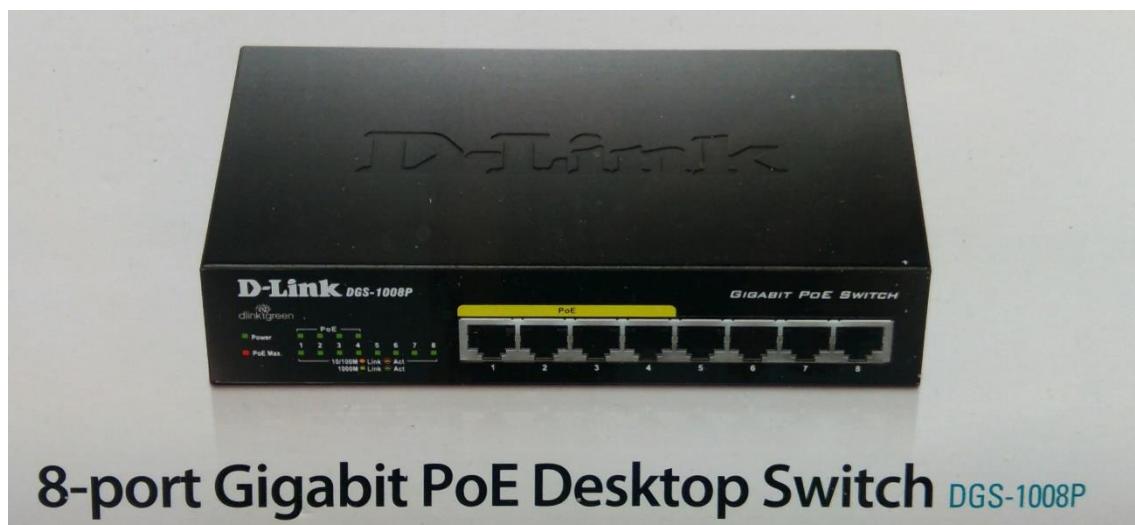


Figura 16 Switch POE D-Link

Entonces se hizo una instalación y red previa, en el suelo de la habitación, con el fin de que al terminar la configuración del sistema se comprobara el funcionamiento antes de hacer la instalación final. En la Figura 17 se muestra el diagrama de red del sistema.

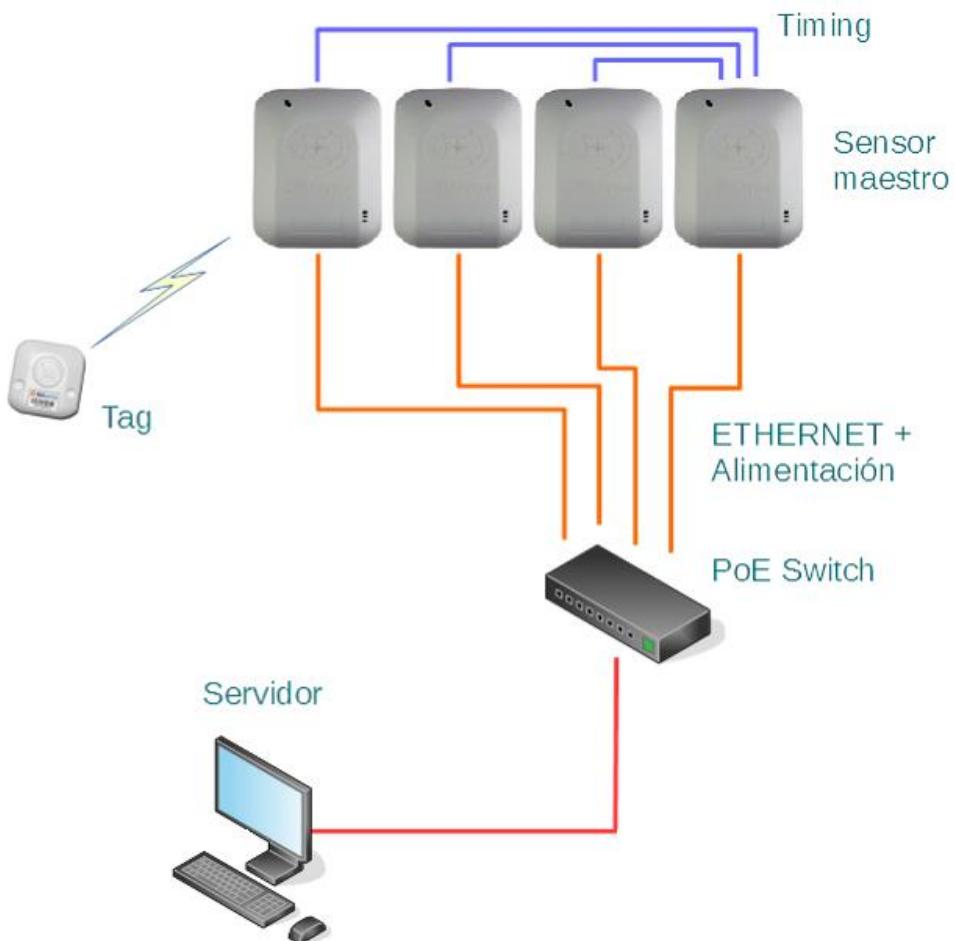


Figura 17 Diagrama de red del sistema

El siguiente paso fue configurar el servidor para que cumpliera los requisitos del sistema. Estos requisitos estaban marcados en el capítulo de configuración de la documentación y son:

- El servidor debe tener una dirección IP fija. De esta forma la conexión es más fiable y estable, importante para un servidor. En la Figura 17 se muestra la ventana de las propiedades de la red. Para establecer la dirección hay que ir centro de redes y recursos compartidos de Windows, y en las propiedades de la conexión Ethernet de la red pública asignar la dirección IP fija, en nuestro caso 192.168.0.1 y la máscara de subred 255.255.255.0.

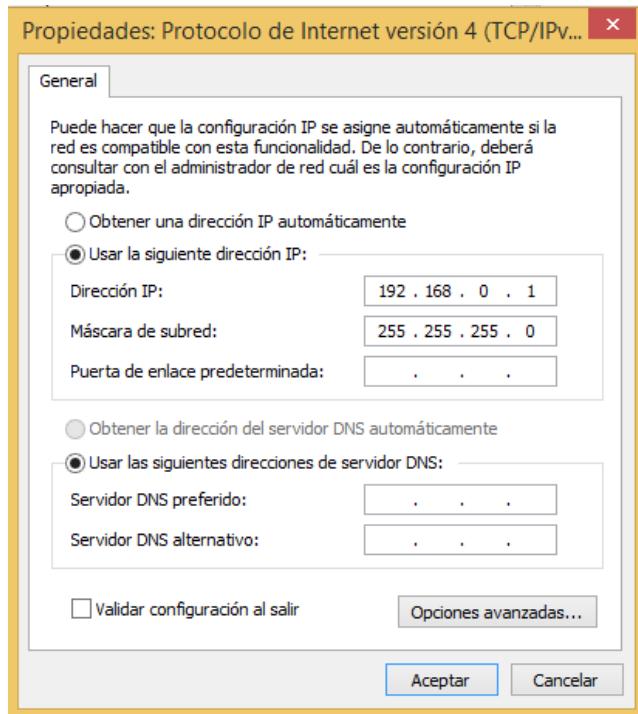


Figura 18 Dirección IP fija del servidor

- Se debe deshabilitar el firewall del servidor. Para que los envíos multicast (más adelante en esta sección se detalla porque esto es necesario) funcionen en la máquina el Firewall debe estar deshabilitado. Estamos creando una red privada que no va a tener contacto con el exterior, por lo que el firewall no es necesario y no supone un riesgo de seguridad si se deshabilita. Hay que deshabilitar tanto el firewall de redes privadas como el de públicas. En la Figura 19 se muestra la personalización del Firewall. Para hacer esto hay que ir al Firewall de Windows a través del panel de control de Windows, y entonces desactivar el Firewall tanto en la red pública como privada.

Personalizar la configuración de cada tipo de red

Puede modificar la configuración del firewall para cada tipo de red que use.

Configuración de red privada



Activar Firewall de Windows

Bloquear todas las conexiones entrantes, incluidas las de la lista de aplicaciones permitidas

Notificarme cuando Firewall de Windows bloquee una nueva aplicación



Desactivar Firewall de Windows (no recomendado)

Configuración de red pública



Activar Firewall de Windows

Bloquear todas las conexiones entrantes, incluidas las de la lista de aplicaciones permitidas

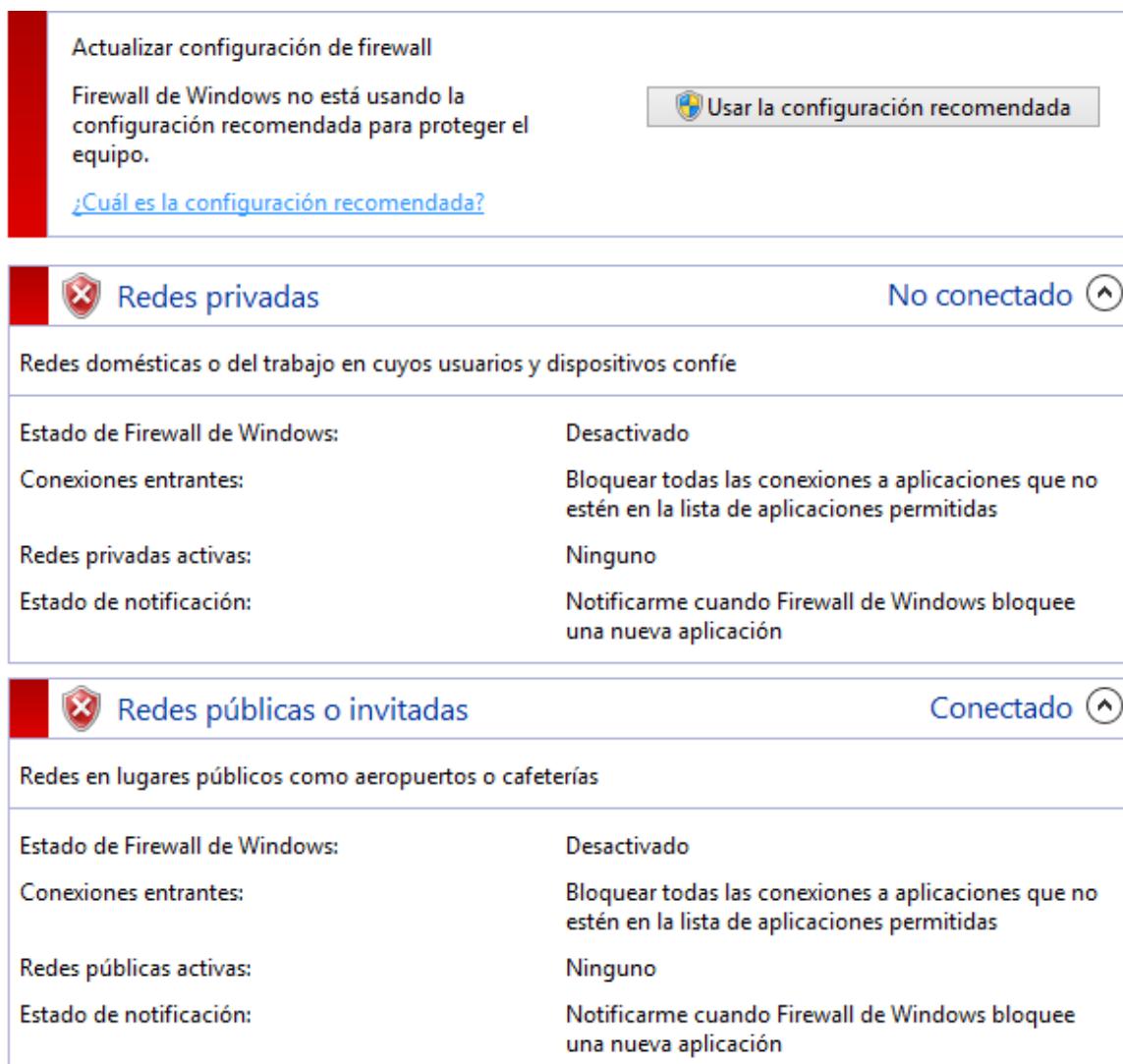
Notificarme cuando Firewall de Windows bloquee una nueva aplicación



Desactivar Firewall de Windows (no recomendado)

Figura 19 Deshabilitar el Firewall

Para comprobar que se ha realizado correctamente, se comprueba que la pantalla de “Firewall de Windows” queda así, como en la Figura 20.



