



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Gestión de Emergencias del Patrimonio Cultural en la
Era Digital. Diseño (e implementación) de un Sistema
IoT para la auto-identificación y localización en
interiores de Obras de Arte

Trabajo Fin de Grado
Grado en Ingeniería Informática

Autor: Joan Vanaclocha Castro

Tutor: Joan Fons Cors

2016-2017

Gestión de Emergencias del Patrimonio Cultural en la Era Digital. Diseño (e implementación) de un Sistema IoT para la auto-identificación y localización en interiores de Obras de Arte



Resumen

La gestión adecuada de Emergencias ante fenómenos naturales (inundaciones, incendios, etc) permite salvar obras de incalculable valor y reducir el impacto de las catástrofes naturales, tal y como se demostró en el Louvre en la reciente crecida del río Sena.

En este proyecto se pretende implementar un sistema de soporte a la Gestión de Emergencia en el ámbito del Patrimonio Cultural. El proyecto, desarrollado en colaboración con el Institut Valencià de Conservació i Restauració de Béns Culturals (IVC+R), tiene como objetivo diseñar, dimensionar y prototipar un sistema que permita conocer en cada momento, qué obras de arte hay dentro del museo, y dónde están ubicadas. Para ello, se deberá concebir una solución tecnológica que permita responder a dos necesidades: qué hay que añadir a una obra de arte para identificarla, y qué hay que añadirle para ubicarla en interiores.

Para ambos casos, se analizarán soluciones basadas en el marcado activo (balizas Bt, Wifi, GPS, etc.), el pasivo (etiquetas NFC/RFID), así como el marcado ‘no tecnológico’ (uso de códigos de barras, fiducials, códigos QR, etc.). Se tendrá en cuenta las necesidades concretas sobre el tipo de seguimiento adecuado (on-demand o modo ‘check-ins’) para las diferentes actuaciones (exposición, restauración, traslado, etc.) que afecten a las obras del museo.

Palabras clave: Planes Emergencia, Patrimonio Cultural, IoT, auto-identificación, localización en interiores

Abstract

Appropriate emergency management before natural phenomena (floods, fires, etc) allows to save incalculable value artworks and reduce the impact of natural disasters, as was shown in Louvre museum in the recent flooding of the river Seine.

This Project aims to implement a support system for emergency management in the field of Cultural heritage. The project developed, in collaboration with the Institut Valencià de Conservació i Restauració de Béns Culturals (IVC+R), aims to design, sizing and prototyping a system that allows to know in every moment, which artworks are in the Museum, and where they are located. Therefore, it must be conceived a technological solution that allows to respond to two needs: what to add to an artwork to identify it, and what to add for indoor positioning.

In both cases, solutions base on active taggin (beacons BT, Wifi, GPS, etc.), liabilities (NFC/RFID tags) as well as “non-technological” markup (use of bars codes, fiducials, QR codes, etc.) will be analysed. Be taken into account the specific requirements on the type of adequate follow-up (on-demand or mode “check-ins”) for different actions (exhibition, restoration, transfer, etc.) relating to the artworks of the Museum.

Keywords: Emergency plans, cultural heritage, IoT, self-identification and location indoor plans.



Gestión de Emergencias del Patrimonio Cultural en la Era Digital. Diseño (e implementación) de un Sistema IoT para la auto-identificación y localización en interiores de Obras de Arte



Tabla de contenidos

1.	Introducción.....	9
1.1.	Motivación.....	9
1.2.	Objetivos.....	9
1.3.	Plan de trabajo	9
1.3.1.	Diagrama de Gantt.....	10
2.	Contexto tecnológico.....	11
2.1.	Tecnologías existentes en la actualidad.....	11
2.1.1.	GPS, Glonass, Beidu, Galileo.....	11
2.1.2.	GSM	12
2.1.3.	Infrarrojos	12
2.1.4.	Bluetooth de baja energía	12
2.1.5.	Wi-Fi.....	13
2.1.6.	Wi-Max.....	15
2.1.7.	Zigbee	15
2.1.8.	UWB.....	16
2.1.9.	Ultrasound	17
2.1.10.	RFID	17
2.1.11.	Marcado no tecnológico	18
2.2.	Métodos de cálculo para el posicionamiento.....	19
2.2.1.	Método de división del espacio	19
2.2.2.	Método de triangulación de magnitudes.....	19
2.3.	Herramientas utilizadas	20
2.3.1.	REST	20
2.3.2.	MQTT.....	22
2.4.	Proyectos actuales.....	22
2.4.1.	Situm.....	22
2.4.2.	Pozyx	23
2.4.3.	indoo.rs	24
2.4.4.	Accuware	24
2.4.5.	Meridian apps	25
2.4.6.	Seeketing.....	26
2.5.	Conclusiones.....	27
3.	Planteamiento inicial y análisis del problema	28
4.	Planteamiento de la solución	31
4.1.	Modelo conceptual	31
4.2.	Casos de uso	32
5.	Diseño de la solución.....	34
5.1.	Arquitectura de microservicios.....	34
5.2.	Big Picture	35
5.3.	Comunicaciones entre servicios	37



Gestión de Emergencias del Patrimonio Cultural en la Era Digital. Diseño (e implementación) de un Sistema IoT para la auto-identificación y localización en interiores de Obras de Arte

5.4.	Base de datos	39
6.	Implementación	40
6.1.	Capa de persistencia	40
6.2.	Capa lógica	40
6.2.1.	Entidades	41
6.2.2.	AbstractFacade.....	41
6.2.3.	APIs REST	42
6.2.4.	MQTT.....	44
7.	Ejemplo de uso del escenario	46
7.1.	Caso de uso Consultar la ubicación de una obra	46
7.2.	Caso de uso consultar el historial de una obra.....	46
7.3.	Caso de uso consultar ubicaciones actuales de todas las obras	48
7.4.	Caso de uso añadir nueva ubicación de una obra	49
8.	Conclusiones.....	50
8.1.	Conclusión.....	50
8.2.	Objetivos conseguidos y objetivos no conseguidos	50
8.3.	Trabajos futuros.....	50
9.	Bibliografía.....	51



Tabla de ilustraciones

Ilustración 1: Red de satélites GPS.....	11
Ilustración 2: Tipología estrella	13
Ilustración 3: Ángulo de llegada (AoA)	14
Ilustración 4: Tiempo de llegada (ToA)	15
Ilustración 5. Zigbee	16
Ilustración 6: Tags RFID	18
Ilustración 7: Marcador no tecnológico – código de barras.....	18
Ilustración 8: Marcador no tecnológico – Fiducials	19
Ilustración 9: Marcador no tecnológico – Códigos QR	19
Ilustración 10. Servicio REST	21
Ilustración 11: Captura de pantalla Situm.....	23
Ilustración 12: Captura de pantalla indoo.rs	24
Ilustración 13: Captura de pantalla Accuware	25
Ilustración 14: Captura de pantalla Meridian apps	25
Ilustración 15: Nodo Seeketing.....	26
Ilustración 16: Subida Río Sena en las inmediaciones del Louvre.....	28
Ilustración 16: Beacon instalado en un museo.....	30
Ilustración 17: Modelo conceptual	31
Ilustración 18: Casos de uso	32
Ilustración 19: Big Picture	35
Ilustración 20: Diagrama de la base de datos	39
Ilustración 21: Tablas en PgAdmin 4	40
Ilustración 22: Entidad ubicación	41
Ilustración 23: Métodos AbstractFacade	42
Ilustración 24: API ubicaciones	43
Ilustración 25: Métodos añadidos a AbstractFacade	44
Ilustración 26: Método de publicación mediante MQTT	44
Ilustración 27: Método consulta última ubicación.....	46
Ilustración 28: Ubicación actual devuelta por la API.....	46
Ilustración 29:Método de consulta del historial.....	47

Gestión de Emergencias del Patrimonio Cultural en la Era Digital. Diseño (e implementación) de un Sistema IoT para la auto-identificación y localización en interiores de Obras de Arte

Ilustración 30: Historial de ubicaciones.....	47
Ilustración 31: Método consulta ubicación todas las obras	48
Ilustración 32: Ubicación actual de todas las obras	48
Ilustración 33: Método añadir nueva ubicación.....	49
Ilustración 34: JSON enviado con el método POST.....	49
Ilustración 35: Mensaje enviado a los suscriptores de MQTT	49

1. Introducción

1.1. Motivación

Este proyecto surge como respuesta al problema al que se enfrentan museos hoy en día al no tener un sistema de emergencias eficiente que seguir en caso de emergencia.

En los museos hay obras de incalculable valor y un desastre mal gestionado podría significar una pérdida irreparable y de incalculable valor.

Sólo los museos de mayor envergadura poseen la tecnología y los planes necesarios para tratar cualquier tipo de emergencia y aun así, poseen algunas carencias.

El Institut Valencià de Conservació i Restauració de Béns Culturals (IVC+R) se pone en contacto con un equipo de investigación de la UPV para pedirles una solución a este problema.

Problemas tales como distintos modos de fijación de las obras junto con la carencia de una base de datos que aúne dicha información, es uno de los problemas a los que se enfrentan los museos cuando surge una situación de emergencia.

Por otro lado, las obras van cambiando de salas, pasan por exposiciones temporales, almacenes, por lo que no tener un registro en tiempo real de su ubicación pasa a ser también una variable decisiva.

1.2. Objetivos

Los objetivos principales del presente proyecto se pueden enumerar cómo:

- ❖ Realizar un estudio de las tecnologías de posicionamiento en interiores
- ❖ Realizar un microservicio de ubicaciones que ofrezca una API REST
- ❖ Enviar las actualizaciones a una cola de mensajería MQTT

1.3. Plan de trabajo

A continuación, se observa un Diagrama de Gantt, el cual expone el tiempo de dedicación previsto para cada una de las diferentes tareas o actividades a lo largo del desarrollo del presente proyecto.

Gestión de Emergencias del Patrimonio Cultural en la Era Digital. Diseño (e implementación) de un Sistema IoT para la auto-identificación y localización en interiores de Obras de Arte

1.3.1. Diagrama de Gantt



2. Contexto tecnológico

En este capítulo se exponen las tecnologías existentes en la actualidad para resolver el problema de la localización y posicionamiento en interiores, los métodos de cálculo de posición más utilizados, así como proyectos o aplicaciones desarrollados en el presente.

2.1. Tecnologías existentes en la actualidad

2.1.1. GPS, Glonass, Beidu, Galileo

La localización por GPS es la tecnología por excelencia para dicha tarea en nuestros días. Todos los sistemas de localización por satélite utilizan el mismo principio de funcionamiento, ya sea GPS (EE.UU.), GLONASS (Russia), BEIDU (China) o GALILEO (Europa). Los receptores satélite reciben soporte de una constelación de hasta 24 satélites, que orbitan por todo el globo terrestre enviando sus señales.

Un receptor recibe la señal de al menos 4 satélites (cuanto mayor sea el número de satélites encontrados, mejor será la estimación de la posición) con su posición y el tiempo de envío. Estos datos permiten llevar a cabo la triangulación de la posición absoluta (latitud, longitud y altura), así como la velocidad. Corrigiendo los retardos de la propagación de la señal y utilizando más de 4 satélites, obtenemos una precisión de hasta 10 metros, lo que lo convierte en un sistema idóneo para localizar a nivel global.

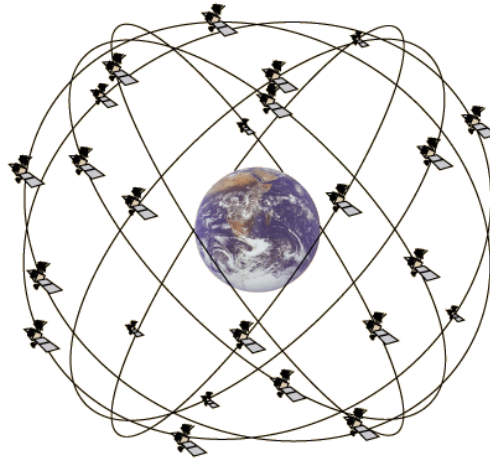


Ilustración 1: Red de satélites GPS

Sin embargo, no es un sistema tan eficiente en cuanto a posicionamiento en interiores debido a las siguientes causas:

Gestión de Emergencias del Patrimonio Cultural en la Era Digital. Diseño (e implementación) de un Sistema IoT para la auto-identificación y localización en interiores de Obras de Arte

- ❖ Se necesita línea de visión directa (LoS). Hay veces en las que no se tiene esta línea de visión directa con suficientes satélites, debido a la posición del receptor, para permitir una localización correcta.
- ❖ Las señales de los satélites viajan muchos kilómetros y son bastante tenues, por lo que un receptor en el interior de un edificio tiene complicado recibir suficientes señales con la potencia adecuada para que le sirvan para localizarse.
- ❖ El sistema de referencia que ofrecen estos sistemas es de posicionamiento absoluto, mientras que en localización en interiores se necesitan referencias lógicas y adaptadas al entorno, como por ejemplo la recepción del museo.

