



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Sistema de soporte a la localización de recursos en entornos industriales evolucionando hacia a un mundo digital. Una solución aplicada

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Carrillo Calvo, Adrián

Tutor/a: Fons Cors, Joan Josep

CURSO ACADÉMICO: 2022/2023

Sistema de soporte a la localización de recursos en entornos industriales
evolucionando hacia a un mundo digital. Una solución aplicada

Resumen

Actualmente, en un mundo en constante evolución y donde la industria está viviendo su gran revolución hacia la Industria 4.0, apoyada y fomentada por los gigantes tecnológicos, es necesario dotar de soluciones a cualquier aspecto mejorable.

En este trabajo se va a presentar una solución aplicada a Embalpack para mejorar la trazabilidad y localización de sus productos en el almacén. Se creará un nuevo sistema para el registro de localizaciones de cada producto, que podrá ser interconectado con otras aplicaciones externas y sumará a la evolución tecnológica que viven las PYMES de hoy en día.

La solución además se crea con unas premisas básicas: no debe suponer un coste alto para la empresa y debe tener un mínimo impacto, tanto para el funcionamiento habitual de la empresa como para el personal que implante el sistema y su mantenimiento diario.

Palabras clave: software, base de datos, localización, sistema.

Abstract

Currently, in a world that is constantly evolving and where the industry is experiencing its great revolution towards Industry 4.0, supported and promoted by technological giants, it is necessary to provide solutions to any aspect that can be improved.

In this work, a solution applied to Embalpack will be presented to improve the traceability and location of its products in the warehouse. A new system for location registration system will be created for each product, which can be interconnected with other external applications and will add to the technological evolution that SMEs are experiencing today.

The solution is also created with some basic premises: it should not entail a high cost for the company and should have a minimum impact, both for the normal operation of the company and for the personnel who implement the system and its daily maintenance.

Keywords: software, database, location, system.

Sistema de soporte a la localización de recursos en entornos industriales evolucionando hacia a un mundo digital. Una solución aplicada

Tabla de contenidos

1.	Introducción	9
1.1	Motivación	9
1.2	Objetivos	9
2.	Estado del arte	10
2.1	Software.	10
2.2	Comunicación inalámbrica.....	10
2.2.1	Wifi.....	11
2.2.2	Zigbee	11
2.2.3	Bluetooth.....	12
2.2.4	GSM.....	12
2.3	Intercambio de mensajes.....	13
2.3.1	MQTT.....	13
2.3.2	AMQP.	13
2.4	Bases de datos.....	13
2.4.1	SQL.....	14
2.4.2	No-SQL.....	14
2.5	Soluciones ya existentes	14
3.	Análisis del problema.....	15
3.1	Problema físico.....	15
3.2	Problema tecnológico.	16
3.3	Seguridad:	16
3.3.1	Comunicaciones:.....	16
3.3.2	Base de datos:	16
3.3.3	Integridad de datos:	17
3.4	Presupuesto:	18
4.	Modelo de localización.....	20
4.1	Modelos aleatorios para almacenes automatizados.....	20
4.2	Modelos clásicos.	22
4.3	Tabla comparativa:	23
5.	Diseño de la solución.....	25



5.1	Arquitectura general del sistema.....	25
5.2	Arquitectura detallada del sistema.	26
5.2.1	Estructura de tablas SQL	26
5.2.2	Casos de uso.	27
5.2.3	Colas de mensajería.	28
5.2.4	Consulta de datos	29
5.2.5	Consulta de datos desde fuentes externas.....	29
5.2.6	Terminales de operarios.....	31
5.2.7	Modelo de localización 1:	33
5.2.8	Modelo de localización 2:	35
5.2.9	Destino automático y control de la ocupación.....	35
6.	Implementación prototipo y pruebas	37
7.	Conclusiones	41
8.	Referencias	42

Tabla de ilustraciones

Ilustración 1: Ranking de lenguajes de programación más utilizados.....	10
Ilustración 2. Diagrama Zigbee	11
Ilustración 3. Esquema básico GSM	12
Ilustración 4. Logo Mosquito	13
Ilustración 5. Logo RabbitMQ.....	13
Ilustración 6: Esquema de seguridad.....	17
Ilustración 7: Estantería en Amazon	21
Ilustración 8: Sistema carrusel Fedefarma.....	22
Ilustración 9: Almacén clásico. Tipos de áridos similares almacenados de manera contigua.....	23
Ilustración 10: Diagrama de arquitectura.....	25
Ilustración 11: Diagrama de esquema tablas	26
Ilustración 12: Ejemplo tráfico para cola de operarios.....	28
Ilustración 13: Ejemplo tráfico para consultas	29
Ilustración 14: API REST	30
Ilustración 15: Representación en código de barras y QR de un ID concreto.....	31
Ilustración 16: Tipos de pistolas lectoras de códigos de barra o QR	32
Ilustración 17: Teclado numérico	32
Ilustración 18: Diagrama de disposición de AP Wifi para posicionamiento.....	33
Ilustración 19: Diagrama beacons BLE para posicionamiento.....	34
Ilustración 20: Diagrama ubicaciones Embalpack S.L.....	35
Ilustración 21: Secuencia ejecución registros	37
Ilustración 22: Emisor de información	38
Ilustración 23: Receptor en servidor para procesado de mensajes	38
Ilustración 24: Búsqueda último registro si lo hay y liberación de esa localización.....	39
Ilustración 25: Grabación del nuevo registro y ocupación de la nueva ocupación	39
Ilustración 26: Grafica llegada mensajes.....	40
Ilustración 27: Resultado inserciones.....	40
Ilustración 28: Resultado tabla ocupación.....	40