The screenshot shows the Windows Firewall settings. At the top, a red banner displays the message: "Actualizar configuración de firewall" (Update firewall configuration), "Firewall de Windows no está usando la configuración recomendada para proteger el equipo." (Windows Firewall is not using the recommended configuration to protect the computer), and a "Usar la configuración recomendada" (Use recommended configuration) button. Below this, there are two main sections: "Redes privadas" (Private networks) and "Redes públicas o invitadas" (Public or guest networks). Each section contains the following information:

Redes privadas	Redes públicas o invitadas
Redes domésticas o del trabajo en cuyos usuarios y dispositivos confíe	
Estado de Firewall de Windows:	Desactivado
Conexiones entrantes:	Bloquear todas las conexiones a aplicaciones que no estén en la lista de aplicaciones permitidas
Redes privadas activas:	Ninguno
Estado de notificación:	Notificarme cuando Firewall de Windows bloquee una nueva aplicación
Redes en lugares públicos como aeropuertos o cafeterías	
Estado de Firewall de Windows:	Desactivado
Conexiones entrantes:	Bloquear todas las conexiones a aplicaciones que no estén en la lista de aplicaciones permitidas
Redes públicas activas:	Ninguno
Estado de notificación:	Notificarme cuando Firewall de Windows bloquee una nueva aplicación

Figura 20 Comprobación de Firewall

- Se debe establecer un servicio DHCP. Un servicio DHCP se encarga de asignar automáticamente una dirección IP a dispositivos que poseen una dirección MAC. Los sensores son dispositivos Ethernet estándar y cada uno de ellos tiene asignada una dirección MAC, empezando por 00:11:CE que es el rango del que Ubisense es propietario. Los sensores envían información al sensor a través de mensajes UDP. El protocolo UDP funcionan en la capa de Transporte, por encima de la de Red. Debido a esto, los sensores necesitan tener una dirección IP asignada. Por esto, es necesario un servidor DHCP. La empresa Ubisense ofrede una herramienta [UAR06] para instalar este servicio en la máquina del servidor.

Para realizar la instalación del servicio hay que abrir la herramienta `dhcpsrv.exe` y elegir nuestra red, en este caso la de dirección IP 192.168.0.1. En la Figura 21 se muestra que debe aparecer en la herramienta.

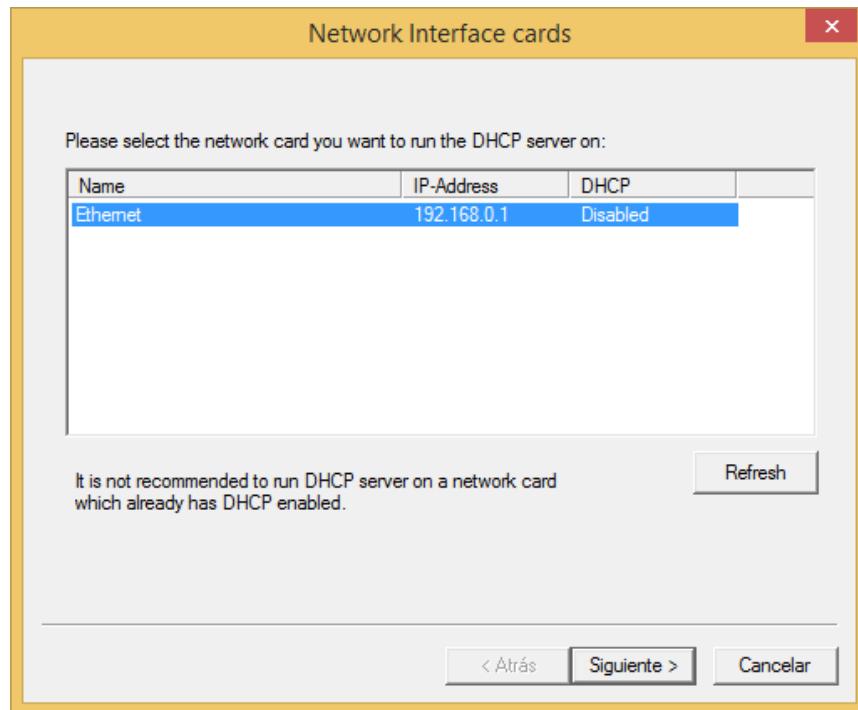


Figura 21 Paso 1: DHCP

Después hay que elegir la IP-Pool para los sensores, por ejemplo 100-254, como en la Figura 22.

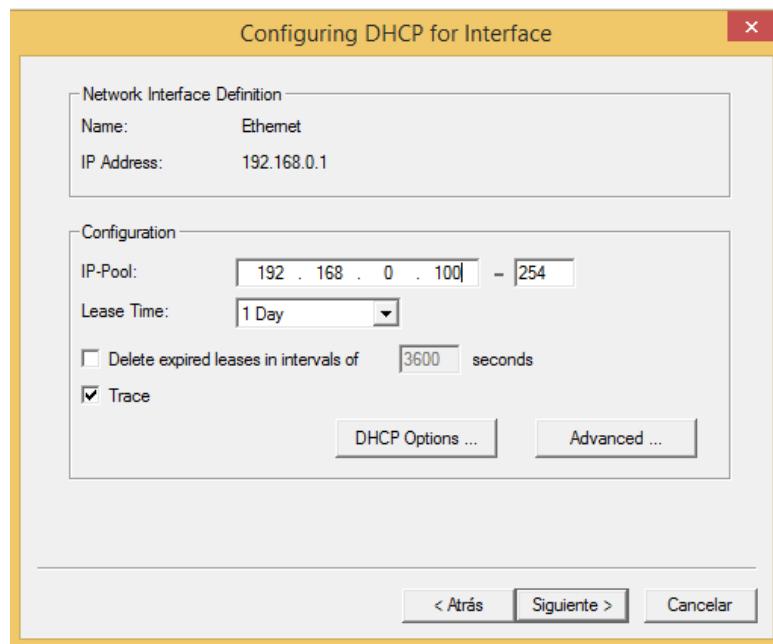


Figura 22 Paso2: DHCP

El siguiente paso es escribir el nuevo archivo INI sobrescribiéndolo por si ya estaba creado. INI [INI16] es una extensión de archivo para denotar ficheros de configuración utilizados por aplicaciones de Windows. En la Figura 23 se muestra como sobrescribirlo.

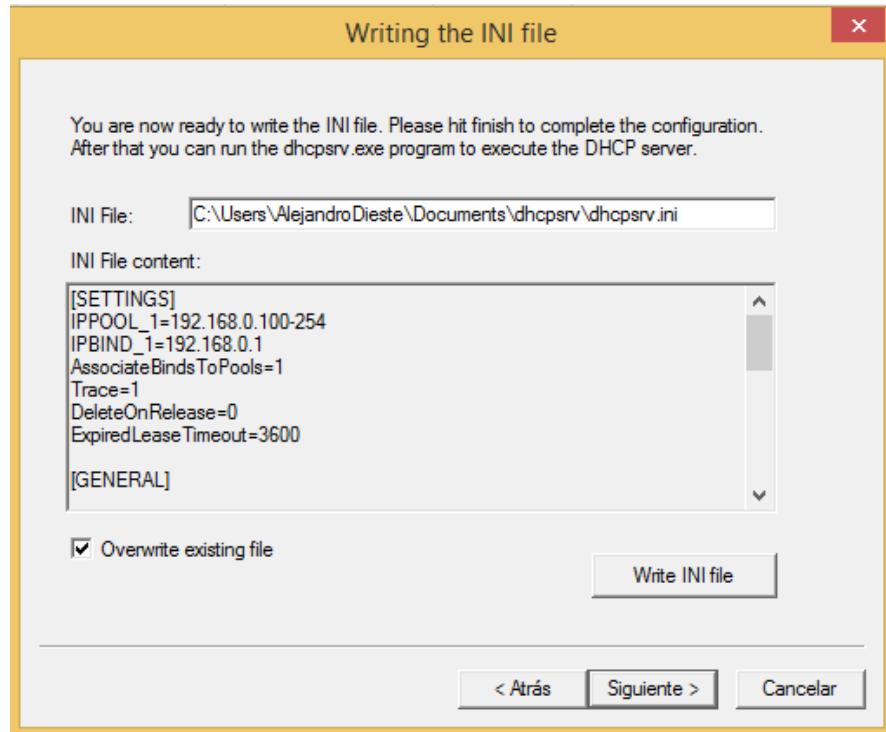


Figura 23 Paso3: DHCP

Finalmente hay que instalar e iniciar el servicio. En la Figura 24 se muestra cómo hacerlo.

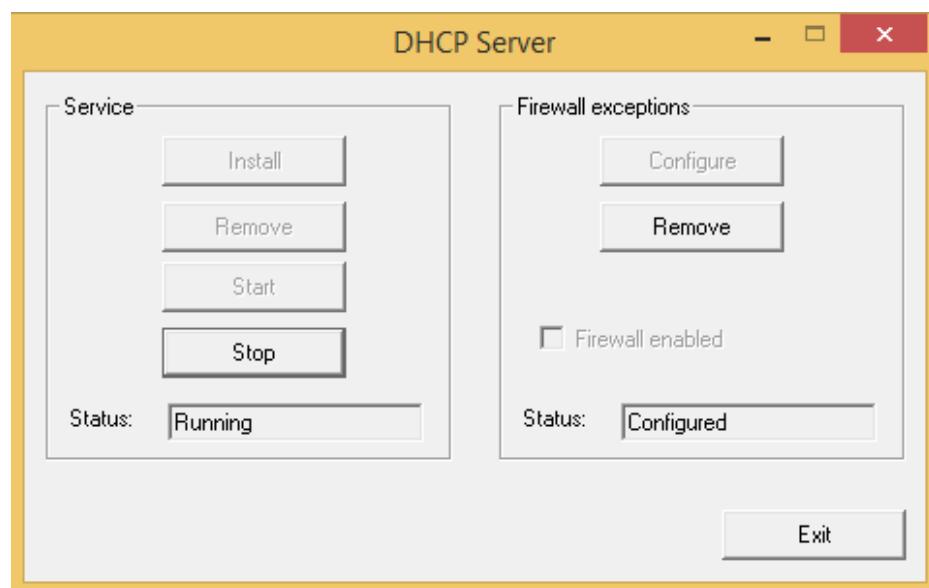


Figura 24 Paso4: DHCP

Una vez hecho esto el servicio DHCP ya estaba puesto en marcha. Para comprobar que funcionaba bien, en la misma herramienta se puede lanzar una aplicación que mostraba en una ventana emergente la dirección IP asignada a cada dirección MAC, en este caso de los sensores. Además, en el fichero INI que se ha creado se añadieron las direcciones MAC de la red con su correspondiente dirección IP.

- Para que el sistema funcione, se deben poder enviar paquetes multicast desde el servidor a todos los sensores (dispositivos con dirección IP asignada). Para que el servidor obtenga respuesta de los sensores, debe enviar un mensaje a cada uno de los sensores y estos le deberán responder con los datos de las posiciones de los TAGs. Este tipo de mensajes que son iguales y se envían a todos los dispositivos de una red especificada se denominan mensajes multicast.

Por defecto, en Windows el envío de mensajes multicast no está habilitado, además se deben tener permisos de administrador y el Firewall debe estar deshabilitado para que se puedan enviar este tipo de mensajes. En este caso, el Firewall ya estaba deshabilitado y se tenían permisos de administrador. Así que para habilitar estos mensajes se cambió el registro IGMPLevel de Windows 8 de valor 0 a valor 2. En la Figura 25 se muestra dónde está dicho registro. El registro tiene 3 valores. El valor 0 indica que no se soportan mensajes multicast, el valor 1 indica que solo se permite enviar paquetes IP multicast y el valor 2 indica que se permiten envíos y recepciones de paquetes IP multicast.