2.1.2. GSM

Existen compañías de telefonía móvil que ofrecen localización vía red móvil. Esta tecnología funciona mediante triangulación de antenas móviles. La precisión de esta tecnología la sitúa en clara desventaja respecto a otras ya que no puede dar precisiones mayores de 50 metros, lo que la convertiría en inservible para interiores.

2.1.3. Infrarrojos

La tecnología IR es muy sencilla de usar, barata, compacta y de bajo consumo, características que la hacen muy atractiva en el desarrollo de sistemas de localización inalámbrica en interiores. Sin embargo, cuenta con ciertas deficiencias que hacen descartarla para la localización de interiores.

Tiene un alcance de unos 5 metros aproximadamente y sus enlaces requieren LoS (Línea de visión directa) entre sí. Además, la presencia de luz fluorescente y luz solar, interfieren en la propia señal IR.

2.1.4. Bluetooth de baja energía

Bluetooth de baja energía, es una tecnología de red inalámbrica que permite la comunicación de corto alcance entre dispositivos. Es la nueva especificación de la tecnología bluetooth para garantizar el bajo consumo de energía. A pesar de utilizar la misma marca y una tecnología similar, debe considerarse como una tecnología diferente, con distintas aplicaciones y objetivos.

Está diseñado para la transmisión de pequeñas cantidades de datos y con un consumo de energía muy bajo. Los dispositivos solo están en modo conectado cuando se les pide para una transmisión de datos. Debido a esto, el consumo de energía se ve reducido. Gracias a su bajo consumo y su rapidez de transmisión de datos, es usado en nuevas aplicaciones relacionadas con el cuidado de la salud o el fitness entre otras.

La principal ventaja respecto a la versión anterior de Bluetooth es la reducción considerable del consumo de energía y los costes. Además, es capaz de mantener un rango de comunicación similar a su predecesor. Las principales ventajas son:

- ❖ Transferencia de datos. Se logra una velocidad de transferencia de unos 1 Mbps.
- ❖ Salto de frecuencia. Utiliza un salto de frecuencia como sus versiones anteriores para minimizar las interferencias posibles de otras tecnologías.
- ❖ Latencia. Puede soportar una configuración de conexión y transferencias de unos 3ms, por lo tanto, puede crear la conexión y transferir los datos casi en tiempo real.
- ❖ Rango. Permite un rango de aproximadamente unos 100 metros. Lo que lo convierte en una gran ventaja respecto a sus competidores.
- ❖ Seguridad. Proporciona una fuerte encriptación y autenticación de sus paquetes gracias al sistema de alta seguridad con un cifrado AES-128 que utiliza.
- ❖ Tipología. Se utiliza una dirección de acceso de 32 bits en cada paquete, permitiendo que millones de dispositivos estén conectados. La tecnología esta optimizada para conexiones one-to-one permitiendo al mismo tiempo una conexión de muchas conexiones utilizando una tipología de estrella.

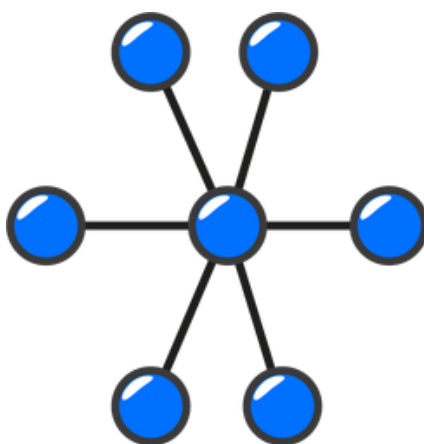


Ilustración 2: Tipología estrella

2.1.5. Wi-Fi

El protocolo 802.11 (WiFi) establece un sistema de comunicación centralizado. Existe una red con uno o varios puntos de acceso (APs: Acces Points) a los que se conectan los clientes. Los puntos de acceso envían una señal periódicamente.

Esta emisión y recepción de señal permite la localización en interiores usando diferentes técnicas:

- ❖ **Indicador de fuerza de señal recibida (RSSI)**

Esta técnica se basa en el uso de varios puntos de acceso (con un mínimo de tres) para realizar la localización.

Se divide en dos métodos: fingerprinting y triangulación.

-El fingerprinting (huella digital) consiste en generar un mapa de intensidades RSSI del área de interés. Cuando se requiere la localización, se lee el RSSI de todos los puntos

Gestión de Emergencias del Patrimonio Cultural en la Era Digital. Diseño (e implementación) de un Sistema IoT para la auto-identificación y localización en interiores de Obras de Arte

de acceso visibles por el cliente y mediante un algoritmo se encuentra el punto que mejor coincide con el mapa trazado.

-La triangulación (modelo de propagación) consiste en conocer las coordenadas de los puntos de acceso. Si la distancia r desde el punto de acceso hasta el objeto puede ser medida (mediante la atenuación de señal), se puede dibujar un círculo con radio r , la intersección de estos círculos resulta en un punto el cual es la posición del objeto.

❖ **Ángulo de llegada (AoA)**

Este método se basa en triangulación mediante el uso de antenas sensitivas a la dirección en los receptores de manera que se pueda obtener la dirección hacia el emisor.

El ángulo de llegada se calcula trazando una línea entre el receptor y emisor y otra desde el receptor hacia una dirección conocida como pueden ser los puntos cardinales (Ilustración 3). Usando varios transmisores en posiciones conocidas se puede determinar la posición del receptor mediante triangulación.

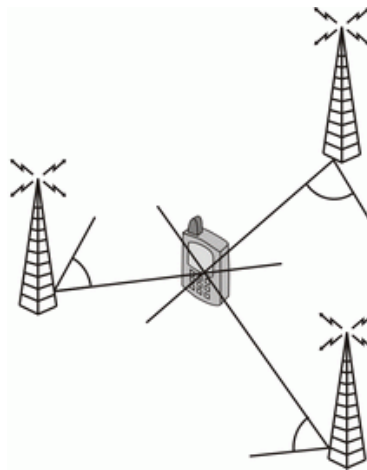


Ilustración 3: Ángulo de llegada (AoA)

❖ **Tiempo de llegada (TOA)**

Este método se basa en triangulación y en el tiempo de propagación de la señal entre el transmisor y los receptores. El tiempo de propagación es el que necesita la señal para viajar desde el transmisor al receptor y puede calcularse como $(t. \text{ transmisor} - t. \text{ receptor})$.

A partir de este tiempo y sabiendo que la velocidad de propagación es aproximadamente la de la luz, se obtiene la distancia entre transmisor y receptor. Esta técnica tiene dos problemas básicos, el primero de ellos es que no haya una trayectoria directa entre el transmisor y el receptor, la segunda es que los relojes de ambos dispositivos deben estar sincronizados para obtener una distancia correcta, lo cual afecta en el coste de los dispositivos.

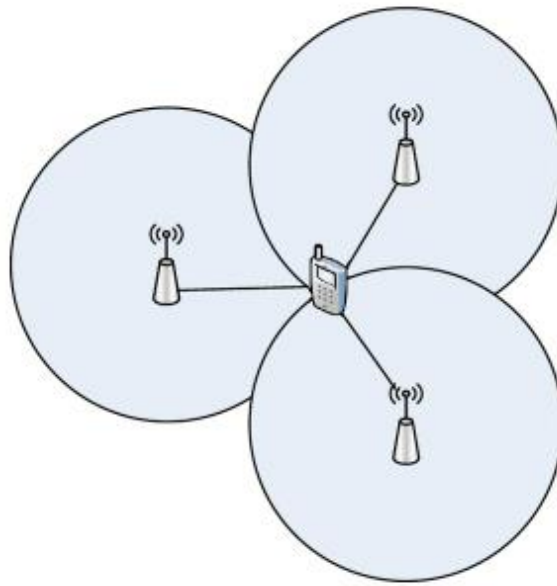


Ilustración 4: Tiempo de llegada (ToA)

Se puede encontrar una variante de este método llamada diferencia del tiempo de llegada (TDOA). Éste se basa en la estimación de la diferencia entre los tiempos de llegada de la señal emitida en múltiples receptores.

❖ **Tiempo de vuelo (TOF)**

Este método basado en triangulación utiliza el tiempo transcurrido en una transmisión entre emisor y receptor, consiguiendo una velocidad de propagación estimada de la señal.

Receptores con relojes de alta exactitud transmiten señales con valores de tiempo de salida conocidos hacia los transmisores con este tipo de relojes. El tiempo de salida se compara con el tiempo de llegada y usando una estimación de la velocidad de propagación se puede determinar la distancia entre los dispositivos. Este método sufre de los mismos problemas que DOA.

2.1.6. Wi-Max

La tecnología Wi-Max sigue el estándar 802.16. Está pensada para la intercomunicación de áreas muy extensas, de hasta 48 kilómetros de radio, y puede llegar a transmitir hasta 70Mbps. Debido al gran alcance, no se puede pensar en establecer un sistema de localización en interiores usando esta tecnología.

2.1.7. Zigbee

ZigBee es un conjunto de protocolos de alto nivel de comunicación. Utiliza una tecnología basada en el estándar IEEE 802.15.4.

Está centrado en la comunicación entre dispositivos con una tasa de datos baja con el fin de tener el menor consumo energético posible. Esta tecnología está muy centrada en

Gestión de Emergencias del Patrimonio Cultural en la Era Digital. Diseño (e implementación) de un Sistema IoT para la auto-identificación y localización en interiores de Obras de Arte

la domotica, es decir, dotar una casa de inteligencia. Cuenta con una serie de características que lo diferencian de otras tecnologías:

Su bajo consumo.

- ❖ Su tipología de red en malla. Un nodo esta conectado a su vez a más nodos de la misma red, lo que asegura que siempre habrá un camino a seguir en caso de caída de alguno.
- ❖ Su fácil integración (se pueden fabricar nodos con muy poca electrónica).
- ❖ Máximo de 65535 nodos distribuidos en subredes de 255 nodos.

Las categorías de dispositivos que nos encontramos son:

- ❖ El Coordinador ZigBee. Siempre tiene que existir en la red. Es el nodo más completo y se encarga de controlar toda la red y los caminos para su comunicación.
- ❖ Router Zigbee. Interconecta los nodos para poder ejecutar el código de usuario, es decir, ofrece un nivel de aplicación dentro de la torre de protocolos.
- ❖ Dispositivo final ZigBee. Sólo recibe información y se comunica unicamente con el nodo padre. La ventaja de este dispositivo es que puede permanecer dormido y “despertarse” en ciertos momentos para alargar la duración de la batería.

ZigBee es una tecnología interesante para la comunicación de dispositivos que necesiten un consumo energético reducido. Tiene el apoyo de muchos fabricantes importantes como Samsung, LG, Logitech o Bosch, una alianza que quiere imponer esta tecnología para conectar todos los dispositivos.

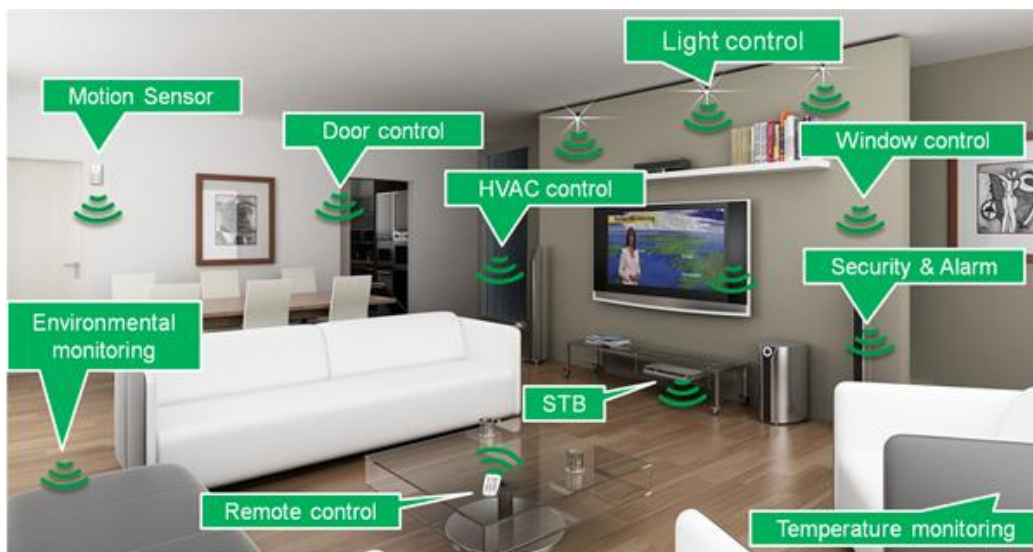


Ilustración 5. Zigbee

2.1.8. UWB

La tecnología UWB parece una buena candidata para la localización en interiores. Sus principales ventajas son la precisión, que puede alcanzar hasta un metro, y su robustez ante obstáculos y cambios del entornos (puertas, paredes, presencia o movimiento de personas, etc.).