Nombre	Tipo	Datos
ab (Predeterminado)	REG_SZ	(valor no establecido)
ab DataBasePath	REG_EXPAND_SZ	%SystemRoot%\System32\drivers\etc
91 DeadGWDetect...	REG_DWORD	0x00000001 (1)
ab Domain	REG_SZ	
91 DontAddDefault...	REG_DWORD	0x00000000 (0)
91 EnableICMPRedi...	REG_DWORD	0x00000001 (1)
91 ForwardBroadca...	REG_DWORD	0x00000000 (0)
ab Hostname	REG_SZ	Alejandro
ab ICSDomain	REG_SZ	mshome.net
91 IGMPLevel	REG_DWORD	0x00000002 (2)
91 IPEnableRouter	REG_DWORD	0x00000000 (0)
ab NameServer	REG_SZ	
ab NV Hostname	REG_SZ	Alejandro
ab SearchList	REG_SZ	
91 SyncDomainWit...	REG_DWORD	0x00000001 (1)
91 UseDomainNam...	REG_DWORD	0x00000001 (1)

Figura 25 Registro IGMP Level de Windows

IGMP es un protocolo de red que se utiliza para intercambiar información acerca del estado de pertenencia entre routers IP que admiten multidifusión. Así pues, al cambiar el valor del registro IGMPLevel se añadieron las interfaces del router al que estaba conectado el servidor al protocolo IGMP. Esto permitió el envío de mensajes multicast.

3.2 Instalación del software de Ubisense

Para manejar la gestión del sistema, la empresa Ubisense proporciona una serie de paquetes que se deben instalar en el servidor. Estos paquetes o aplicaciones permiten visualizar el estado de los sensores y las posiciones de los TAGs. También dan la posibilidad de configurar todas las características del sistema en relación al espacio sobre el que funciona el sistema y la precisión y calibración de este.

Antes de empezar con la instalación del software se obtuvo la licencia del software. Una licencia de software es un contrato entre el usuario y el autor del programa informático. El usuario se compromete a cumplir una serie de términos y condiciones establecidos por el autor. Es obligatorio tener la licencia, sino ninguna aplicación funciona. Así, se envió a Iberware la información de la máquina servidor y al día siguiente se recibió un fichero licencia. Una vez obtenida, se ejecutó un instalador proporcionado por Ubisense que instaló automáticamente todos los paquetes del sistema, dichos paquetes se muestran en la Figura 26.

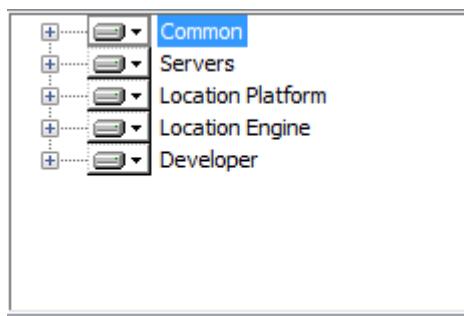


Figura 26 Paquetes de instalación

Ubisense también proporciona los servicios necesarios, que corren en el servidor, para que el sistema funcione definitivamente. Estos servicios ponen en marcha el servidor, la recepción y envío de información de los sensores y TAGs y los paquetes instalados previamente.

Dos de estos servicios se lanzan, a través de la aplicación instalada PlatformControl, como se observa en la Figura 27, el servicio del servidor y el controlador de los servicios del sistema, dichos servicios se instalan en el siguiente paso.

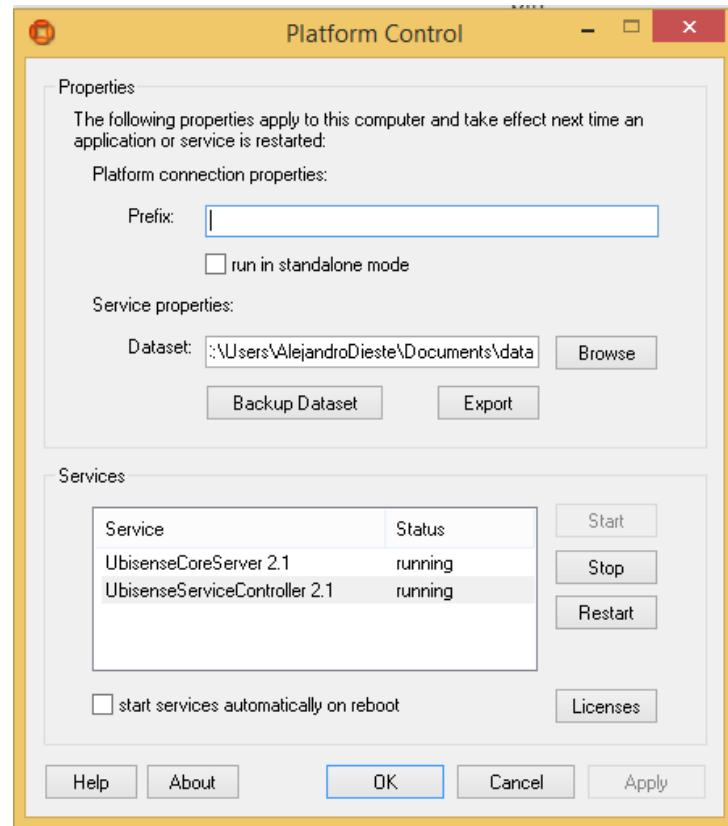


Figura 27 PlatformControl

Una vez esté funcionando el servidor se instalan los servicios comentados mediante el ejecutable ServiceInstaller, como se observa en la Figura 28. Estos servicios se encuentran en el paquete que proporciona Ubisense.

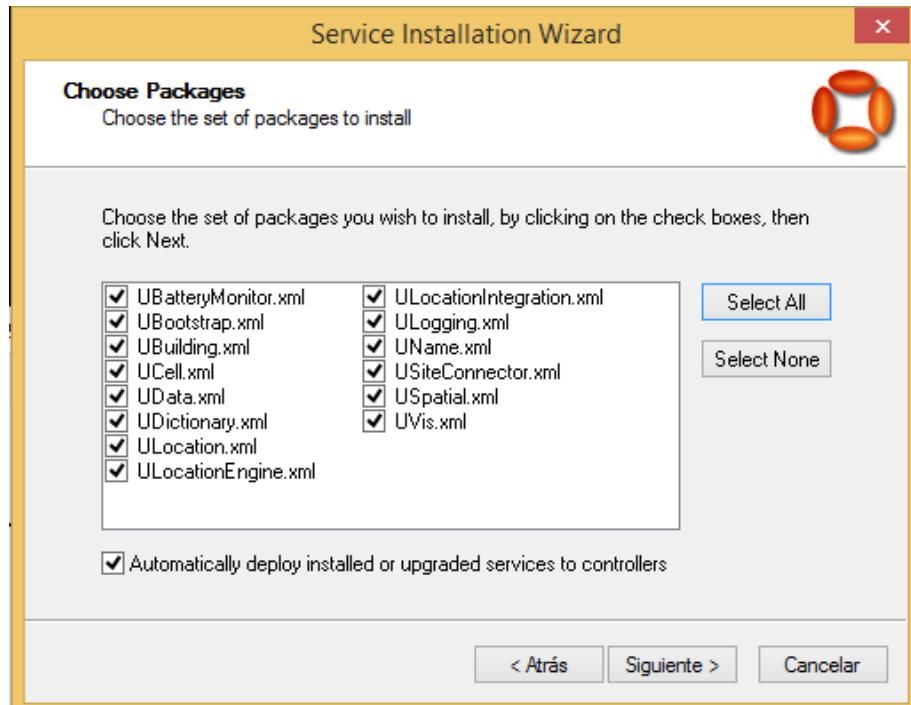


Figura 28 Servicios de Ubisense

Se puede comprobar que los servicios están en funcionamiento mediante el ServiceManager, como se observa en la Figura 29, aplicación que muestra todos los servicios que están instalados y si están o no en funcionamiento.

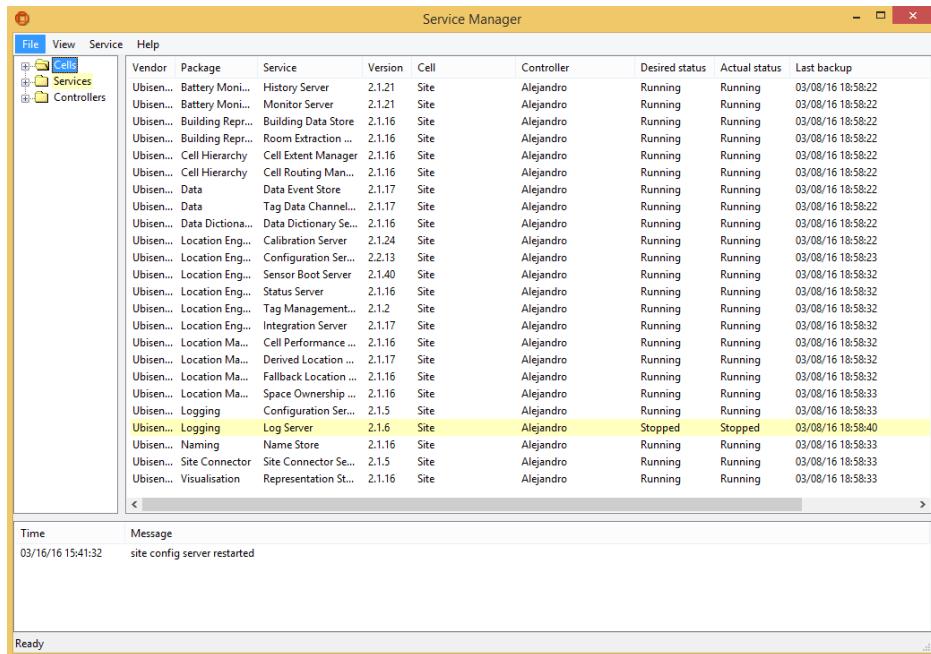


Figura 29 ServiceManager

3.3 Comprobación de funcionamiento

Si se han seguido los pasos el sistema está puesto en marcha y funcionando. Pero para comprobar que efectivamente esto es así se utiliza la aplicación LocationEngineConfig, aunque principalmente se utiliza para la calibración de los sensores y eliminación del ruido.

Al abrir la aplicación deben aparecer todos los sensores reconocidos por el servidor como se observa en la Figura 30. En este sistema se reconocen cuatro, con sus respectivas direcciones MAC, no IP.

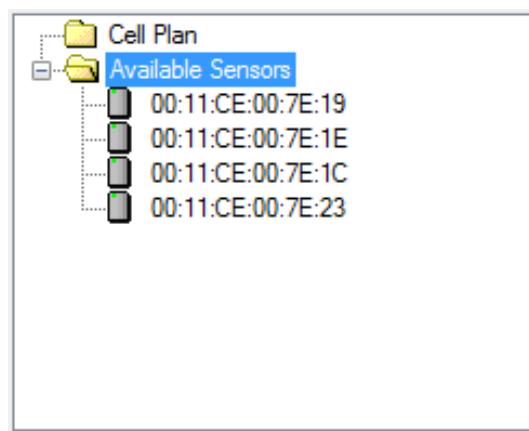


Figura 30 Sensores reconocidos

Además, los sensores tienen una luz que es de color rojo, amarillo o verde y Ubisense tiene unos códigos de color que indican en qué estado está el sistema. En la Figura 31 se observan los códigos de Ubisense. Este código son 4 destellos seguidos de alguno de los 3 colores. A lo largo de la configuración e instalación surgieron varios problemas. Todos fueron resueltos gracias al código de error de los sensores y consultando dicho código de error con el capítulo de la documentación “Solución de problemas”. En concreto surgieron los siguientes problemas:

No se conseguía poner en funcionamiento el servicio de DHCP debido a que el Firewall estaba habilitado, con el código (DHCP timeout) se averiguó que esa era la razón. Permitir el envío de mensajes multicast fue otro problema que llevó trabajo resolver. El código de los sensores indica el error (Multicast test timeout), pero no como resolverlo. Se investigó durante varios días por qué no estaban habilitados este tipo de mensajes y también la manera de habilitarlos en Windows. Finalmente se halló la forma, como se ha explicado anteriormente, y se habilitaron.