Esta tecnología se encuentra en estos momentos en estudio de aplicabilidad y de explotación.

2.1.9. Ultrasound

Son sistemas basados en la utilización de nodos emisores y receptores de ultrasonidos, que utilizan principalmente el tiempo de vuelo entre ondas ultrasónicas como métrica de localización para determinar la localización del usuario.

Los ultrasonidos se han convertido en una alternativa dentro del mercado RTLS debido a que proporcionan una alta precisión,. Pero, debido a su baja cobertura y a la necesidad de desarrollar una infraestructura propia con un elevado número de nodos fijos en el entorno, puesto que es necesario garantizar una línea de visión directa entre los dispositivos, sus costes resultan muy elevados.

2.1.10. RFID

El propósito de la tecnología de Identificación por radiofrecuencia (RFID), es identificar mediante un lector, sin contacto y a distancia, una tarjeta o etiqueta (tag).

El lector realiza peticiones o preguntas por radiofrecuencia al chip que integra la etiqueta RFID, las cuales emiten una respuesta al recibir las señales, lo que permite una identificación con gran seguridad y precisión en tiempo real.

Los tags (etiquetas) contienen un pequeño chip y una antena impresa o bobinada para la comunicación con el lector. El chip tiene grabado un ID único que lo identifica, y en algunos casos dispone de una pequeña memoria que los lectores son capaces de leer y escribir.

Hay tres tipos de etiquetas RFID, activas, semipasivas y pasivas.

- ❖ Las etiquetas pasivas no necesitan alimentación interna, toman la energía necesaria de la propia emisión de las antenas del lector y sólo se activan si están en el campo de cobertura del lector.
- ❖ Las etiquetas activas utilizan alimentación de una pequeña batería, por lo que se pueden comunicar con el lector a una distancia mucho mayor y con un procesamiento de datos superior.
- ❖ Las etiquetas semipasivas poseen una fuente de alimentación propia, aunque se utiliza principalmente para alimentar el microchip y no para transmitir la señal.

Los tags RFID son de multitud de formas, tamaños y materiales, dependiendo de los entornos en los que se van a utilizar.



Gestión de Emergencias del Patrimonio Cultural en la Era Digital. Diseño (e implementación) de un Sistema IoT para la auto-identificación y localización en interiores de Obras de Arte

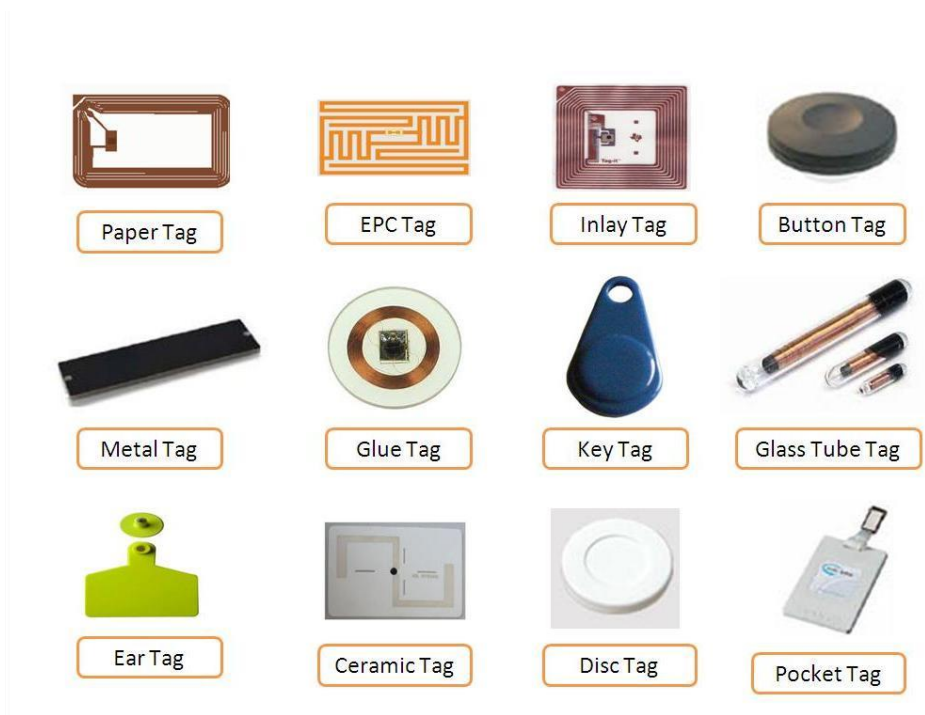


Ilustración 6: Tags RFID

En la actualidad, esta tecnología representa numerosos usos: control de inventarios, movimiento de mercancías, identificación de ganado y mascotas, peajes, control de procesos industriales, control de acceso a edificios, pasaportes digitales, e incluso, para localización en interiores.

2.1.11. Marcado no tecnológico

Otra posibilidad, sería utilizar sistemas de marcado no tecnológico. Este método utiliza imágenes únicas que deben ser escaneadas con un lector, ya sea un dispositivo específico, un smartphone o tablet, o incluso una cámara de vigilancia.

Los sistemas más utilizados son:

- ❖ Códigos de barras



Ilustración 7: Marcador no tecnológico – código de barras

- ❖ Fiducials

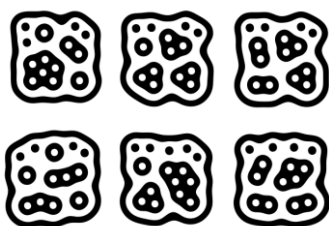


Ilustración 8: Marcador no tecnológico – Fiducials

- ❖ Códigos QR



Ilustración 9: Marcador no tecnológico – Códigos QR

2.2. Métodos de cálculo para el posicionamiento

En lo referente al cálculo de la posición de un dispositivo en interiores, independientemente de la tecnología y de la arquitectura utilizada, existen dos principales métodos utilizados en la actualidad:

2.2.1. Método de división del espacio

Este método está basado en la división del espacio en celdas de igual tamaño a las cuales se les asigna un punto de acceso. Cada punto de acceso posee la misma cobertura, por tanto, cuando una baliza emite una señal, ésta será localizada en el área de responsabilidad de un determinado punto de acceso.

2.2.2. Método de triangulación de magnitudes

Este método está basado en el cálculo matemático de la geometría de triángulos. A partir de magnitudes conocidas entre emisor-receptor y conocida la localización de dos puntos de referencia, se puede estimar la posición de un dispositivo en el plano. La localización precisa y sin errores se consigue con tres puntos de referencia conocidos. Las magnitudes medidas y analizadas por los receptores fijos son:

- ❖ Ángulo de llegada de la señal del emisor (AoA).
- ❖ Tiempo de llegada de la señal del emisor (ToA).
- ❖ Nivel de señal recibido (RSS).
- ❖ Tiempo de vuelo (ToF)

Gestión de Emergencias del Patrimonio Cultural en la Era Digital. Diseño (e implementación) de un Sistema IoT para la auto-identificación y localización en interiores de Obras de Arte

2.3. Herramientas utilizadas

En esta sección, se van a tratar las herramientas más relevantes que se utilizarán en este proyecto. Herramientas, entornos, librerías, etc que sean triviales como Java, NetBeans IDE, HTTP, etc quedarán excluidas de esta sección.

2.3.1. REST

La Transferencia de Estado Representacional (en inglés Representational State Transfer) o REST, es un estilo de arquitectura que abstrae los elementos de dicha arquitectura dentro de un sistema hipermedia distribuido.

En otras palabras, REST es un conjunto de principios que define la interacción entre distintos componentes, es decir, las reglas que dichos componentes deben seguir. El protocolo más usado que cumple esta definición es el protocolo HTTP.

Esto quiere decir, por extensión, que toda aplicación web bajo el protocolo HTTP es a su vez una aplicación REST. Sin embargo, eso no implica en absoluto que todas las aplicaciones web sean servicios web RESTful, como veremos más abajo. Existen, además, otros métodos para implementar servicios web, como RPC, SOAP o WSDL. No obstante, RESTful es mucho más fácil de entender e implementar, por lo que se suele recomendar su uso.

REST define una serie de reglas que toda aplicación debe cumplir para llamarse REST. Como podemos comprobar, estas reglas ya nos vienen dadas si vamos a utilizar el protocolo HTTP.

- ❖ Arquitectura cliente-servidor. Consiste en una separación clara de los dos agentes básicos que intervienen en un intercambio de información. El cliente y el servidor deben ser independientes entre sí, lo que permite una flexibilidad muy alta.
- ❖ Stateless. Significa que nuestro servidor no tiene porqué almacenar datos del cliente para mantener un estado del mismo.
- ❖ Cacheable. El servidor debe definir algún modo de cachear dichas peticiones para aumentar el rendimiento, escalabilidad, etc.
- ❖ Sistema por capas. El sistema no debe forzar al cliente a que sepa por qué capas se tramita la información, lo que permite que el cliente sea independiente a las otras capas.
- ❖ Interfaz uniforme. Simplifica el protocolo y aumenta la escalabilidad y el rendimiento del sistema. La interfaz de comunicación no depende del servidor, ni mucho menos del cliente, lo que garantiza que no importa quién haga las peticiones ni quien las reciba, siempre que los dos cumplan una interfaz definida de antemano.

Por otra parte, para que una aplicación sea REST al 100%, tendrá que implementar 4 principios básicos:

- ❖ Identificación de recursos. Toda aplicación REST debe poder identificar sus recursos de manera uniforme.

- ❖ Recursos y representaciones. Visto que todo recurso debe tener una identificación, REST define también la manera en que podemos interactuar con la representación del mismo, ya sea para editarlo o borrarlo directamente del servidor.
- ❖ Mensajes auto descriptivos. Cuando hacemos peticiones a un servidor, este deberá devolver una respuesta que nos permita entender sin lugar a duda cual ha sido el resultado de la operación, así como si dicha opción es cacheable, si ha habido algún error, etc.
- ❖ HATEOAS. Por último, y algo que la mayoría de servicios web no cumplen, es la necesidad de incluir en las respuestas del servidor toda aquella información que necesita el cliente para seguir operando con este servicio web. En otras palabras, el cliente no tiene por qué saber que cuando obtenemos, por ejemplo, un objeto cualquiera, tenemos además las opciones de modificarlo o eliminarlo. El servidor debe enlazar a estas operaciones en la respuesta dicha petición. De esta manera, lo único que necesita saber el cliente de una petición REST, es el punto de entrada (endpoint).

Ahora que se ha hablado sobre las reglas y principios básicos de REST, es posible dar una definición más precisa de los servicios web RESTful, lo cuál dejará claro cómo se tiene que implementar. Un servicio web RESTful contiene lo siguiente:

- ❖ URI del recurso. Por ejemplo: `http://api.servicio.com/museo/Obra/1` (esto nos daría acceso al recurso “Obra” con el ID “1”).
- ❖ El tipo de la representación de dicho recurso. Por ejemplo, podemos devolver en nuestra cabecera “Content-type: application/json”, por lo que el cliente sabrá que el contenido de la respuesta es una cadena en formato JSON, y podrá procesarla como prefiera. El tipo es arbitrario, siendo los más comunes JSON, XML y TXT.
- ❖ Operaciones soportadas: HTTP define varios tipos de operaciones (verbos), que pueden ser GET, PUT, POST, DELETE, PURGE, entre otros. Es importante saber para qué están pensados cada verbo, de modo que sean utilizados correctamente por los clientes.
- ❖ Hipervínculos: por último, nuestra respuesta puede incluir hipervínculos hacia otras acciones que podamos realizar sobre los recursos. Normalmente se incluyen en el mismo contenido de la respuesta, así si por ejemplo, nuestra respuesta es un objeto en JSON, podemos añadir una propiedad más con los hipervínculos a las acciones que admite el objeto.

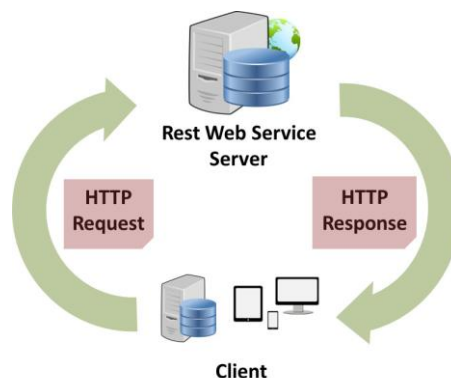


Ilustración 10. Servicio REST

Gestión de Emergencias del Patrimonio Cultural en la Era Digital. Diseño (e implementación) de un Sistema IoT para la auto-identificación y localización en interiores de Obras de Arte

2.3.2. MQTT

Message Queue Telemetry Transport (MQTT) es un protocolo de conectividad abierto Machine-to-Machine (M2M) que permite enviar datos estilo telemetría como mensajes a través de redes de alta latencia o con restricciones.