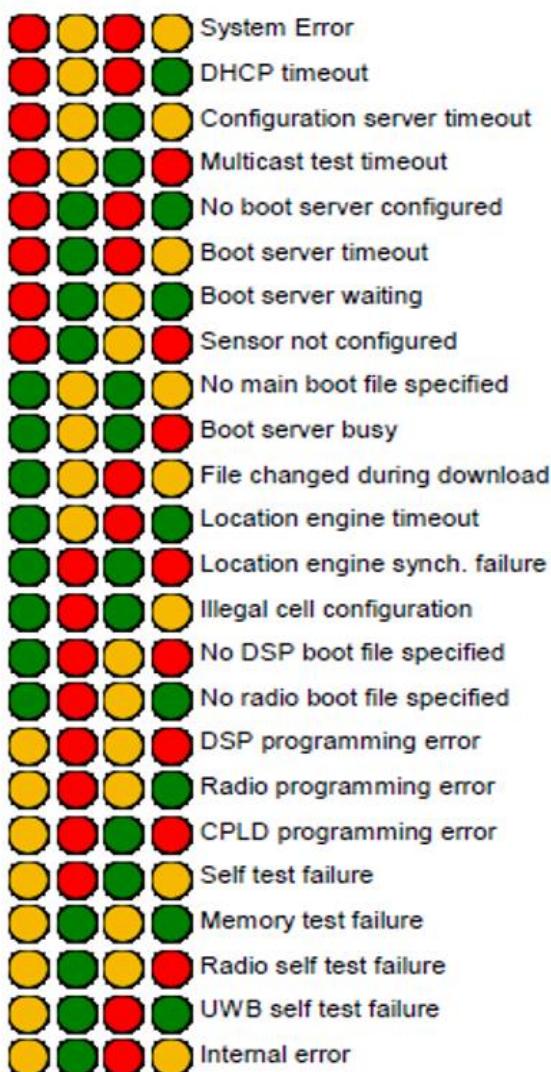


Figura 31 Códigos de error de los sensores

Tras realizar todos estos pasos se comprobó que el sistema funcionaba correctamente en la instalación previa. Las luces de los sensores indicaban 4 destellos de color verde, lo que indica que el sistema está configurado y funcionando. Así pues, se procedió a la instalación final.

4. Instalación

En este apartado se describe el proceso de instalación siguiendo el plan de instalación explicado en el apartado 2. En la Figura 32 se muestra cable Ethernet y un sensor aún sin instalar.

4.1 Preparación de la instalación



Figura 32 Porción de cable Ethernet al lado de un sensor

El primer paso era encargar el cable Ethernet. En la documentación de Ubisense se indica que el cable debe ser de categoría 5 blindado, pero que también es compatible con cable de categoría 6. El cable de cat 5 es un estándar de cables que se usa comúnmente en redes de computación como Ethernet y el blindaje sirve para evitar el ruido de otras redes cercanas. El cable de cat 6 es también un estándar de cables que posee características y especificaciones que evitan la diafonía y el ruido, sin necesidad de estar blindado. Además, la velocidad de una red con cable de cat 6 es de 10 Gigabit frente a 1 Gigabit de una red con cable cat 5. Aunque la velocidad de la red cat 6 baja cuando el cable mide más de 100 metros, pero en este trabajo ningún cable supera este valor. Así pues, se decidió que el cable fuera cat 6, por lo que se encargaron 100 metros de cable Ethernet. Se encargó un excedente en el caso de que se estropeara algún metro de cable. También se encargaron los cabezales y fundas del cable y la herramienta para unir estos al cable, se puede ver esta herramienta en la Figura 33. Se denomina crimpadora y es una herramienta utilizada para corrugar o crimpar dos piezas metálicas o de otros materiales maleables mediante la deformación de una o ambas piezas; esta deformación es lo que las mantiene unidas.



Figura 33 Herramienta crimpadora

Cuando se recibió el cable se procedió a utilizar la crimpadora para cortar el cable Ethernet y obtener los cables que necesitábamos. Posteriormente, hubo que unir los cabezales con los extremos de los cables para tener un cable Ethernet correcto para el funcionamiento. Finalmente, estos son los pasos que se siguieron para crimpar los cables.

- Con la crimpadora se retiran 2 centímetros del protector del cable por los extremos. Al hacerlo se ve que los cables del interior están enrollados por parejas, como se observa en la Figura 34.

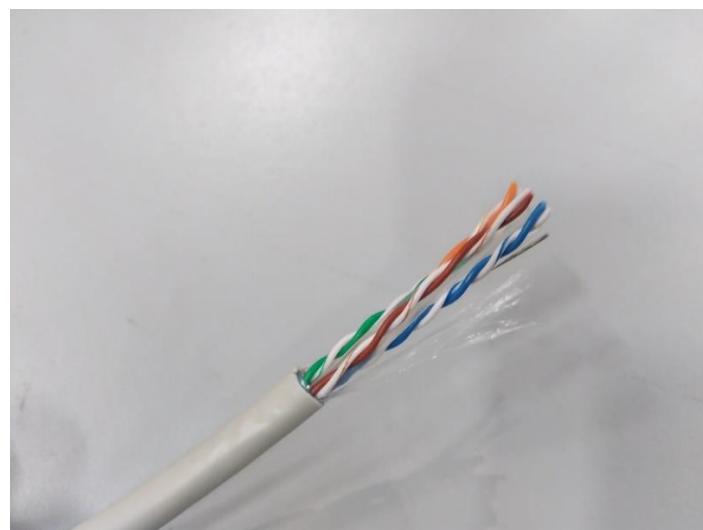


Figura 34 Cable Ethernet tras retirar la protección

- Se desenrollan los cables del interior, se alisan lo máximo posible y se colocan de la forma de la Figura 35. Naranja rayado – Naranja – Verde rayado – Azul – Azul rayado – Verde – Marrón rayado – Marrón. Este orden sigue el estándar T-568B para la conexión mediante cables Ethernet [TIA56]. Este estándar se utiliza cuando se quiere conectar un computador con un switch Ethernet, que es el caso de este sistema. Si el orden de los cables es distintos, el cable no funciona correctamente. Después se introduce la funda protectora del cable antes de poner el cabezal.

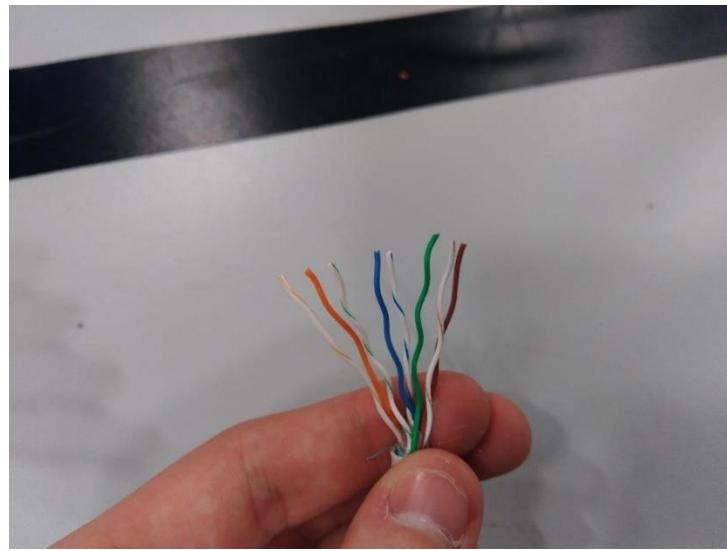


Figura 35 Estándar T568B

- Una vez se tienen los cables rectos, se meten en el cabezal de Ethernet. El orden descrito en el anterior punto se debe observar desde la parte plana del cabezal como en la Figura 36. Por último, se utiliza la crimpadora para cerrar definitivamente el cable. Una vez hecho esto en los dos extremos del cable, este ya estará dispuesto para funcionar.



Figura 36 Cable Ethernet crimpado

Se realizó este procedimiento para los 6 cables. Para comprobar el funcionamiento correcto del cable se conectaba un ordenador personal a la red de Internet (disponible en la habitación) y se observaba si dicho ordenador conectaba a la red. Si no lo hacía, se cortaba el cable por los extremos y se repetía el proceso. Cuando se tuvieron todos los cables preparados se procedió a la instalación final.

4.2 Instalación final

Para la realización de la instalación final se marcaron en el suelo, como en la Figura 37, las posiciones correspondientes en el techo de los sensores para dar una idea de donde se debían colocar los sensores. Se necesitó de la ayuda de los técnicos del edificio Etopia, ya que ellos disponen de las herramientas necesarias para anclar los sensores en el techo, además de un andamio para poder alcanzarlo. Así, se concertó una cita unos días después con dos técnicos para realizar la instalación. Se estuvo presente en ese momento para poder supervisar la instalación, es decir, controlar que los sensores quedaban como se había diseñado en la planificación.



Figura 37 Marca en el suelo de sensor

La instalación consistió en la colocación de los sensores en el techo, así como el cableado de Ethernet que conectaba los sensores y el switch. Los cables se colocaron en unas canaletas que hay en el techo del espacio. En la Figura 38 y 39 se observa parte del proceso de la instalación.



Figura 38 Instalación final - 1

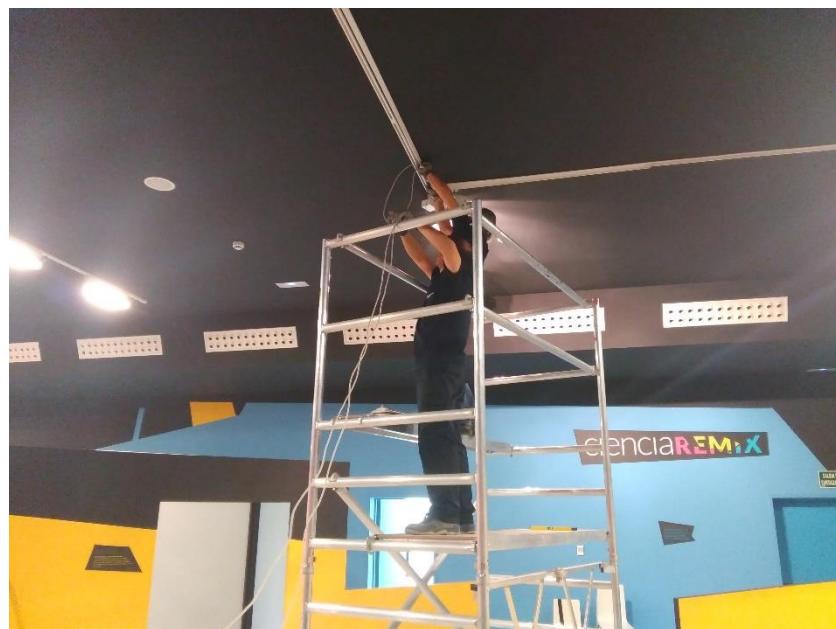


Figura 39 Instalación final - 2

Con un medidor de ángulos y un medidor horizontal, como el de la Figura 40, se colocaron los sensores con los ángulos deseados, ya que como se ha comentado anteriormente para alcanzar el máximo ángulo de visión los sensores deben de estar colocados de una forma determinada. Además, con el puntero láser se midió de manera exacta la distancia desde el suelo hasta el sensor. Esta información se utiliza más adelante en la fase de optimización y se explica en el punto 6 que se hace con ella.



Figura 40 Medidor horizontal

Por último, se conectaron los sensores al switch POE y este al servidor en su posición definitiva, como se ha indicado anteriormente en la planificación. Tras la realización de la instalación, se puso en marcha el sistema y se indicó que el sensor 1 fuera el sensor Maestro como se había planificado. Un sensor se designa maestro mediante los **Flags Timing source** y **Master** de las propiedades del sensor, como se muestra en la Figura 41. Después se comprobó que todo funcionaba correctamente, como en la instalación previa, con cuatro destellos verdes de los sensores.

El siguiente paso fue el desarrollo de la aplicación para conseguir la integración del sistema con el resto de la red JUGUEMOS.

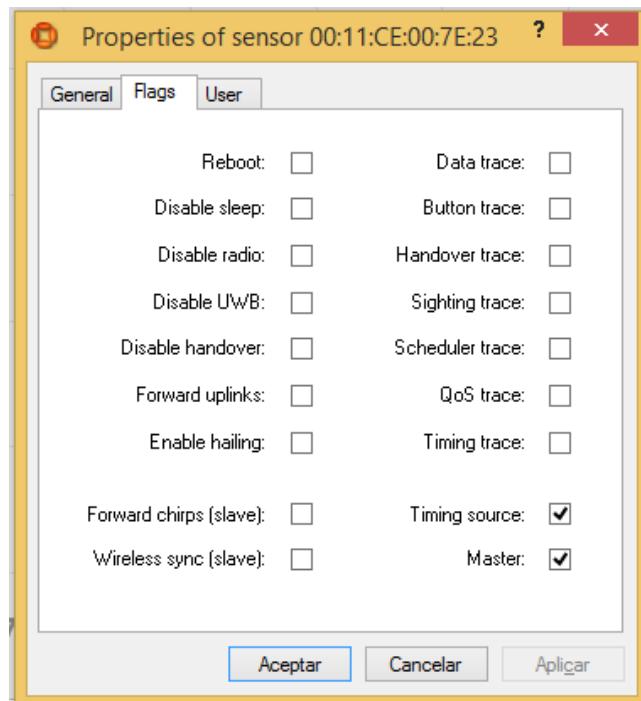


Figura 41 Propiedades sensor maestro

5. Desarrollo de aplicación

En este apartado se explica cómo se realizó una aplicación para visualizar los resultados, además de cómo se integró esta aplicación con el resto de elementos en la red JUGUEMOS.

5.1 Aplicación para mostrar resultados

Tras ver que el sistema estaba funcionando como se ha mencionado, el siguiente paso fue desarrollar una aplicación para poder visualizar los resultados. Ubisense almacena en el servidor, en memoria, todos los eventos que van sucediendo en el sistema, es decir, las posiciones actualizadas de los Tags. Aunque también ofrece de manera opcional tener una base persistente en disco, pero en este trabajo no es necesario. Ubisense ofrece una API para que se pueda acceder a esta información, aunque no permite insertar, modificar o borrar información. Está en el lenguaje de programación C# y por ello es el lenguaje que se ha utilizado para desarrollar la aplicación que muestra los resultados y posteriormente, que integra el sistema con el resto de elementos de la red JUGUEMOS.

La aplicación utiliza los objetos que ofrece la API para acceder a la información de la base de datos y mostrarla en una tabla. En concreto la aplicación realiza los siguientes pasos. Se crea un objeto esquema o **schema**. **Schema** es una clase que representa una base de datos y que permite la conexión y desconexión a esta. Se realiza la conexión con la base de datos a través del objeto. Se registra un controlador de eventos asociado al objeto **schema**. Se crea una operación de lectura en el objeto **schema** para que pueda ser consultado. Entonces cada vez que se actualiza la posición de un Tag se consultan los datos de la base mediante dicha operación y se muestran en la tabla de resultados. Por último, al cerrar la tabla o para el programa se realiza la desconexión de la base de datos y se hace **dispose** del objeto **schema**, es decir, se borra.

Nombre	Tipo	X	Y	Z	Fecha/Hora
(unnamed 00000000000000...	ULocationIntegration::Tag	2,45	6,4	0,2	17/08/2016 17:25:06
(unnamed 00000000000000...	ULocationIntegration::Tag	2,09	7,91	1,85	17/08/2016 17:25:07
(unnamed 00000000000000...	ULocationIntegration::Tag	5,03	6,59	0,97	17/08/2016 17:25:07
(unnamed 00000000000000...	ULocationIntegration::Tag	1,38	6,15	0,81	17/08/2016 17:25:07
(unnamed 00000000000000...	ULocationIntegration::Tag	2,19	8,09	1,8	17/08/2016 17:25:08
(unnamed 00000000000000...	ULocationIntegration::Tag	5,02	6,55	0,95	17/08/2016 17:25:08
(unnamed 00000000000000...	ULocationIntegration::Tag	1,89	7,49	2,45	17/08/2016 17:25:08
(unnamed 00000000000000...	ULocationIntegration::Tag	4,97	6,67	0,97	17/08/2016 17:25:09
(unnamed 00000000000000...	ULocationIntegration::Tag	5,02	6,65	0,92	17/08/2016 17:25:09
(unnamed 00000000000000...	ULocationIntegration::Tag	5,05	6,62	0,98	17/08/2016 17:25:10
(unnamed 00000000000000...	ULocationIntegration::Tag	5,01	7,41	1,58	17/08/2016 17:25:11
(unnamed 00000000000000...	ULocationIntegration::Tag	2,05	7,81	2,23	17/08/2016 17:25:12
(unnamed 00000000000000...	ULocationIntegration::Tag	5,04	6,58	1,12	17/08/2016 17:25:12
(unnamed 00000000000000...	ULocationIntegration::Tag	2,11	7,86	1,8	17/08/2016 17:25:13
(unnamed 00000000000000...	ULocationIntegration::Tag	5,05	6,55	0,92	17/08/2016 17:25:13
(unnamed 00000000000000...	ULocationIntegration::Tag	2	7,77	1,69	17/08/2016 17:25:14
(unnamed 00000000000000...	ULocationIntegration::Tag	1,9	7,68	2,19	17/08/2016 17:25:15
(unnamed 00000000000000...	ULocationIntegration::Tag	2,1	7,92	2,08	17/08/2016 17:25:16
(unnamed 00000000000000...	ULocationIntegration::Tag	1,93	7,79	1,89	17/08/2016 17:25:17

Figura 42 Tabla con la información de los Tags

La parte de visualización de resultados consiste en mostrar una tabla como la de la Figura 42, utilizando la clase de C# **DataGridView**, que permite determinar cuántas columnas tiene la tabla y cuál es su nombre. Cada fila contiene el nombre del Tag (determinado por Ubisense), el tipo del objeto, que en este trabajo siempre es Tag, las coordenadas en 'x', 'y' y 'z' (en metros) y la fecha (dd/mm/aa – hh:mm:ss) en la que dicho Tag estaba en esa posición. Las coordenadas están tomando como posición 0,0,0 el mismo punto que se ha nombrado anteriormente.

5.2 Integración del sistema con la red JUGUEMOS

Como se ha explicado en la introducción, el objetivo inicial del proyecto JUGUEMOS es la creación de juegos pervasivos para niños. Este proyecto está en un espacio también denominado JUGUEMOS y está formado por varios elementos: mesas NIKVision, **kinects**, juguetes aumentados digitalmente, etc. Todos estos elementos se comunican con una aplicación HOST, que es un programa que se encarga de ejecutar la lógica de los juegos.

A esta aplicación HOST deben llegar en tiempo real el estado de todos los dispositivos para poder reaccionar el juego a las acciones de los usuarios. La configuración de la red JUGUEMOS no ha sido el objeto de este trabajo de fin de grado, pero sí la integración del sistema Ubisense en ella. Antes de explicar cómo se ha realizado esta integración se va a detallar la arquitectura de la red JUGUEMOS, en la Figura 43 se muestra el diagrama de red.

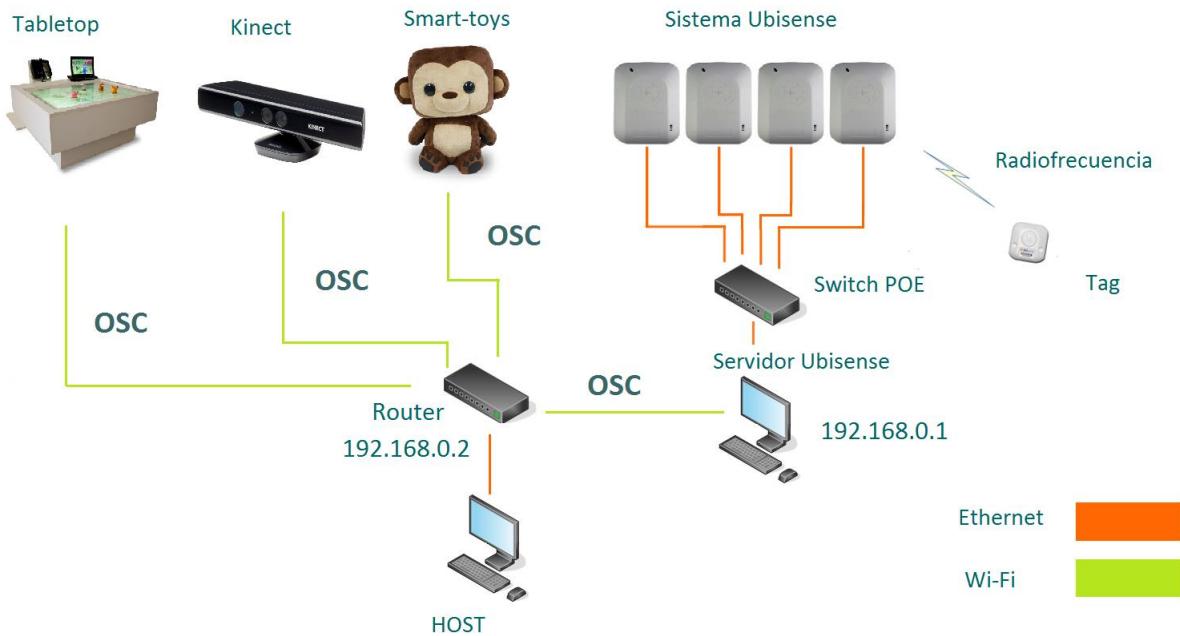


Figura 43 Diagrama de red JUGUEMOS

Como se observa en el diagrama de red, todos los dispositivos de la red JUGUEMOS están conectados por Wi-Fi a un router. A este router le mandan mensajes de red que contienen su estado. Este router que recibe los paquetes los reenvía a un ordenador que está ejecutando la aplicación HOST. Así pues, la integración del sistema Ubisense en la red JUGUEMOS consiste en enviar los mensajes de los sensores de Ubisense al router de JUGUEMOS.

En computación ubicua, hace unos años surgió la necesidad de tener un sistema de comunicación para conectar sensores heterogéneos y que estuvieran corriendo en distintas máquinas. Este sistema debía ser multiplataforma y que fuera ligero para que los requisitos de las máquinas fueran muy bajos. Así pues, nació en computación ubicua el protocolo Open Sound Control [OSC16], una alternativa al estándar MIDI que se utilizaba solo con instrumentos musicales. Con el tiempo OSC se ha convertido en un estándar, es un protocolo muy básico basado en UDP y funciona en máquinas de muy bajos requisitos y soporta distintas plataformas (ordenadores, móviles, arduino, Raspberry PI...) y sistemas operativos (Windows, iOS, Linux, Android...). Se detalla el protocolo OSC en el

anexo B. Por esta razón y porque ya se estaba usando este protocolo en la red JUGUEMOS, se decidió seguir usando OSC como protocolo de comunicación.

De este modo, para conseguir la integración del sistema, a la aplicación desarrollada se añadió código en C# para incluir la funcionalidad de comunicarse con la red JUGUEMOS a través de OSC. Esta comunicación consiste en enviar mensajes OSC a una dirección IP y puerto conocidos, los del router de JUGUEMOS. La dirección IP es 192.168.0.2 y el puerto es 32000. Se han establecido 3 tipos de mensajes formados por distintos campos:

- Mensaje de conexión, que solo contiene el patrón de dirección, la cadena de caracteres “Connect”. Cuando el router de la red recibe este paquete desde la dirección IP del sistema de Ubisense, el servidor ya no rechaza los paquetes de información. Si no se hace la conexión, por el contrario, rechaza los paquetes de información.
- Mensaje de información. En estos paquetes van las posiciones de los Tags del sistema. Estos constan del patrón de dirección, la etiqueta (que se ha utilizado como identificador del Tag) y tres argumentos. El patrón de dirección es la cadena “sensor/ubisense”. La etiqueta es el identificador del Tag que consiste en los 3 últimos dígitos del identificador de la base de datos del sistema de Ubisense. El primer argumento es la coordenada ‘x’ del tag dentro del sistema en metros. El segundo argumento es la coordenada ‘y’ del tag dentro del sistema en metros. El tercer argumento es la coordenada ‘z’ del tag dentro del sistema en metros. En la Tabla 3 se muestra un ejemplo de los campos de un paquete (coordenadas en metros).

Patrón de dirección	Etiqueta (Identificador Tag)	Arg 1 (X)	Arg 2 (Y)	Arg 3 (Z)
sensor/ubisense	043	2.81	3.14	0.7

Tabla 3 Campos de los paquetes OSC

- Mensaje de desconexión, que solo contiene el patrón de dirección, la cadena de caracteres “Disconnect”. Cuando el servidor recibe este paquete desde la dirección IP del sistema Ubisense, el servidor rechaza todos los paquetes que haya de información.