Está diseñado para ser una mensajería Publish-Suscribe muy ligera, es muy útil para conexiones donde es importante no consumir mucho ancho de banda (se usa por ejemplo en comunicaciones entre sensores, conexiones GPRS, etc). Es también ideal para aplicaciones móviles por su envío eficiente. Se usa por ejemplo en el Facebook Messenger para iPhone y Android.

En este proyecto se utilizará la tecnología Eclipse Paho para realizar los publish y subscribe correspondientes.

El proyecto Eclipse Paho es parte de Eclipse Foundations's M2M mission para proporcionar implementaciones de alta calidad de librerías y herramientas basadas en M2M.

2.4. Proyectos actuales

La localización en interiores está siendo un ámbito de estudio y desarrollo en la actualidad debido a sus numerosas aplicaciones: localización de objetos, control y cuidado de personas con necesidades especiales, control de personal en empresas, guía de maquinaria en planta, guía en rutas de interés en museos, localización de asientos en teatros y estadios, etc. Si, además de la localización, se consigue la interacción entre el usuario y el entorno inteligente en el que se encuentra, el rango de aplicaciones de este tipo de proyectos se hace casi infinito.

En el presente apartado se exponen algunos de los proyectos desarrollados o en vía de desarrollo por diferentes organismos en la actualidad.

2.4.1. Situm

En cuanto a los proyectos actuales en España, nos encontramos con Situm, un sistema de orientación en espacios cerrados creado por la empresa Situm Technologies. Es capaz de determinar la posición de cualquier dispositivo con sensores, como smartphones o wearables, con alta precisión y consistencia. Sus principales características son las siguientes:

- ❖ Alta precisión en interiores en cualquier espacio, incorporando también detección automática de planta.
- ❖ Mínima infraestructura. La tecnología multisensorial hace uso de la infraestructura preexistente, evitando los costes de la adquisición y mantenimiento de nuevo hardware.

- ❖ Robustez. Si un punto de acceso WiFi o un beacon es retirado del entorno (o fuera de batería), el sistema continúa funcionando como si ese dispositivo nunca hubiera estado allí.
- ❖ Posición y orientación. El motor de posicionamiento es capaz de estimar tanto la posición del smartphone como la dirección en la que está caminando la persona.
- ❖ Despliegue inmediato. La herramienta de calibración sin pausa de Situm permite el mapeo de cualquier lugar en cuestión de minutos.
- ❖ Gestor de cartografía. Puedes crear o subir tus propios mapas, puntos de interés, etc.
- ❖ Gestor de rutas. Puedes configurar la navegación en los edificios, creando rutas, situando cambios nivel, etc
- ❖ Visor en tiempo real
- ❖ Gestor de informes
- ❖ Kit de desarrollo.

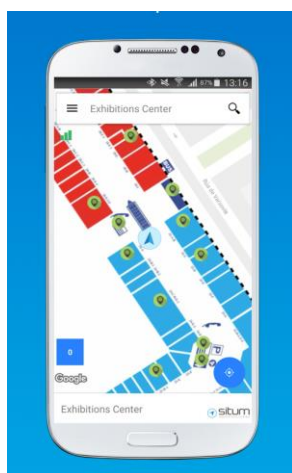


Ilustración 11: Captura de pantalla Situm

2.4.2. Pozyx

Pozyx es una tecnología basada en señales Ultra-WideBand (UWB) y implementado sobre una placa Arduino, que ha surgido recaudando fondos mediante Kickstarter.

El sistema consiste en 4 antenas colocadas alrededor del perímetro que se quiere utilizar. Cada una de ellas transmite una señal de radio ultra-wideband que puede atravesar paredes y otros obstáculos, y tiene un rango de unos 200 metros.

Hay una quinta pieza: El Pozyx tag, la cual interpreta las señales para obtener su posición en un espacio 3D. El tag también incluye un sensor de movimiento para obtener la orientación del modulo y un sensor de presión para saber si el robot se ha chocado con algo.

Pozyx ofrece una precisión de unos 10 cm porque sus antenas emiten diferentes señales con las que comparar. Además, al ser construido sobre Arduino, es muy fácil de programar e integrar.

Aunque el caso de uso más obvio es el de los robots con movimiento automático, se espera que esta tecnología crezca muy rápidamente y se le den diferentes usos en el

Gestión de Emergencias del Patrimonio Cultural en la Era Digital. Diseño (e implementación) de un Sistema IoT para la auto-identificación y localización en interiores de Obras de Arte

posicionamiento de interiores. Para ello, han puesto sus librerías Arduino y parte del firmware a disposición de la comunidad.

2.4.3. indoo.rs

Indoo.rs es una solución comercial de sistema de posicionamiento en interiores basado en Bluetooth de baja energía (BLE). Sus principales características son:

- ❖ Posicionamiento preciso.
- ❖ Routing. Te permite calcular una ruta desde un punto inicial a uno final (Figura X).
- ❖ Mapas offline. No es necesaria conexión a internet para cargar el mapa con el que se va a trabajar.
- ❖ Solución unificada. Permite trabajar con dispositivos iOS y Android.
- ❖ Uso reducido de batería

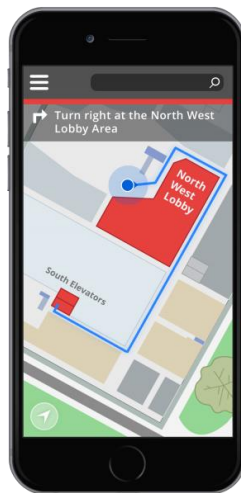


Ilustración 12: Captura de pantalla indoo.rs

2.4.4. Accuware

Se trata de una tecnología de posicionamiento de personas u objetos en espacios cerrados. Para ello, utiliza como referencia puntos de acceso inalámbricos basados en WiFi y utiliza la técnica de triangulación entre dichos puntos de acceso para calcular la posición. Sus principales características son:

- ❖ Posicionamiento mediante la técnica de triangulación. Accuware calcula el posicionamiento de los objetos o personas usando la triangulación de las señales WiFi.
- ❖ Colaboración abierta distribuida (Crowdsourcing). Se basa en una comunidad de más de un millón de usuarios para recopilar la información sobre los puntos de acceso WiFi y así tener una base de datos dinámica y actualizada y una cobertura global.
- ❖ Base de datos global. Como hemos comentado, Accuware posee una base de datos con las localizaciones de todos los puntos de acceso WiFi registrados.
- ❖ Gestor de rutas.

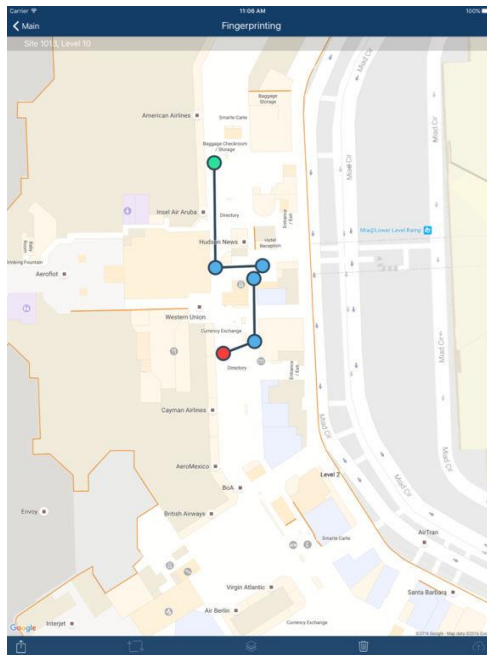


Ilustración 13: Captura de pantalla Accuware

2.4.5. Meridian apps

Este proyecto consiste en un conjunto de soluciones utilizando la tecnología iBeacon y el Bluetooth de baja energía.

Ofrece diferentes características tales como:

- ❖ Sistema de navegación en interiores. Permite cargar un mapa en la app y ofrece un guiado de punto a punto.
- ❖ Posicionamiento en interiores. Ofrece al usuario su posición real en interiores utilizando la técnica de trilateración entre varias balizas instaladas.
- ❖ Notificaciones de publicidad. Al pasar por un comercio, ofrece al usuario notificaciones sobre descuentos o publicidad

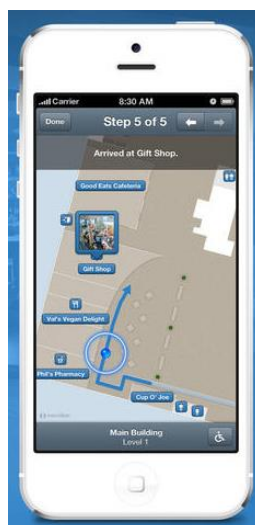


Ilustración 14: Captura de pantalla Meridian apps

Gestión de Emergencias del Patrimonio Cultural en la Era Digital. Diseño (e implementación) de un Sistema IoT para la auto-identificación y localización en interiores de Obras de Arte

2.4.6. Seeketing

Estos llamados Nodos Seeketing no emiten señales, sino que las reciben. Poseen una antena y escuchan las señales que emiten todos los móviles y tablets continuamente en las bandas de 2.4-5 Ghz y 900-1800 Mhz, es decir señales de sus tarjetas Wifi, Bluetooth y Red Móvil (2G/3G/4G).



Ilustración 15: Nodo Seeketing

Los Nodos detectan los móviles de forma anónima y sin necesidad de que exista una APP instalada en ellos, y pueden enviar mensajes en tiempo real: vía email, sms, whatsapp (si no tienen APP) o notificaciones push (si tienen una APP instalada)

Otras diferencias son que los nodos Seeketing requieren tener una toma de alimentación pero por el contrario, su cobertura es mucho mayor (unos 400 metros cuadrados).

Seeketing ofrece WebServices y SDK para iOS, Android y HTML5 que asignan el mismo identificador único al usuario cuando está offline (en tienda), cuando está usando la APP o navegando por la WEB fuera de tienda

En un proyecto con Nodos Seeketing se detectan entre un 65% y un 100% del público real de una tienda o centro comercial (todos los smartphone que estén emitiendo algún tipo de señal Wifi, Bluetooth o de red móvil).

Estos dispositivos ya están siendo usados en proyectos para centros comerciales, grandes y pequeñas tiendas, aeropuertos y estaciones de Tren, hospitales, Museos y Parques de atracciones, oficinas, etc.

2.5. Conclusiones

Como se ha podido comprobar, actualmente existen muchas tecnologías para resolver el problema del posicionamiento en interiores. Siendo ZigBee, BlueTooth y Wi-Fi, los candidatos más idóneos para el tema a tratar en este proyecto.

Dicho esto, como la instalación de la tecnología no atañe a este proyecto y como se verá en la solución, no afecta al desarrollo del mismo, no se ha elegido ninguna tecnología en concreto.

3. Planteamiento inicial y análisis del problema

Actualmente, solo los museos más importantes, y no en todos los casos, tienen un sistema de emergencias seguro, eficiente, bien implementado y que contemple un gran abanico de posibles emergencias a las que se pueden enfrentar. Los museos o entidades de menor envergadura como el Institut Valencià de Conservació i Restauració de Béns Culturals (IVC+R), no disponen de estos sistemas y por tanto, todas las obras de arte que contienen quedan expuestas ante cualquier emergencia no contemplada o mal gestionada por falta de recursos o conocimientos.

Si nos fijamos en el museo Louvre de París, el cual tuvo que atender una emergencia importante en junio de 2016, podemos observar que hasta los museos más importantes del mundo tienen carencias en sus sistemas de emergencia. Se pueden ver frente a situaciones que no habían contemplado y esto puede acarrear una pérdida de patrimonio irreparable, pues contienen obras de valor incalculable.

En este caso, se enfrentaron al contratiempo del desbordamiento del río Sena, al estar situado en sus inmediaciones. El 70% de los depósitos del museo (8.600 metros cuadrados) se encuentran en zona con riesgo de inundación y por lo tanto, unas 220.000 obras se encontraban en peligro. El plan de evacuación previsto, en el cual participaron medio millar de trabajadores voluntarios del museo, preveía trasladar las obras en un plazo de 72 horas, tiempo totalmente insuficiente para evacuar la totalidad de las mismas si el problema hubiera evolucionado en una situación peor. Además, el museo no disponía de 8.600 metros cuadrados de zonas no inundables para poder transferirlas, y hasta 2018 no inaugurarán un nuevo depósito dónde poder albergarlas.



Ilustración 16: Subida Río Sena en las inmediaciones del Louvre

Como se puede comprobar, si extrapolamos esta situación a un escenario dónde la emergencia necesite un tiempo de actuación menor, como un incendio, o a un museo de

dimensiones menores y por tanto con menos recursos, las consecuencias de una emergencia pueden ser fatales.

Por lo tanto, ante una situación así, el tiempo de actuación se convierte en un elemento crítico. Se necesita conocer cómo está fijada la pieza que va a ser evacuada; las herramientas que se necesitan para desmontarla; la ruta a seguir según sus medidas, el tipo de emergencias y las zonas libres de peligro; las personas necesarias para su transporte; etc.

Cuando no hay tiempo para la evacuación de la totalidad de las obras, se tiene que proceder a hacer una selección de las obras más prioritarias, puesto que no todas ellas tienen el mismo valor. Aquí se tiene que lidiar con el problema de que no todas las obras tienen una ubicación fija. Algunas están guardadas en los depósitos, otras cambian de sala, otras están en exposiciones temporales, incluso pueden haber obras de diferentes museos en alguna ocasión. Por esto, se tiene que conocer en todo momento la ubicación exacta de cada una.

Para el presente proyecto, miembros del equipo de investigación de la UPV, se reunieron en el museo con los responsables del mismo para llevar a cabo una evaluación de la situación. Se pudo comprobar que el sistema de emergencia contaba con notables carencias. Uno de los ejemplos más claros es un retablo de grandes dimensiones, el cual, si se produjera algún tipo de emergencia, no podría ser evacuado o simplemente reubicado para, en la medida de lo posible, evitar su deterioro o incluso destrucción.

Otra dificultad añadida con la que cuenta este museo, es la diversidad de formas con que las obras de arte están fijadas. No hay un estándar, pues cada obra cuenta con una forma y unas características determinadas (esculturas, retablos, etc). Por esto, en el momento de evacuar, el operario o la persona encargada de la evacuación debe saber de antemano qué herramientas va a necesitar y cómo debe proceder. De acuerdo a lo anterior, este museo necesita un sistema de emergencias eficiente para evitar cualquier tipo de deterioro que pueda ser ocasionado en las obras.

En muchos museos, tanto alrededor del mundo como en España últimamente, se han implantado tags RFID en las obras de arte para incrementar su seguridad. Esta medida está más enfocada a impedir un problema de grandes magnitudes como es el del robo de obras de arte, debido a su gran valor, y los desperfectos que puedan sufrir las obras debido al vandalismo, más que a tener controladas las obras en todo momento por si hubiera alguna emergencia.

Por otra parte, también están siendo instalados beacons bluetooth en algunos museos, con una finalidad interesante, pero totalmente diferente a la que buscamos en este proyecto. En estos casos sirve para dar más información o interactuar con la obra mediante algún dispositivo como smartphones o tablet, o para marcar un recorrido por el museo.

Gestión de Emergencias del Patrimonio Cultural en la Era Digital. Diseño (e implementación) de un Sistema IoT para la auto-identificación y localización en interiores de Obras de Arte

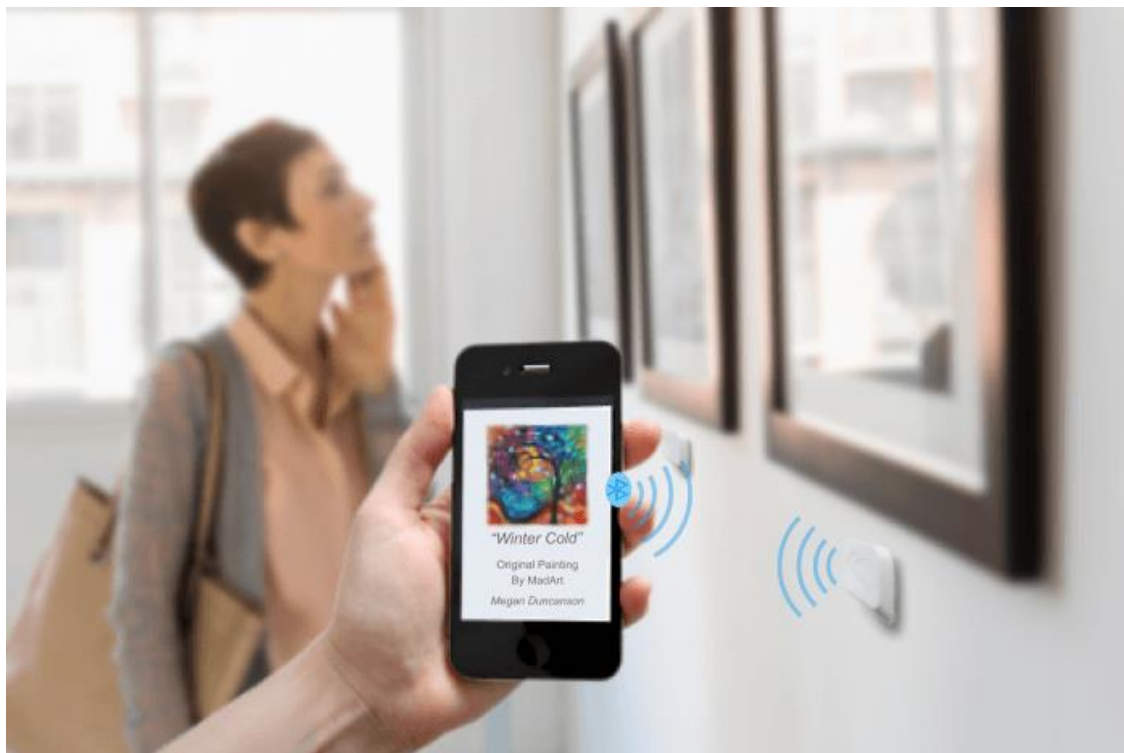


Ilustración 16: Beacon instalado en un museo

4. Planteamiento de la solución

Tras el capítulo anterior, en el que se detalla el problema, se puede empezar a definir una solución al mismo: crear un sistema de emergencias con planes definidos en los que sea contemplada cualquier posibilidad.

Para ello, es necesario conocer la información de las obras de arte, un modelo del edificio del museo que proporcione las ubicaciones lógicas donde se pueden encontrar las obras, unos planes de actuación dependiendo del tipo y la zona de emergencia, un sistema que proporcione las ubicaciones exactas (ya sea lógica o físicamente), un sistema que sabiendo el plan de emergencia aplicable dependiendo de la situación y la ubicación de las obras, proporcione el plan de emergencia y los pasos a seguir por los operarios que los recibirían mediante una aplicación móvil o de escritorio.

Para llevar a cabo esta solución, se implementaría un servicio en un cloud.

Sin embargo, todo lo anterior es un proyecto de mayor envergadura siendo el presente proyecto una parte del mismo: la creación de un servicio de ubicación de obras de arte.

Este sistema, permitirá que las obras de arte se encuentren localizadas en todo momento y proporcionará información rápida y precisa para ser consultada en cualquier momento que se necesite.

Además, se cree conveniente, para apoyar el proyecto y para que se pueda demostrar su funcionamiento, desarrollar brevemente el sistema que proporcione la información del patrimonio y la información de la estructura lógica del museo.

4.1. Modelo conceptual

Un primer modelo conceptual del mismo sería el siguiente:

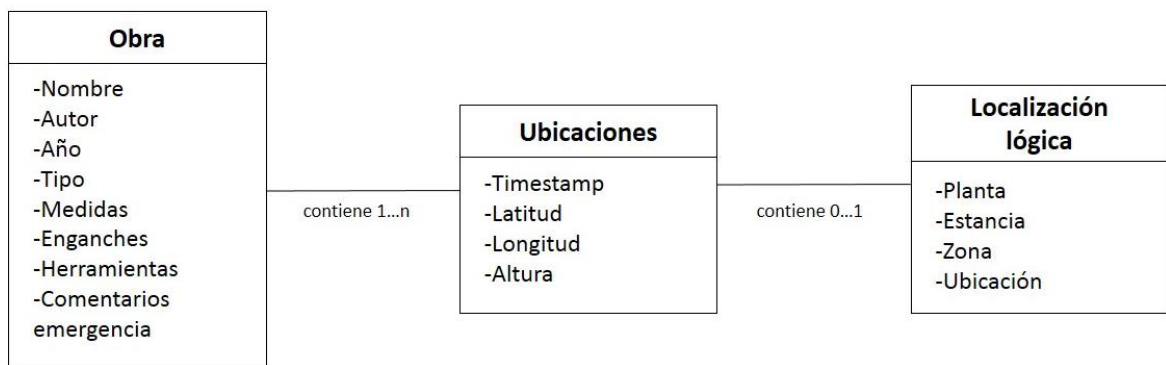


Ilustración 17: Modelo conceptual

En este modelo conceptual se pueden observar tres partes claramente definidas:

- ❖ Patrimonio cultural u obras de arte. Deberá contener la información de las obras de arte, como por ejemplo: el nombre, el autor, el año de creación etc. Pero lo más

Gestión de Emergencias del Patrimonio Cultural en la Era Digital. Diseño (e implementación) de un Sistema IoT para la auto-identificación y localización en interiores de Obras de Arte

interesante para este proyecto es la información que deberá proporcionar sobre sus mediadas, forma de fijación, herramientas necesarias para el desmontaje y otros comentarios importantes en caso de emergencia.

- ❖ Modelo lógico del museo. Contendrá las ubicaciones lógicas del edificio. Es decir, las ubicaciones de las obras siguiendo el modelo lógico: planta, estancia, zona y ubicación.
- ❖ Ubicaciones de las obras. Actuará como registro de las ubicaciones lógicas o físicas (latitud, longitud y altura) en la que están o, para tener un historial, en las que han estado las obras de arte. Para crear el registro y saber cual es la ubicación actual de la obra, se tendrá que incluir un campo timestamp que registre el momento de la actualización de la ubicación.

En cuanto a la relación entre estos componentes:

- ❖ Una obra de arte puede contener entre una o n ubicaciones. Conviene recordar que siempre hay que tener las obras de arte localizadas, por lo que nunca podrá haber una obra de arte que no tenga una ubicación. Por otra parte, puede tener n ubicaciones porque como ya hemos dicho, puede ir cambiando de zona del museo y por lo tanto se tendría un registro con su respectivo timestamp, donde el valor más alto sería la ubicación actual.
- ❖ Una ubicación puede contener 0 o 1 localizaciones lógicas. Una ubicación debe contener una, y solo una localización, pero esta puede ser lógica o física.

4.2. Casos de uso

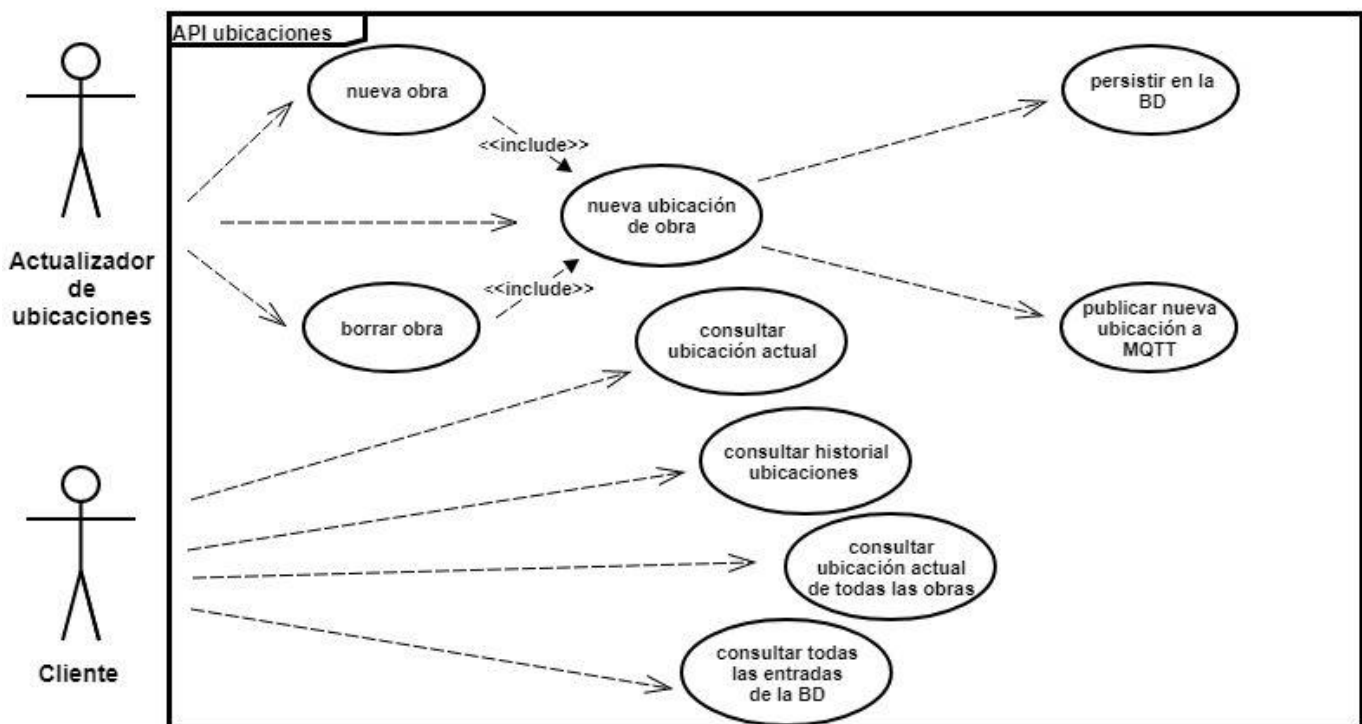


Ilustración 18: Casos de uso

El primer actor sería el actualizador de ubicaciones. Éste no se va a desarrollar en el presente proyecto, se asume que ya viene dado al ser otra parte del proyecto global, pero para ver su funcionamiento, se va a hacer una simulación del mismo.