Enviando los mensajes OSC con este formato, el router se encarga de redirigirlos a la aplicación HOST para que esta tenga acceso a la información de la posición de los Tags en el espacio JUGUEMOS. De esta forma se ha conseguido la integración del Sistema de Localización en Tiempo Real en la red de JUGUEMOS.

6.Optimización

Tras realizar la integración del sistema, se habían cumplido los objetivos principales de este trabajo, pero los requisitos no funcionales no se habían comprobado. En este apartado se explica el trabajo realizado para cumplir y comprobar que se cumplen los requisitos no funcionales.

6.1 Datos en tiempo real

En primer lugar, hay que indicar que, en las propiedades de los Tags se pueden cambiar algunos parámetros que influyen en los cálculos de las posiciones. Dichas propiedades se muestran en la Figura 44. Por ejemplo, en las pestañas **Slower QoS** y **Faster QoS** se indica que según la velocidad a la que se mueve el Tag, la frecuencia de envío de información de los sensores puede ser más rápida (actualización cada 16 **time slots**) o más lenta (actualización cada 32 **time slots**). Un **time slot** como ya se ha comentado son 27.023 milisegundos. El **Threshold** es el valor de la velocidad límite para aplicar el **Slower QoS** o el **Faster QoS** y se estableció en 0.05 m/s, así cuando el Tag esté parado o prácticamente parado no se sobrecargue la red con información prácticamente igual.

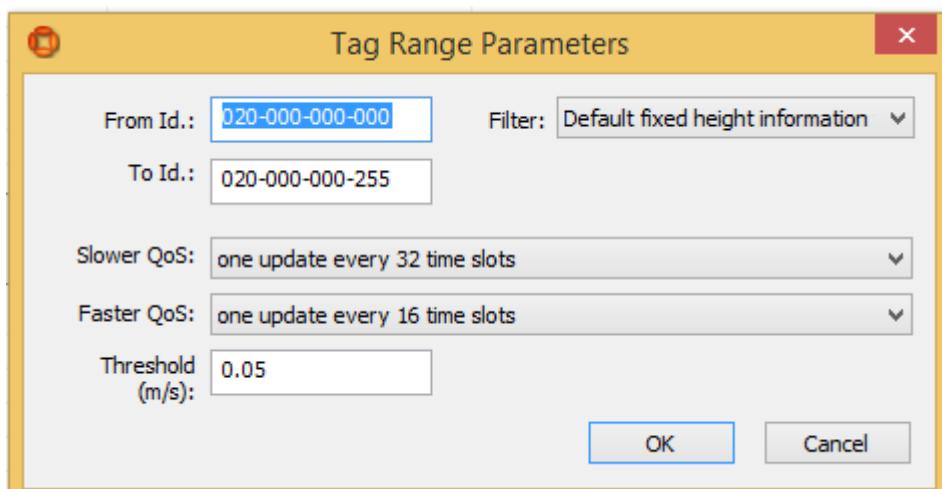


Figura 44 Propiedades de los Tags

La empresa Ubisense asegura que su sistema funciona en tiempo real, pero se decidió comprobar que el sistema montado realmente trabaja en tiempo real, es decir, si la información de localización de los Tags se actualiza en tiempo real. Para ello, basándose en la aplicación desarrollada en la que se muestran las coordenadas de los Tags y las de estas posiciones, se ha calculado el intervalo de tiempo que transcurre desde que se visualiza la posición de un Tag hasta que se actualiza en su nueva posición. Además, este cálculo se ha realizado varias veces para distintos números de Tags activos a la vez. En la Tabla 4 se ven los resultados de estos cálculos.

Número de Tags activos	Intervalo de tiempo (rango en segundos)
11	0.91 - 0.93
10	0.9 – 0.93
9	0.89 - 0.92
8	0.87 - 0.9
7	0.85 - 0.89
6	0.85 - 0.89
5	0.85 - 0.89
4	0.85 - 0.88
3	0.83 - 0.85
2	0.82 - 0.84
1	0.8 - 0.82

Tabla 4 Intervalos de tiempo entre datos

Con los resultados de esta tabla se observa que la frecuencia de actualización de los datos es de 0.93 en el peor de los casos (11 Tags enviando datos a la vez) y de 0.8 segundos en el mejor de los casos (1 Tag enviando datos). Es decir, el sistema aporta un dato por segundo para cada Tag. Esta tasa de refresco es suficiente para las necesidades de los juegos del proyecto JUGUEMOS. También, como se observa, el tiempo no es mucho menor cuando hay pocos Tags que cuando hay más, por lo que no se sobrecargará la red con mensajes constantes de un único Tag. En un sistema así, en el que los Tags los llevan personas o están en objetos que no se mueven a gran velocidad, no es necesario una frecuencia en milisegundos de envío de datos.

6.2 Precisión del sistema

Otra de las aplicaciones que se instalaron de Ubisense permite monitorizar posiciones de los Tags indicando cual es el error de precisión de dichas posiciones. Cuando se acabó de instalar el sistema se miró en esta aplicación que la precisión de los datos tenía errores mayores a un metro. Debido a esto se decidió realizar varias optimizaciones en el sistema para cumplir con el requisito no funcional de que el error fuera menor de 30 centímetros.

En primer lugar, se introdujeron las coordenadas exactas, en metros, de las posiciones de los sensores. Las medidas 'x' e 'y' se habían tomado en la planificación de la instalación. La medida 'z' se tomó en la instalación final. Conociendo las coordenadas exactas de los sensores el sistema realiza un cálculo mejor de las posiciones de los Tags. En la Figura 45 se visualizan las coordenadas introducidas a mano de uno de los sensores.

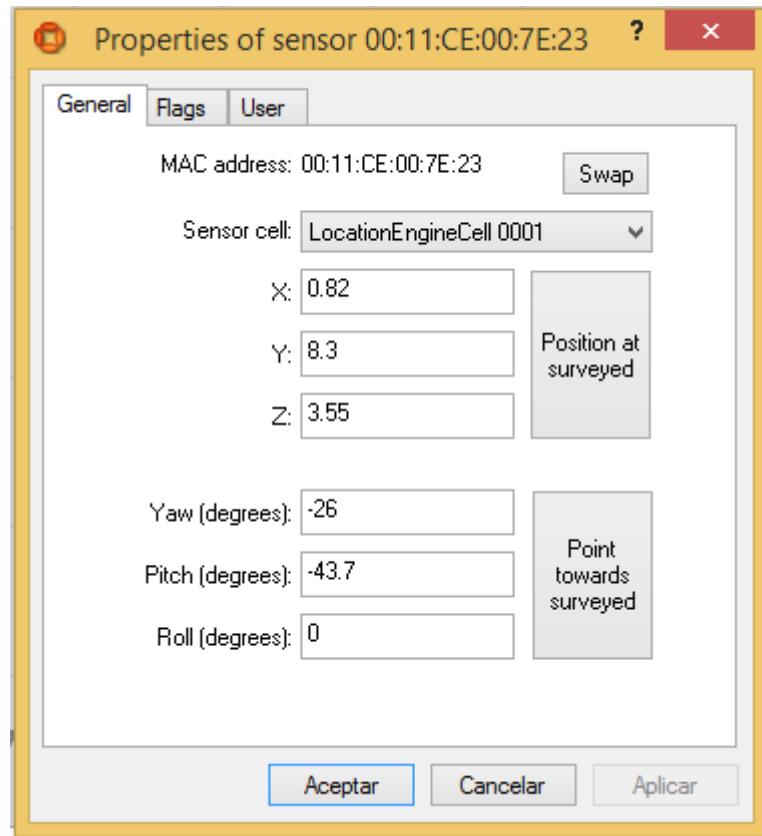


Figura 45 Coordenadas de un sensor

En segundo lugar, se redujo el ruido que afectaba a los sensores, provocado por otros dispositivos de la habitación. Para reducir este ruido Ubisense también ofrece una funcionalidad explícita para esto. En la Figura 46 se muestra la ventana de dicha funcionalidad. Con esta se estableció un **threshold** en la señal. Así, cuando los sensores detectan una señal más baja que este **threshold**, estos la ignoran y no influye en el sistema. Pero antes de establecer este límite de señal, hay que desactivar la señal de radio de los propios sensores para que solo detecten la señal del resto de dispositivos. Esto se hace marcando el **flag** “*Disable radio*” de las propiedades de los sensores. Una vez eliminado el ruido hay que desmarcar dicho **flag**.

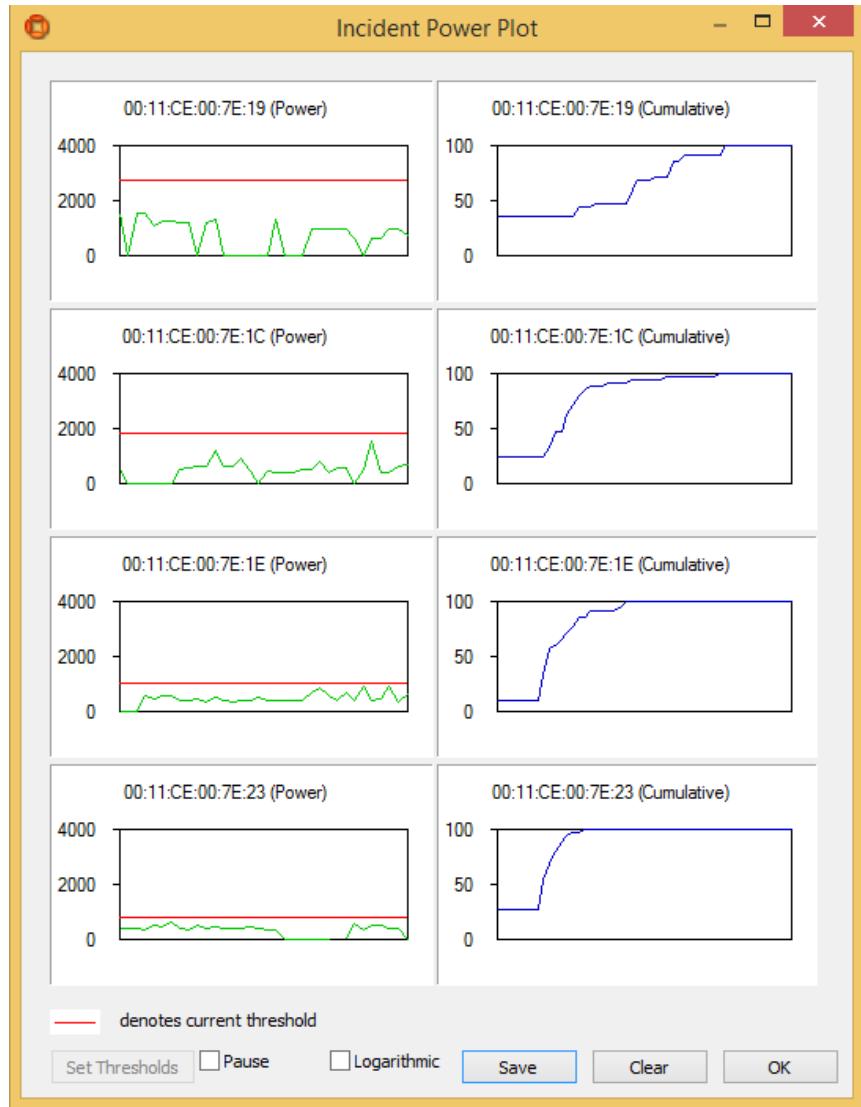


Figura 46 Eliminación del ruido

En tercer lugar, se decidió realizar varias calibraciones del sistema para que este fuera más preciso. Ubisense también ofrece estas calibraciones mediante otra funcionalidad. Las distintas calibraciones de dicha funcionalidad se muestran en la Figura 47. Primero se realizó una calibración total, que establece los ángulos en los que están colocados los sensores, lo que mejora precisión de estos. Una vez inicializada la calibración, el proceso consiste en colocar un Tag durante 10 segundos en distintas posiciones del espacio (mínimo 5 posiciones y separadas entre por lo menos 2 metros). Al acabar este proceso la aplicación de Ubisense recalculta los ángulos de los sensores como ya se ha comentado. Después de introducir las coordenadas exactas de los sensores, de eliminar el ruido y de realizar la calibración se comprobó mediante la aplicación comentada al principio de este punto que el error de la precisión había disminuido de ser mayor a un metro a ser mayor a 50 centímetros. El resultado era mejor, pero no cumplía el requisito no funcional de ser menor de 30 centímetros. Así, se realizaron otras calibraciones con el mismo procedimiento de la anterior para establecer mejor la orientación de los sensores y la distancia de cable exacta entre sensores.

Move to available	Del
Dual Calibration...	Ctrl+D
Orientation Calibration...	Ctrl+O
Cable Calibration...	Ctrl+A
Equidistant Cable Calibration...	Ctrl+E
Goto	Ctrl+G
Rename	Ctrl+R
Properties...	Enter

Figura 47 Calibraciones del sistema

Por último, en las propiedades de los Tags, en la pestaña **Filter** se indicó que se ignorara la coordenada 'z' de las posiciones para mejorar la precisión. Es decir, a la hora de realizar los cálculos el sistema no tiene en cuenta la variable 'z'. Esto supone desechar el valor de la 'z' que venga de los sensores, pero en este trabajo eso se puede hacer ya que en los juegos no se va a utilizar dicho valor. Solo se necesita la posición de los objetos en el suelo, no su altura. Tras estas calibraciones y el cambio en el filtro se observó en la aplicación de monitorización que el error de la precisión había disminuido a menos de 30 centímetros e incluso algunas posiciones no tenían error.

Con esto se concluyó que se habían cumplido también con los requisitos no funcionales del trabajo y por lo tanto con todos los objetivos.

7. Gestión del proyecto

A continuación, se va a mostrar la planificación del proyecto, así como las horas dedicadas y las herramientas utilizadas.

7.1 Planificación

La fecha de inicio del proyecto fue el día 8 de febrero de 2016, con una reunión con el director para concretar tanto como los objetivos del trabajo como la organización inicial que llevaría este. Esta reunión dio lugar a la planificación que se ha seguido en el proyecto, dividido en las fases de investigación, planificación, configuración, instalación, desarrollo de aplicación y optimización. A esto hay que añadir la última fase de documentación de la memoria y la preparación para la defensa del trabajo. En la Figura 48 se muestra el diagrama de Gantt del proyecto.

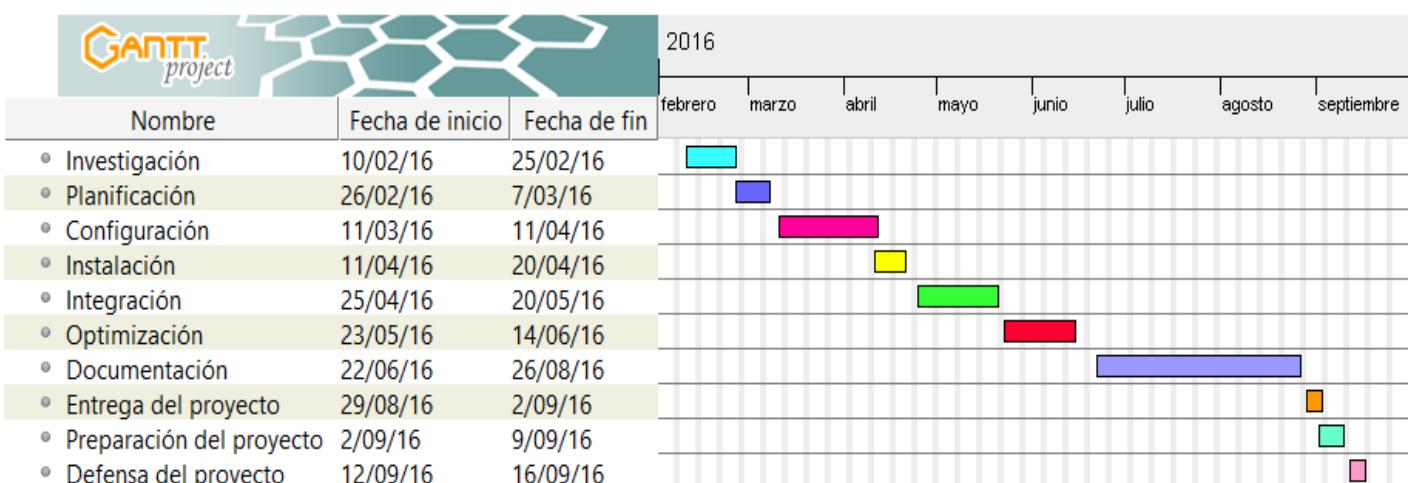


Figura 48 Diagrama de Gantt

7.2 Tiempo dedicado

Respecto al tiempo dedicado en la elaboración del proyecto, se han invertido unas 250 horas, incluyendo todas las fases, además de la documentación, como se puede apreciar en la Figura 49.

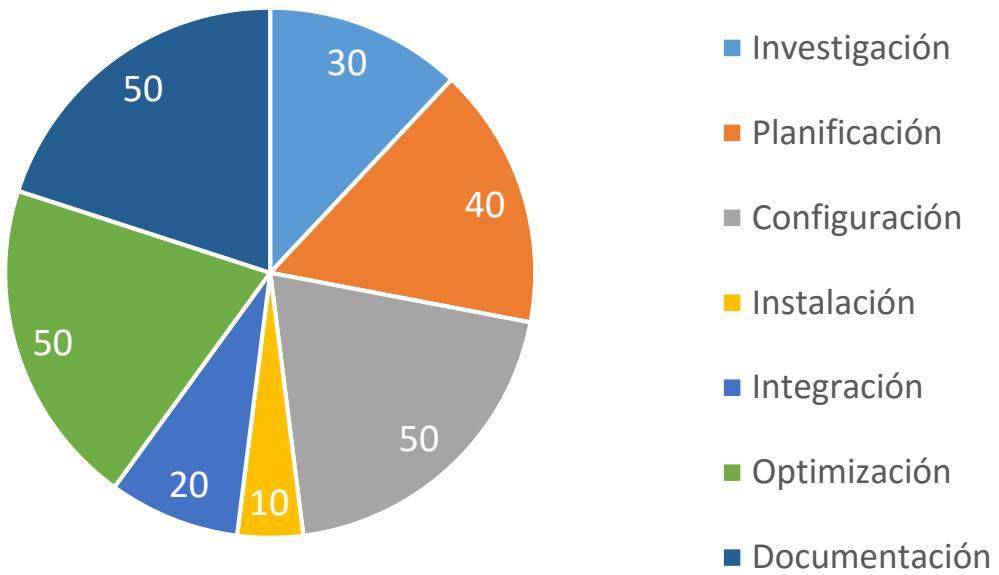


Figura 49 Tiempo dedicado por fase en horas

7.3 Gestión

Como herramienta de gestión se ha utilizado el servicio de alojamiento Dropbox. En él se ha guardado la documentación de Ubisense, un archivo de Microsoft Excel para llevar el cálculo de las horas dedicadas, un archivo Microsoft Word donde escribir la memoria del trabajo, las imágenes de esta y la aplicación desarrollada en Virtual Studio. El sistema en sí de Ubisense ha estado en el servidor, que se encuentra en el espacio JUGUEMOS del edificio Etopia, ya que solo se ha dispuesto de una licencia del sistema. Por esta misma razón todo el trabajo realizado, excluyendo investigación y documentación, se ha realizado en dicho espacio.

8. Conclusiones

Una vez acabado el trabajo y haber experimentado con él se han conseguido unos resultados y alcanzado una serie de conclusiones que se exponen a continuación.

8.1 Resultados

Se han cumplido todos los objetivos del proyecto. Se ha realizado la configuración del RTLS y se ha comprobado su correcto funcionamiento de manera independiente y separada del resto del espacio interactivo. Se ha instalado físicamente el sistema en el espacio JUGUEMOS. Se ha integrado mediante software el RTLS con el resto del espacio interactivo. Además, se han cumplido con los requisitos no funcionales marcados al principio del proyecto. Se ha conseguido que el sistema funcione en tiempo real y también se ha conseguido que el error de la precisión sea menor o igual a 30 centímetros.

Con estos resultados se puede concluir que se ha creado un sistema que funciona correctamente. Un sistema que permite monitorizar un máximo de 11 dispositivos Tags dentro del espacio de JUGUEMOS. Además, al haberse integrado con el resto de la red de JUGUEMOS, se puede utilizar dicha información para cualquier juego que se desarrolle para ese espacio. De hecho, ya se están desarrollando juegos que utilizan la información del sistema de Ubisense.

8.2 Conclusiones personales

La realización de este proyecto ha sido diferente pero muy enriquecedora, porque he hecho cosas que no me han enseñado en la carrera. Estaba acostumbrado a hacer proyectos software, calcular costes de algoritmos o teclear código para hacer un programa, pero nunca había realizado un trabajo de estas características. He trabajado con hardware comercial que vale miles de euros que debes cuidar bien para no estropearlo. No es como en la carrera que apenas pasa nada si se rompe una placa o un robot de Lego. He tenido que tratar con el personal de Etopía, con el servicio técnico de Ubisense o con proveedores de tiendas de electricidad y creo que esto es algo muy importante de cara al ejercicio profesional. Todo esto me ha aportado madurez, responsabilidad, paciencia y creo que ha sido una gran experiencia.

También he hecho cosas que se han dado, con mayor o menor grado, en la carrera. He tenido que investigar mucho sobre redes, algo que en mi especialidad no he tocado, para establecer la red interna con los sensores y el

servidor. He programado con un lenguaje que no conocía (C#) y que he tenido que aprender. Y también he realizado un cálculo de rendimiento, de tiempo y precisión para mejorar el sistema y que fuera lo mejor posible.

En resumen, he hecho cosas que he aprendido en la carrera y otras que las he tenido que aprender en el transcurso del proyecto. Y este no olvidado en un cajón, o en GitHub, ya que será utilizado en Etopia para los proyectos en los que está trabajando Affective Lab. Eso es algo que me hace muy feliz.

9. Anexo A

En este anexo se explica cómo funciona el Sistema de Localización en Tiempo Real de la empresa Ubisense. El RTLS realiza un seguimiento de la ubicación basado en tecnología de radiofrecuencia **Ultra-Wideband** (UWB), que permite envío de paquetes de información muy grandes en distancias cortas (480 Mbits/s) [UWB16] y detección de presencia basada en tecnología de radiofrecuencia convencional. En la Figura 50 se muestra un esquema de las frecuencias de la tecnología UWB.

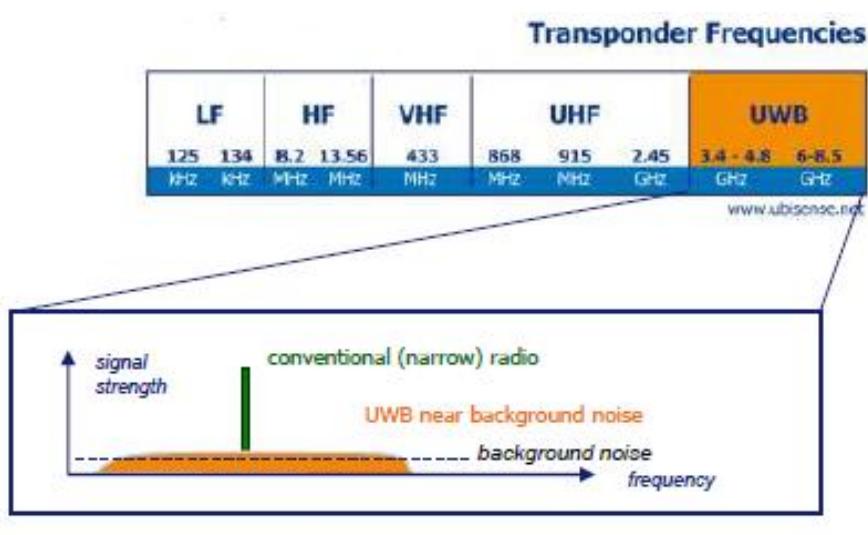


Figura 50 Frecuencias (imagen proveniente de la documentación de Ubisense)

La solución utiliza dispositivos activos UWB llamados Tag y dispositivos pasivos UWB llamados sensores. Los sensores son dispositivos conectados en red que usan el ángulo y diferencia de tiempo de la señal UWB para calcular la posición de los Tags de manera precisa. Trabajan con señales de 2.4 GHz para detectar la presencia (no confundir con localización). Soportan comunicación en 2 sentidos, es decir, puede recibir y enviar información. Los Tags son dispositivos que envían señales de radiofrecuencia para que los sensores les puedan localizar. Tienen una luz LED, zumbador, pulsadores y un acelerómetro. En la Figura 51 se muestra una imagen de un sensor y en la Figura 52 se muestra una imagen de un Tag.