Este actor puede realizar diferentes acciones que conllevarán respuestas del sistema. Puede añadir una nueva ubicación de una obra, añadir una nueva o borrarla. Estas dos últimas conllevan que se realice la acción de añadir una nueva ubicación. Ante cualquiera de estas acciones, el servicio responderá persistiendo la información en la base de datos y publicando en una cola de mensajería que la base de datos ha cambiado y mostrando la nueva ubicación de la obra de arte.

El segundo actor sería cualquier cliente que realizara una consulta sobre las ubicaciones del sistema. En esta parte solo serán incluidas las consultas relacionadas con las ubicaciones y no con el patrimonio cultural y el modelo lógico del museo, ya que estos dos últimos componentes se tratan en otros proyectos y en este proyecto solo se desarrollan de apoyo y para tener una visión un poco más amplia del servicio de emergencias en su totalidad.

Las consultas que se podrían ser realizadas sobre las ubicaciones son:

- ❖ Consultar la ubicación actual de una obra.
- ❖ Consultar el historial de ubicaciones de una obra.
- ❖ Consultar las ubicaciones actuales de todas las obras.
- ❖ Consultar todas las ubicaciones de todas las obras.

5. Diseño de la solución

5.1. Arquitectura de microservicios

La Arquitectura de microservicios, conocida por las siglas MSA (del inglés MicroServices Architecture) es una aproximación para el desarrollo software que consiste en construir una aplicación como un conjunto de pequeños servicios, los cuales se ejecutan en su propio proceso y se comunican con mecanismos ligeros (normalmente una API de recursos HTTP).

Cada servicio se encarga de implementar a funcionalidad completa del negocio. Son desplegados de forma independiente y pueden estar programados en distintos lenguajes y usar diferentes tecnologías de almacenamiento de datos.

Se suele considerar la arquitectura de microservicios como una forma específica de realizar una arquitectura SOA. Se ha escogido esta arquitectura porque, entre otras, posee las siguientes características, las cuales se adaptan a este proyecto:

- ❖ **Los componentes son servicios.** La principal manera de crear componentes es mediante la decomposición en servicios en lugar de librerías. Los servicios son componentes separados que se comunican mediante mecanismos como los servicios web o los RPC.
- ❖ **Organizada en torno a las funcionalidades del negocio.** El sistema se divide en distintos servicios donde cada uno está organizado en torno a una capacidad del negocio. Es muy importante limitar la responsabilidad de cada servicio.
- ❖ **Productos, no proyectos.** Sigue la idea de que un equipo debe estar a cargo de un componente (servicio) durante todo el ciclo de vida del mismo, desde la etapa de diseño y construcción, la fase de producción y hasta la de mantenimiento.
- ❖ **Extremos inteligentes y tuberías simples.** Las aplicaciones creadas desde microservicios pretender ser tan disociadas y cohesivas como sea posible, ellas poseen su propia lógica de dominio y actúan como filtros en el clásico sentido UNIX: recibe una solicitud, aplica la lógica apropiada y produce una respuesta. Estos pasos son coreografiados usando protocolos simples (típicamente HTTP con REST o mensajería liviana como RabbitMQ o ZeroMQ).
- ❖ **Tener gobierno descentralizado permite usar tecnologías que se adapten mejor a cada funcionalidad.** Con el sistema con múltiples servicios colaborativos, podemos decidir utilizar diferentes lenguajes de programación y tecnologías dentro de cada servicio. De esta forma podemos elegir la herramienta adecuada para cada tipo de trabajo en lugar de tener una estandarizada.
- ❖ **Gestión de datos descentralizada.** Los microservicios prefieren dejar a cada servicio que gestione su propia base de datos, sean éstos diferentes instancias de la misma tecnología de base de datos o sistemas de base de datos completamente diferentes.

5.2. Big Picture

Se ha diseñado un esquema general (Big Picture), el cual muestra cómo se va a desarrollar el proyecto del sistema de emergencias en el museo.



SISTEMA DE EMERGENCIAS DEL MUSEO

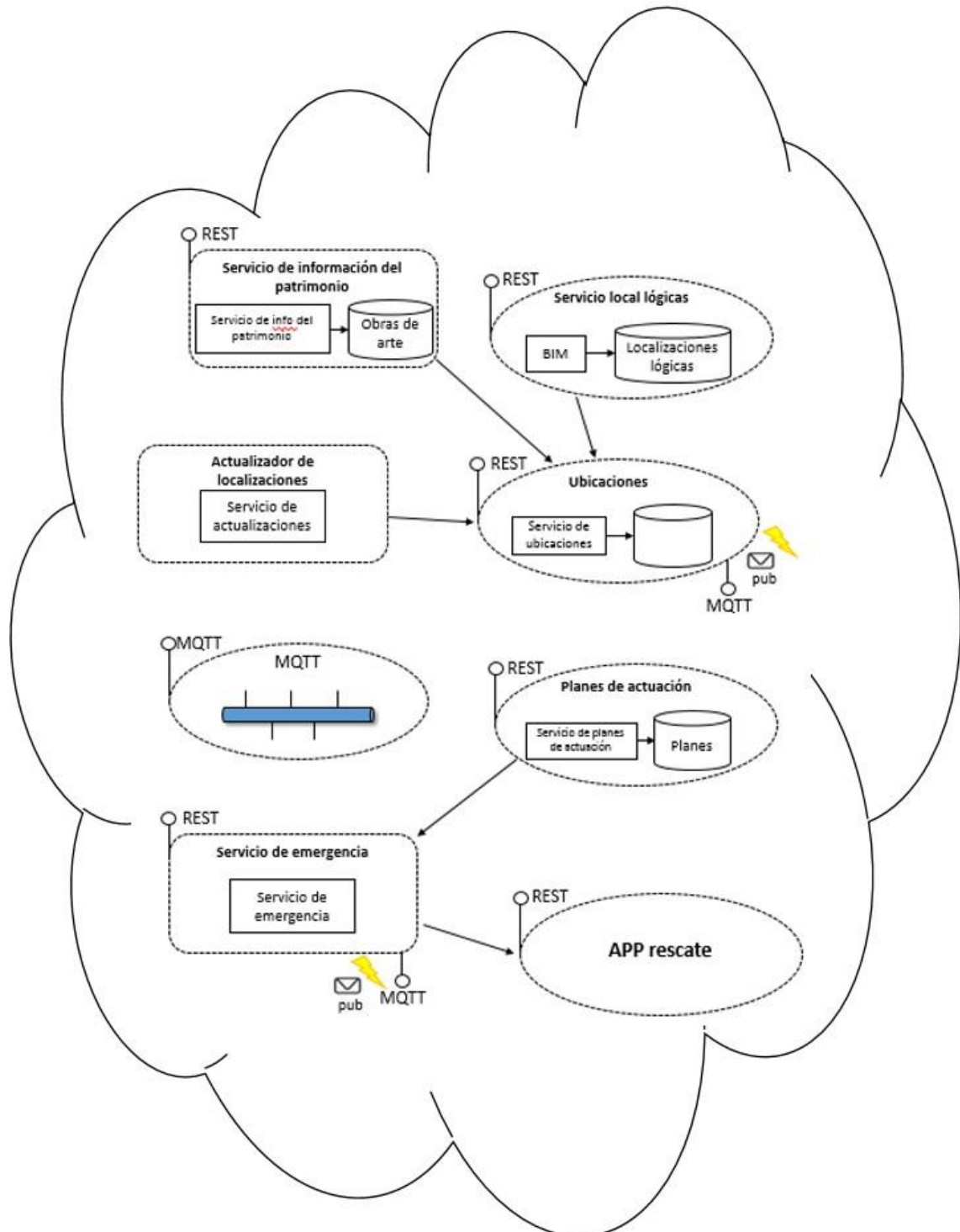


Ilustración 19: Big Picture

Gestión de Emergencias del Patrimonio Cultural en la Era Digital. Diseño (e implementación) de un Sistema IoT para la auto-identificación y localización en interiores de Obras de Arte

Como se puede observar, se ha diseñado un servicio en la nube que está conformado a su vez de distintos microservicios relacionados entre sí.

Los diferentes microservicios que se pueden identificar son:

- ❖ **Servicio de información del patrimonio.** Este servicio contendrá una base de datos con la información de todas las obras de arte pertenecientes al museo. Contiene información tanto descriptiva como útil para tratar cualquier emergencia. El servicio también ofrecerá una API REST para consultar esta base de datos.
- ❖ **Servicio de estructura lógica del museo.** En este servicio se encontrará la base de datos que contiene la descripción de las ubicaciones lógicas donde pueden o están ubicadas las obras dentro del museo conforme al modelo lógico extraído del edificio, es decir: en qué planta está ubicada, en qué estancia, en qué zona, etc.
- ❖ **Servicio actualizador de localizaciones.** Este servicio estará compuesto por la tecnología instalada en todas las obras de arte. Cada vez que una obra de arte se mueva a una ubicación diferente, enviará un mensaje al siguiente módulo, que hará de registro.
- ❖ **Servicio de ubicación de las obras de arte.** Este módulo contendrá una base de datos con las ubicaciones de las obras del museo, tanto las actuales como un registro de las mismas. Ofrecerá un servicio REST para que cualquier otro módulo que necesite esta información pueda consultarla. Además, cada vez que el servicio de actualización de ubicaciones envíe una actualización, publicará un mensaje a la cola de mensajería MQTT indicando que una obra ha cambiado de ubicación y enviando la información la nueva información para que los servicios que estén suscritos a ella, sean conocedores del cambio en tiempo real.
- ❖ **Cola de mensajería MQTT.** Este módulo tendrá la función de broker de mensajes entre los módulos que publiquen cambios y los módulos que se suscriban a estas publicaciones.
- ❖ **Planes de actuación.** Este módulo contendrá, en una base de datos, los planes de actuación aplicables a las diferentes emergencias contempladas. También ofrece un servicio para que otros módulos puedan obtener el plan que necesiten.
- ❖ **Servicio de emergencia.** Este módulo contendrá un servicio que se activará en caso de emergencia. Dependiendo del tipo de emergencia, recibirá el plan de actuación del módulo anterior y consultando las ubicaciones de las obras de arte del módulo correspondiente, enviará la información de rescate o evacuación a una app para que los trabajadores sean conocedores de los pasos a seguir. Estará suscrito a la cola de mensajería MQTT para saber en todo momento los cambios de ubicaciones de las obras de arte.
- ❖ **APP de rescate.** Los operarios o trabajadores del museo recibirán información del plan de rescate de las obras: obras prioritarias, donde está ubicada, con que herramientas, por donde tienen que evacuar, etc.

5.3. Comunicaciones entre servicios

En este apartado se definirán las comunicaciones entre los diferentes microservicios. Algunos ofrecerán una API REST a la que se le pueden realizar diferentes peticiones HTTP:

- ❖ Servicio de información del patrimonio.
- ❖ Servicio de localizaciones lógicas.
- ❖ Servicio de ubicación de obras de arte.
- ❖ Servicio de planes de actuación.
- ❖ Servicio de emergencias.

Se ha escogido esta forma de comunicación por ciertas ventajas que ofrece REST para este tipo de proyectos:

- ❖ **Separación de cliente/servidor.** Al ser sistemas independientes (solo se comunican con un lenguaje de intercambio como JSON), se pueden desarrollar los diferentes microservicios de forma autónoma. No es relevante para el cliente cómo está desarrollada la API y tampoco lo es para el servidor qué se va a hacer con los datos que proporciona.
- ❖ **Independencia de tecnologías/lenguajes.** Se puede desarrollar en cualquier tipo de tecnología o lenguaje. Es indiferente si en el futuro se cambian totalmente las tecnologías con las que está implementada la API REST, siempre y cuando se respete “el contrato”, es decir, que tenga las mismas operaciones en la API y haga las mismas cosas que se supone que deben hacer.
- ❖ **Fiabilidad, escalabilidad y flexibilidad.** Solo hay que preocuparse de que el nexo cliente/servidor esté correcto. Se pueden hacer cambios en el servidor, lenguajes, bases de datos, etc. Mientras se devuelvan los datos correctos todo funcionará correctamente.
La API puede crecer en todo momento y responder a otro tipo de operaciones o versionarse tanto como se desee. El lado del cliente también puede crecer todo lo necesario con el tiempo, incluso crear otros frontales, no solo web, también Apps para cualquier dispositivo.
Las APIs pueden estar alojadas en servidores independientes, tantos como se necesite. Por la característica stateless de REST, es indiferente qué servidor atienda cada solicitud, así que el balanceo de carga es mucho más simple.
- ❖ **Experiencia del usuario.** Cuando se hace un solicitud, lo que se recibe son datos planos, que requieren tiempos de transferencia menores.
- ❖ **Requiere menos recursos del servidor.** Como no mantiene estado, no requiere memoria y se pueden atender más peticiones. Además, al no requerir escribir el HTML, se necesita menos procesamiento del servidor.