Figura 52 Sensor UbiSense



Figura 51 Tag UbiSense

UbiSense garantiza una precisión de 30 cm de error en un espacio 3D y fiabilidad garantizada si al menos dos sensores tienen visión del objeto. También garantiza que la tasa de refresco de los datos de los Tags sea en tiempo real. La actualización dinámica de la posición del Tag se basa en el movimiento de este. Funciona dentro y fuera de edificios y proporciona una plataforma software para aplicaciones de lectura.

La localización de objetos mediante ondas de radio se hace a través de la medición del camino entre el transmisor y el receptor, es decir entre el Tag y el sensor. Solo la señal del camino directo determina la posición real, como se observa en la Figura 53, el resto, las señales que llegan como resultado de reflexiones con las paredes contienen error.

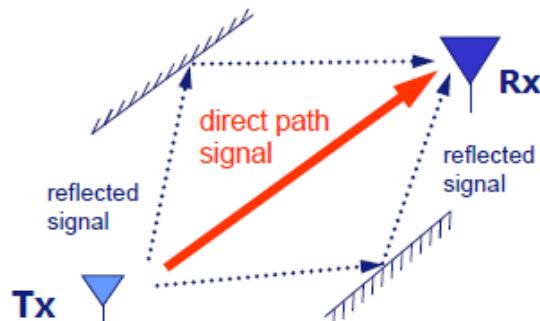


Figura 53 Señal directa y reflexiones (imagen proveniente de la documentación de UbiSense)

Con una tecnología convencional de radiofrecuencia, las reflexiones dentro en ambientes de interiores distorsionan la señal del camino directo, haciendo difícil el cálculo de la precisión. Sin embargo, con la tecnología UWB, la señal directa se puede distinguir de las reflexiones haciendo más fácil el cálculo de la precisión. Por esto UbiSense utiliza UWB. En la Figura 54 se muestra una comparación entre ambas tecnologías. A la izquierda tecnología tradicional, a la derecha UWB.

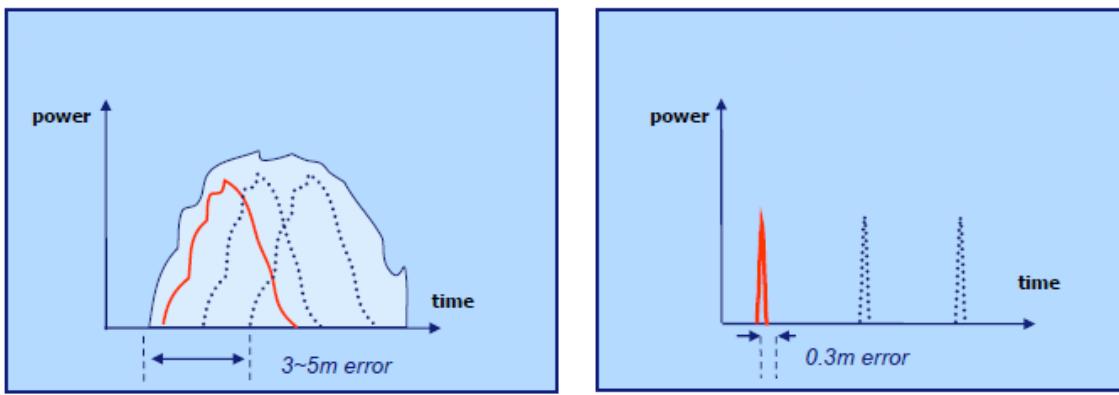


Figura 54 Señal con radiofrecuencia tradicional y UWB (imagen proveniente de la documentación de Ubisense)

A continuación, se explica cómo se realiza el control de los Tags en tiempo real. En primer lugar, hay que decir que dentro de un sistema de Ubisense se designa a uno de los sensores como **Master** o maestro

A través del canal de radiofrecuencia convencional, el sensor maestro pide a los Tags que emitan la señal UWB cuando es su turno según la configuración del sistema, hasta 10 veces por segundo. Esta configuración hace que el sistema divida el tiempo en **time slots** (27.023 milisegundos) y que la actualización de datos sea cada 16 o 32 **time slots** según el movimiento real del Tag sea rápido o lento. Estos parámetros se pueden configurar y se explica cómo en el punto 6.1. No se puede calcular para más de un Tag a la vez debido a que solo hay un canal UWB, por lo que cada Tag contesta al sensor con su señal UWB en su turno. El canal convencional es bidireccional y permite la telemetría, es decir, consultar el estado de la batería, aviso acústico, etc.). En la Figura 55 se muestra un esquema de cómo funciona el control de los Tags.

Dynamic Control of the Tags

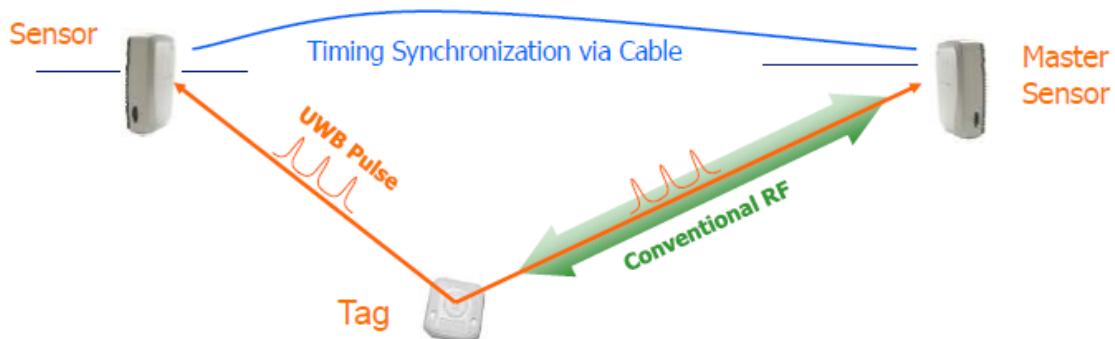


Figura 55 Control dinámico de Tags (imagen proveniente de la documentación de Ubisense)

El cálculo de la posición exacta del Tag se realiza de la siguiente manera. Las señales UWB que los sensores reciben se usan para calcular el ángulo de llegada (Angle of Arrival, AOA) así como el tiempo de diferencia de llegada entre

dos sensores (Time Difference of Arrival, TDOA). Tanto el AOA como TDOA se usa en el algoritmo de Ubisense para calcular la posición. Así que con solo dos sensores recibiendo señales de los Tags se puede determinar una localización robusta. Aunque cuantos más sensores estén recibiendo información de los Tags, mayor será la robustez.

Además, todos los sensores están conectados entre sí mediante un cable (Ethenet) de sincronización o **Timing**. El sensor maestro envía una señal al resto de sensores y se crea un árbol de sincronización de la red que Ubisense utiliza para dar mayor robustez al sistema. Por último, todos los sensores están conectados mediante cable Ethernet y a través de un switch a un servidor. El switch sirve para conectar los sensores con el servidor para que se puedan enviar los segmentos a través de la capa de datos. En el servidor se recibe la información de los sensores y se almacena. También en el servidor están instaladas las aplicaciones software de Ubisense que permiten configurar y gestionar el sistema cuando está en funcionamiento. Estas aplicaciones se comentan en el punto 3, Configuración. En la Figura 56 se muestra un esquema de cómo se calcula la posición.

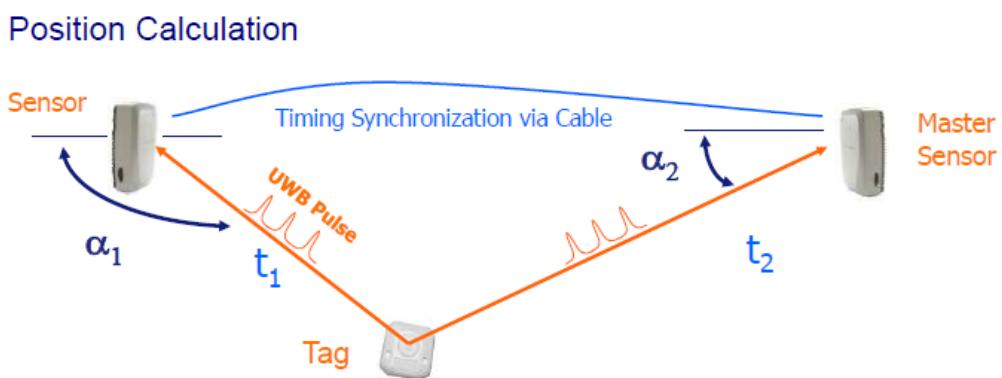


Figura 56 Cálculo de la posición (imagen proveniente de la documentación de Ubisense)

Como se ha comentado los sensores están conectados entre sí y con el servidor, de esta forma se forma una red interna. El protocolo de red que se utiliza para el envío de paquetes en el sistema es UDP, que permite el envío de datagramas a través de la red sin que se haya establecido previamente una conexión.

También es muy importante la forma del espacio, ya que la existencia de paredes que impidan la visión de un sensor puede hacer que el rendimiento del sistema sea menor, aunque si un punto está en la línea de visión de al menos dos sensores esto no sucederá. En la Figura 57 se muestra un ejemplo en el que las paredes obstaculizan la visión de algunos sensores.

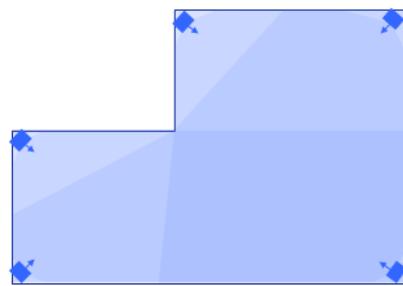


Figura 57 Espacio con paredes obstruyendo la visión de algún sensor (imagen proveniente de la documentación de Ubisense)

10. Anexo B

Los mensajes OSC constan de un patrón de dirección, una etiqueta de tipo, los argumentos y una etiqueta temporal que es opcional. El patrón de dirección forma un espacio de nombres jerárquico, como una ruta del sistema de archivos de Linux o una URL. La etiqueta de tipo es una cadena de caracteres compacta que representa los tipos de los argumentos. Los argumentos se representan en forma binaria con alineación de 3 bytes. Los principales tipos soportados son:

- Enteros con signo de 32 bits en complemento a dos.
- Números de 32 bits en IEEE coma flotante.
- **Arrays** de datos codificados de 8 bits.
- **Blobs** de tamaño arbitrario (audio o video).

11. Bibliografía

- [MCB09] Marco, J., Cerezo, E., Baldassarri, S., Mazzone, E., & Read, J. C. (2009, September). Bringing tabletop technologies to kindergarten children. In Proceedings of the 23rd British HCI Group Annual Conference on People and Computers: Celebrating People and Technology (pp. 103-111). British Computer Society.
- [UBS16] Web de Ubisense <http://ubisense.net/en>
- [UBI16] Ubisense RTLS Platform <https://ubisense.net/en/products/Dimension4>
- [SKE16] Web de Sketch Up <http://www.sketchup.com/>
- [POE16] Fundamentos de PoE
<http://www.imaginart.es/televigilancia/pdf/fundamentosPoE.pdf>
- [UAR16] Uwe a. Ruttkamp, servidor DHCP para Windows
<http://www.dhcpserver.de/cms/>
- [INI16] Extensión INI
[https://es.wikipedia.org/wiki/INI_\(extensi%C3%B3n_de_archivo\)](https://es.wikipedia.org/wiki/INI_(extensi%C3%B3n_de_archivo))
- [TIA56] Estándares cableado comercial para telecomunicaciones
<https://es.wikipedia.org/wiki/TIA-568B>
- [OSC16] Protocolo Open Sound Control <http://opensoundcontrol.org/>
- [UWB16] Ultra-wideband <https://en.wikipedia.org/wiki/Ultra-wideband>