Aunque el microservicio a desarrollar en este proyecto sea el servicio de ubicaciones de obras de arte, como ya se ha dicho, se han desarrollado resumidamente los servicios de información del patrimonio y de localizaciones lógicas, para tener una visión más amplia del proyecto global y para mostrar su funcionamiento. Por lo tanto, en este proyecto se van a desarrollar diferentes APIs REST:



Gestión de Emergencias del Patrimonio Cultural en la Era Digital. Diseño (e implementación) de un Sistema IoT para la auto-identificación y localización en interiores de Obras de Arte

- ❖ **La API REST del servicio de información del patrimonio**, dónde se podrá consultar la información de las obras de arte, añadir una nueva obra, modificarla o borrarla.
- ❖ **La API REST de las localizaciones lógicas del museo**. Se podrá consultar las localizaciones lógicas, añadir nuevas, modificarlas o borrarlas.
- ❖ **La API REST del servicio de ubicaciones de obras de arte**. En ésta se podrá consultar la ubicación actual de una obra dada, el historial de la misma o la ubicación actual de todas las obras de arte del museo. También atenderá las peticiones del servicio modificador de ubicaciones para añadir nuevas ubicaciones de las obras.

Los mensajes de intercambio de información para las APIs se realizaran mediante JSON porque es más simple que XML, puede representar cualquier estructura de dato, pudiendo añadir nuevos campos con total facilidad y puede ser leído por cualquier lenguaje de programación.

Para la cola de mensajería, se ha utilizado la tecnología MQTT porque se puede utilizar para diversas funcionalidades de pasos de mensajes dentro el cloud, además de unas características que lo hacen idóneo como: el ancho de banda utilizado es mínimo, es muy rápido y posibilita un tiempo de respuesta superior al resto de protocolos web actuales, gran fiabilidad, requiere pocos recursos de procesadores y memorias. Además, al intercambiar datos que pueden ser críticos, es posible garantizar la seguridad del intercambio mediante tres niveles: transporte en SSL/TLS, autenticación mediante certificado SSL/TLS y autenticación mediante usuario y contraseña.

En este caso, el servicio de ubicaciones de las obras de arte envía una publicación al broker cuando se produce un cambio en la ubicación de una obra con el mensaje de que ha habido algún cambio y mostrando la nueva información. El broker publica inmediatamente esta información a todos los servicios que estén suscritos al “topic” publicado.

Como ya hemos dicho, la información de las ubicaciones de las obras es crítica, por tanto, esta solución nos resuelve el problema enviando los cambios en tiempo real, así el servicio que necesite esta información, no tiene que estar enviando peticiones REST continuamente ni se quedará nunca desactualizado.

5.4. Base de datos

Para este proyecto se ha diseñado una base de datos donde, por simplicidad, se guardaran tanto la información de las ubicaciones de las obras, como la información del patrimonio y las ubicaciones lógicas.

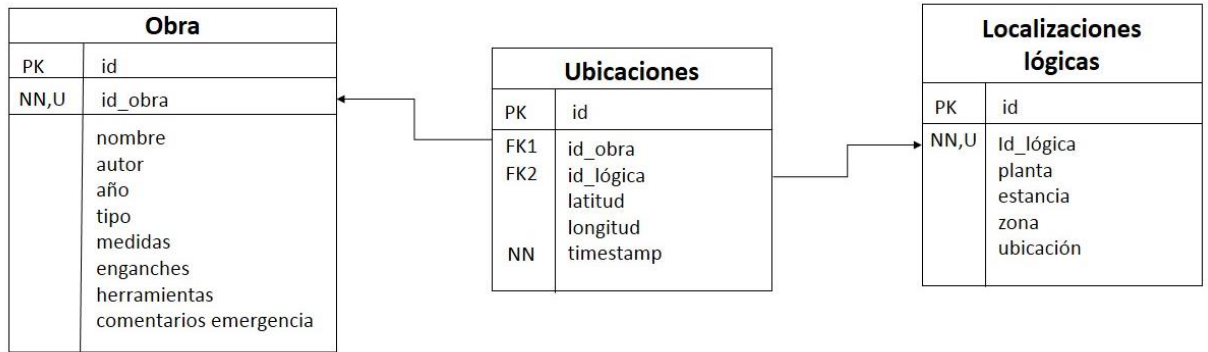


Ilustración 20: Diagrama de la base de datos

6. Implementación

En el presente apartado de implementación se describe qué capas serán desarrolladas, y el desarrollo de cada una de ellas.

6.1. Capa de persistencia

La base de datos se ha implementado en PostgreSQL, basándose en el diagrama de la base de datos mostrado en el capítulo anterior. Es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente.

La herramienta de gestión utilizada es PgAdmin 4. Es la herramienta de gestión Open Source más utilizada para administrar y desarrollar PostgreSQL. PgAdmin 4 proporciona una potente interfaz gráfica que simplifica la creación, el mantenimiento y el uso de objetos de base de datos.

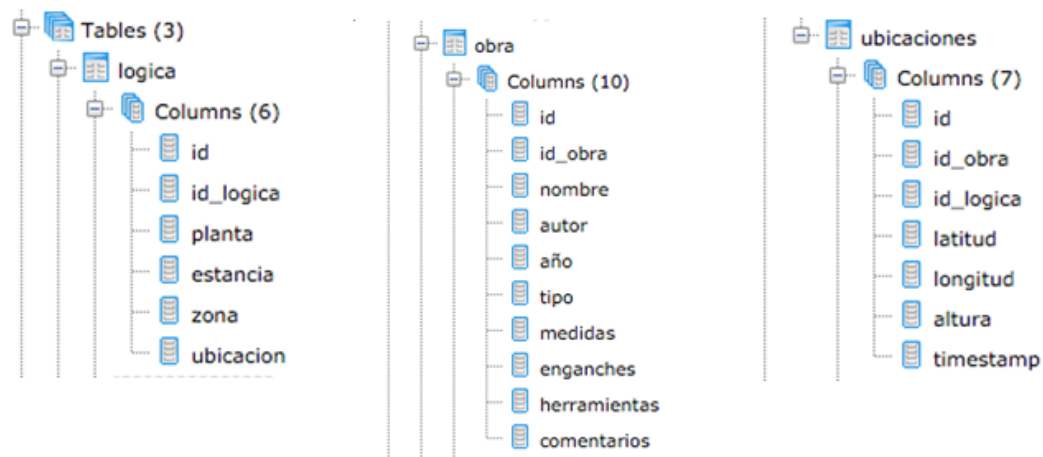


Ilustración 21: Tablas en PgAdmin 4

6.2. Capa lógica

Esta capa se va a implementar en el lenguaje de programación java y en el IDE NetBeans.

6.2.1. Entidades

Una entidad de persistencia (entity) es una clase de Java ligera, cuyo estado es persistido de manera asociada a una tabla en una base de datos relacional.

Se ha creado una entidad para cada tabla diferente de la base de datos. En la siguiente figura mostramos la definición de la entidad y sus variables, que se corresponden con las columnas de la tabla (en este caso, y como ejemplo, la tabla Ubicaciones).

```
@Entity
@Table(name = "ubicaciones")
public class Ubicaciones implements Serializable {

    private static final long serialVersionUID = 1L;
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Basic(optional = false)
    @Column(name = "id")
    private Integer id;
    @Size(max = 2147483647)
    @Column(name = "latitud")
    private String latitud;
    @Size(max = 2147483647)
    @Column(name = "longitud")
    private String longitud;
    @Size(max = 2147483647)
    @Column(name = "altura")
    private String altura;
    @Basic(optional = false)
    @NotNull
    @Column(name = "timestamp")
    @Temporal(TemporalType.DATE)
    private Date timestamp;
    @JoinColumn(name = "id_logica", referencedColumnName = "id_logica")
    @ManyToOne
    private Logica idLogica;
    @JoinColumn(name = "id_obra", referencedColumnName = "id_obra")
    @ManyToOne(optional = false)
    private Obra idObra;
```

Ilustración 22: Entidad ubicación

6.2.2. AbstractFacade

Se va a utilizar el patrón estructural AbstractFacade. Una facade o fachada es un objeto que crea una interfaz simplificada para tratar con otra parte del código más compleja, de tal forma que simplifica y aísla su uso.

Se utilizará AbstractFacade para realizar acciones sobre las entidades creadas en el punto anterior.

Gestión de Emergencias del Patrimonio Cultural en la Era Digital. Diseño (e implementación) de un Sistema IoT para la auto-identificación y localización en interiores de Obras de Arte

AbstractFacade tratará con las entidades creadas pudiendo utilizar estos métodos por defecto:

```
public abstract class AbstractFacade<T> {
    private Class<T> entityClass;

    public AbstractFacade(Class<T> entityClass) {
        this.entityClass = entityClass;
    }

    protected abstract EntityManager getEntityManager();

    public void create(T entity) {
        getEntityManager().persist(entity);
    }

    public void edit(T entity) {
        getEntityManager().merge(entity);
    }

    public void remove(T entity) {
        getEntityManager().remove(getEntityManager().merge(entity));
    }

    public T find(Object id) {
        return getEntityManager().find(entityClass, id);
    }

    public List<T> findAll() {
        javax.persistence.criteria.CriteriaQuery cq = getEntityManager().getCriteriaBuilder().createQuery();
        cq.select(cq.from(entityClass));
        return getEntityManager().createQuery(cq).getResultList();
    }

    public List<T> findRange(int[] range) {
        javax.persistence.criteria.CriteriaQuery cq = getEntityManager().getCriteriaBuilder().createQuery();
        cq.select(cq.from(entityClass));
        javax.persistence.Query q = getEntityManager().createQuery(cq);
        q.setMaxResults(range[1] - range[0] + 1);
        q.setFirstResult(range[0]);
        return q.getResultList();
    }

    public int count() {
        javax.persistence.criteria.CriteriaQuery cq = getEntityManager().getCriteriaBuilder().createQuery();
        javax.persistence.criteria.Root<T> rt = cq.from(entityClass);
        cq.select(getEntityManager().getCriteriaBuilder().count(rt));
        javax.persistence.Query q = getEntityManager().createQuery(cq);
        return ((Long) q.getSingleResult()).intValue();
    }
}
```

Ilustración 23: Métodos AbstractFacade

6.2.3. APIs REST

A continuación, se muestran los posibles métodos HTTP que se pueden utilizar para realizar peticiones a la API.

La URI de esta API es /Museo/Recursos/Ubicaciones/ y cada método sigue un “path” específico para realizar su petición. Por ejemplo, la petición GET para obtener el historial de ubicaciones de una obra de arte se realiza sobre la URI /Museo/Recursos/Ubicaciones/{id}/historial. Siendo {id} el identificador de la obra.

```

@Stateless
@Path("Ubicaciones")
public class UbicacionesFacadeREST extends AbstractFacade<Ubicaciones> {

    @PersistenceContext(unitName = "MuseoPU")
    private EntityManager em;

    public UbicacionesFacadeREST() {
        super(Ubicaciones.class);
    }

    MQTT Cola = new MQTT();

    @POST
    @Override
    @Consumes({MediaType.APPLICATION_JSON})
    public void create(Ubicaciones entity){
        super.create(entity);
        Cola.publicar(entity);
    }

    @PUT
    @Path("/{id}")
    @Consumes({MediaType.APPLICATION_JSON})
    public void edit(@PathParam("id") Integer id, Ubicaciones entity) {
        super.edit(entity);
        Cola.publicar(entity);
    }

    @DELETE
    @Path("/{id}")
    public void remove(@PathParam("id") Integer id) {
        super.remove(super.find(id));
    }

    @GET
    @Path("/{id}")
    @Produces({MediaType.APPLICATION_JSON})
    public Ubicaciones find(@PathParam("id") String idObra) throws SQLException {
        return super.findById(idObra);
    }

    @GET
    @Path("/{id}/historial")
    @Produces({MediaType.APPLICATION_JSON})
    public List<Ubicaciones> findByList(@PathParam("id") String idObra) {
        String id_obra = "id_obra";
        return super.findByList(id_obra, idObra);
    }

    @GET
    @Path("actual")
    @Produces({MediaType.APPLICATION_JSON})
    public List<Ubicaciones> findByList()throws SQLException {
        return super.findActual();
    }

    @GET
    @Override
    @Path ("all")
    @Produces({MediaType.APPLICATION_JSON})
    public List<Ubicaciones> findAll() {
        return super.findAll();
    }

    @GET
    @Path("count")
    @Produces(MediaType.TEXT_PLAIN)
    public String countREST() {
        return String.valueOf(super.count());
    }
}

```

Ilustración 24: API ubicaciones

Gestión de Emergencias del Patrimonio Cultural en la Era Digital. Diseño (e implementación) de un Sistema IoT para la auto-identificación y localización en interiores de Obras de Arte

Para resolver algunas peticiones, se han añadido a la clase AbstractFacade mencionada en el punto anterior, los siguientes métodos.

```
public List<T> findByList(String property, String idObra) {
    javax.persistence.criteria.CriteriaQuery cq = getEntityManager().getCriteriaBuilder().createQuery();
    cq.select(cq.from(entityClass));
    return getEntityManager().createNativeQuery("SELECT id, id_obra, id_logica, latitud, longitud, altura, timestamp FROM "
        + entityClass.getSimpleName() + " " + "c WHERE c." + property + " = '" + idObra +
        "' ORDER BY timestamp DESC", entityClass).getResultList();
}

public T findById(String idObra) throws SQLException {
    Connection con = DriverManager.getConnection("jdbc:postgresql://localhost:5555/Museo", "postgres", "2516");
    Statement stmt = con.createStatement();
    String query = "SELECT * from ubicaciones where id_obra='" + idObra + "' ORDER BY timestamp DESC LIMIT 1 ";
    ResultSet result = stmt.executeQuery(query);
    result.next();
    int id1 = result.getInt("ID");
    return find(id1);
}

public List<T> findActual() throws SQLException {
    javax.persistence.criteria.CriteriaQuery cq = getEntityManager().getCriteriaBuilder().createQuery();
    cq.select(cq.from(entityClass));
    return getEntityManager().createNativeQuery("select * from "
        + "(select *, row_number() over (partition by id_obra order by timestamp desc)"
        + " as row_number from ubicaciones) as rows where row_number = 1", entityClass).getResultList();
}
```

Ilustración 25: Métodos añadidos a AbstractFacade

También se han realizado las APIs de los servicios de información del patrimonio y de localizaciones lógicas del museo, siendo su método más significativo para este proyecto, el método GET para obtener los datos de una obra o localización lógica según su identificador.

6.2.4. MQTT

A continuación se muestra el método de la clase MQTT llamado cuando se añade una nueva ubicación.

```
public void publicar(Ubicaciones entity){
    try {
        client = new MqttClient("tcp://m2m.eclipse.org:1883", "pahomqttpublish1");
        client.connect();

        MqttMessage message = new MqttMessage();
        String mensaje= "Se ha producido el siguiente cambio de ubicación en una obra: \n\n";
        mensaje+= entity.toString();
        message.setPayload(mensaje.getBytes());

        client.publish("Ubicaciones", message);
        client.disconnect();
    } catch (MqttException e) {
        e.printStackTrace();
    }
}
```

Ilustración 26: Método de publicación mediante MQTT

Sirve para publicar un mensaje al broker (en este caso se ha utilizado un sandbox de eclipse) con el “topic” Ubicaciones, para que todo servicio que esté suscrito a este “topic”, reciba la actualización en tiempo real.



7. Ejemplo de uso del escenario

En este capítulo se va mostrar un ejemplo de uso de la API que ofrece el servicio de ubicaciones de las obras de arte desarrollado.

7.1. Caso de uso Consultar la ubicación de una obra

El cliente accede a la ubicación actual de la obra “RET013” mediante un método GET en la URI mostrada (figura).

> <http://localhost:8080/Museo/Recursos/Ubicaciones/RET013>

GET POST PUT DELETE PATCH ▾

Ilustración 27: Método consulta última ubicación

La API devuelve, de entre las ubicaciones de la misma obra en la base de datos, la que tiene un timestamp más alto, es decir, la ubicación actual.

```
{
  "id": 4,
  "-idLogica": {
    "estancia": "1",
    "id": 1,
    "idLogica": "L01",
    "planta": "1"
  },
  "-idObra": {
    "id": 17,
    "idObra": "RET013",
    "nombre": "retable 13 "
  },
  "timestamp": "2017-02-02T00:00:00"
}
```

Ilustración 28: Ubicación actual devuelta por la API

7.2. Caso de uso consultar el historial de una obra

El cliente consulta el historial de ubicaciones de la obra RET012 mediante una solicitud GET enviada en la URI mostrada en la siguiente figura.

> <http://localhost:8080/Museo/Recursos/Ubicaciones/RET012/historial>

GET POST PUT DELETE PATCH ▾

Ilustración 29: Método de consulta del historial

La API devuelve el historial de todas las ubicaciones donde ha estado la obra.

```
[Array[3]
  -0: {
    "altura": "10",
    "id": 5,
    -"idObra": {
      "id": 1,
      "idObra": "RET012",
      "nombre": "Retable 12"
    },
    "latitud": "39.448453",
    "longitud": "-0.372472",
    "timestamp": "2017-02-07T00:00:00"
  },
  -1: {
    "id": 2,
    -"idLogica": {
      "estancia": "2",
      "id": 16,
      "idLogica": "L02",
      "planta": "2"
    },
    -"idObra": {
      "id": 1,
      "idObra": "RET012",
      "nombre": "Retable 12"
    },
    "timestamp": "2017-02-02T00:00:00"
  },
  -2: {
    "id": 1,
    -"idLogica": {
      "estancia": "1",
      "id": 1,
      "idLogica": "L01",
      "planta": "1"
    },
    -"idObra": {
      "id": 1,
      "idObra": "RET012",
      "nombre": "Retable 12"
    },
    "timestamp": "2016-01-01T00:00:00"
  }
],
```

Ilustración 30: Historial de ubicaciones

Gestión de Emergencias del Patrimonio Cultural en la Era Digital. Diseño (e implementación) de un Sistema IoT para la auto-identificación y localización en interiores de Obras de Arte

7.3. Caso de uso consultar ubicaciones actuales de todas las obras

Para consultar el estado las ubicaciones actuales de todas las obras del museo, el cliente envía una solicitud GET en la URI que se muestra en la siguiente figura.

> <http://localhost:8080/Museo/Recursos/Ubicaciones/actual>

GET POST PUT DELETE PATCH

Ilustración 31: Método consulta ubicación todas las obras

La API REST devuelve la ubicación actual de todas las obras que pertenecen al museo.

```
[Array[2]
  -0: {
    "altura": "10",
    "id": 5,
    -"idObra": {
      "id": 1,
      "idObra": "RET012",
      "nombre": "Retable 12"
    },
    "latitud": "39.448453",
    "longitud": "-0.372472",
    "timestamp": "2017-02-07T00:00:00"
  },
  -1: {
    "id": 4,
    -"idLogica": {
      "estancia": "1",
      "id": 1,
      "idLogica": "L01",
      "planta": "1"
    },
    -"idObra": {
      "id": 17,
      "idObra": "RET013",
      "nombre": "retable 13 "
    },
    "timestamp": "2017-02-02T00:00:00"
  }
],
```

Ilustración 32: Ubicación actual de todas las obras

7.4. Caso de uso añadir nueva ubicación de una obra

Este método POST en la URI indicada (figura) es una simulación del mensaje que enviaría el microservicio actualizador de ubicaciones de las obras (figura).

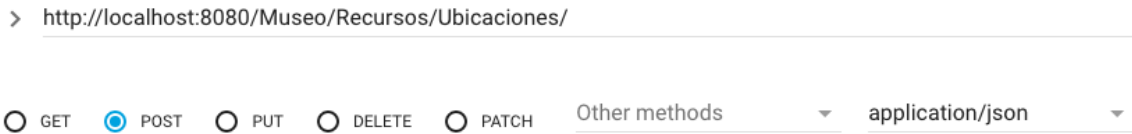


Ilustración 33: Método añadir nueva ubicación

```
{
  "idLogica": {
    "estancia": "1",
    "id": 1,
    "idLogica": "L02",
    "planta": "1"
  },
  "idObra": {
    "id": 17,
    "idObra": "RET013",
    "nombre": "retable 13 "
  },
  "timestamp": "2017-07-07T00:00:00"
}
```

Ilustración 34: JSON enviado con el método POST

Tras persistirse la nueva ubicación en la base de datos, se publica a la cola MQTT el cambio para que los servicios que estén suscritos al topic “Ubicaciones” reciban los datos. Mensaje recibido mediante la suscripción al topic “Ubicaciones” en la MQTT

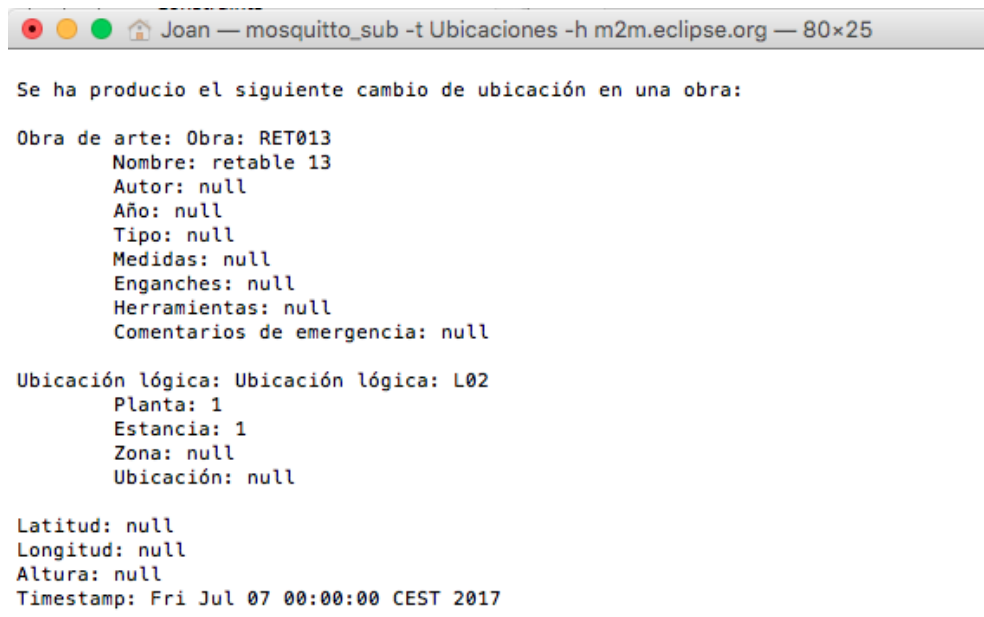


Ilustración 35: Mensaje enviado a los suscriptores de MQTT

8. Conclusiones

En este último capítulo a redactar tendrá por objeto definir una conclusión, objetivos conseguidos y no conseguidos, trabajos futuros y valoración personal.

8.1. Conclusión

Una herramienta clave para los museos en estos días es la implantación de un sistema de emergencias eficaz.

Existe una amplia variedad de tecnologías de posicionamiento en interiores que, actualmente, se encuentran en auge. Prueba de ello es la cantidad de proyectos que nacen en este campo.

La arquitectura formada por microservicios es muy eficaz, ya que se desarrollan componentes independientes entre sí, no necesitan ser implementados con la misma tecnología y se comunican por protocolos simples.

8.2. Objetivos conseguidos y objetivos no conseguidos

Se ha conseguido desarrollar un prototipo de microservicio que realizaría las acciones que se le requieren: ofrecer una API donde poder añadir nuevas ubicaciones y hacer diferentes consultas y publicar cambios en una cola de mensajería MQTT

No se ha conseguido juntar todos los proyectos para crear y probar el servicio global “Servicio de emergencias en un museo”

8.3. Trabajos futuros

Este servicio va a continuar siendo desarrollado por la necesidad de los museos de este tipo de servicios de emergencias.

En concreto, el Institut Valencià de Conservació i Restauració de Béns Culturals (IVC+R) junto con un grupo de investigación de la UPV, tiene la intención de realizar este proyecto.

9. Bibliografía

- [1]. <https://www.by.com.es/blog/que-es-rfid/>
- [2]. <http://www.dipolerfid.es/es/tecnologia-RFID>
- [3]. <http://www.adwe.es/general/colaboraciones/servicios-web-restful-con-http-parte-i-introduccion-y-bases-teoricas>
- [4]. http://en.wikipedia.org/wiki/Representational_state_transfer
- [5]. <http://www.ietf.org/rfc/rfc3986.txt> (1.1.3)
- [6]. <http://www.w3.org/Protocols/HTTP/1.1/draft-ietf-http-v11-spec-rev-06.txt>
- [7]. <https://unpocodejava.wordpress.com/2012/12/06/que-es-mqtt/>
- [8]. <https://www.infoq.com/articles/practical-mqtt-with-paho>
- [9]. <https://situm.es/es>
- [10]. <https://www.pozyx.io/>
- [11]. <https://indoo.rs/>
- [12]. <https://www.accuware.com/>
- [13]. <http://meridianapps.com/>
- [14]. <http://seeketing.com/corp/index.php/es/blog/56-ibeacons-que-funcionan-sin-app-los-nodos-seeketing>
- [15]. https://es.wikipedia.org/wiki/Arquitectura_de_microservicios
- [16]. <https://desarrolloweb.com/articulos/ventajas-inconvenientes-apirest-desarrollo.html>
- [17]. <http://www.hoplasoftware.com/2016/12/ventajas-de-postgresql/>