Sistema de soporte a la localización de recursos en entornos industriales evolucionando hacia a un mundo digital. Una solución aplicada

1. Introducción

En la actualidad, con el auge del comercio electrónico, el consumismo y la instantaneidad que desean los consumidores; las empresas se ven obligados a hacer uso de la tecnología para mejorar sus servicios. Esto ya se puede observar, pues programas como los ERP y los CRM ya son ampliamente utilizados por la mayoría de las empresas. En el primero de los casos le otorga un soporte a la hora de gestionar pedidos, stocks de almacén, contabilidad y facturación. En el segundo de los casos monitorizar la relación con los clientes, emprender acciones comerciales y de marketing. Como adenda a todo esto y dependiendo del sector también se puede ver software de tipo industrial para producción, regulatorio para industria alimentaria y química, o de gestión y seguimiento para logística.

Todo esto ya está normalizado, a pequeña que sea una empresa, mínimo va a haber dos o tres tipos de software distintos, ayudado principalmente por la variedad que hay en la oferta de servicios de IT disponibles hoy en día. Por supuesto que la gran mayoría de estos servicios son útiles y funcionales, cumplen con su cometido y ayudan, tanto al cliente como a la empresa a mejorar su experiencia y su ciclo de vida.

Sin embargo, en la mayoría de los casos se pueden encontrar reticencias a la hora de incorporar estos servicios. Principalmente el ROI, aunque cuando ya se valora la solución, el miedo al cambio, el estado de ciberseguridad, y el aprendizaje por parte de los usuarios pueden acabar apareciendo.

Por eso, aportar una solución mínimamente invasiva pero que cumpla su cometido consigue una gran aceptación.

1.1 Motivación

La motivación principal para este proyecto es la creación de una herramienta que sea útil para el futuro, trabajar sobre un caso práctico que me permita aplicar los conocimientos adquiridos en la carrera como si se tratara de la realidad.

La elección del tema, siempre complicada se facilita en el momento que nos imparten la asignatura de Integración de Aplicaciones, en la cual nos enseñan a interconectar dispositivos diversos para que trabajen en conjunto y puedan proveer de información y actuar de manera autónoma.

Además, las prácticas, realizadas en un entorno industrial y el posterior trabajo, en el que realizo integraciones entre un software industrial y programas de gestión tipo ERP de los clientes.

1.2 Objetivos

Los objetivos para alcanzar para este proyecto serán los siguientes:

- Analizar el problema y las posibles soluciones
- Dar una solución de bajo coste y bajo impacto
- Permitir la interconexión con otros servicios

2. Estado del arte

Es necesario analizar en qué estado se encuentra la tecnología que queremos aplicar en la solución antes de diseñarla. Para ello se secciona en los distintos ámbitos que se van a emplear:

2.1 Software.

Actualmente, el apartado del software proporciona una variedad de posibilidades que, en algunos casos, puede dificultar la elección. Según un estudio de la Northeastern University de Seattle basado en las ofertas de la página de búsqueda de empleo Indeed [1], éstos son los 10 lenguajes de programación más populares en 2020.

TOP 10
Popular Programming
Languages in 2020

1	Python
2	JavaScript
3	Java
4	C#
5	C
6	C++
7	GO
8	R
9	Swift
10	PHP

WWW.NORTHEASTERN.EDU/GRADUATE

Ilustración 1: Ranking de lenguajes de programación más utilizados

Gracias a esta clasificación, podemos ver que hay variedad en las ofertas. Esto lleva a pensar que dependiendo del objetivo o desarrollo que se vaya a realizar, las tecnologías pueden ir cambiando para adaptarse mejor al contexto o al ecosistema que se vaya a emplear.

2.2 Comunicación inalámbrica.

Para comunicar diferentes dispositivos y aplicaciones en un entorno amplio como puede ser el de una nave industrial de dimensiones considerables, hay varias soluciones que ofrece la tecnología actual. Por una parte, tenemos el WiFi, el más conocido por su uso

doméstico, aunque para otras aplicaciones también tenemos ZigBee, Bluetooth, o GSM. Con la llegada del 5G se quiere potenciar esta última ya que amplía considerablemente las capacidades que había hasta ahora con el 4G, como el ancho de banda, la latencia y la capacidad de conectar más dispositivos desde un mismo nodo [2].

Dependiendo del tipo de implementación y de la escala a la que se vaya a trabajar, se puede utilizar varias soluciones incluso de manera simultánea. A continuación, se valorarán las opciones más comunes por separado.

2.2.1 Wifi

Cisco define el Wifi como “una tecnología de redes inalámbricas que permite que dispositivos como computadoras (laptops y de escritorio), dispositivos móviles (teléfonos inteligentes y dispositivos portátiles) y otros equipos (impresoras y cámaras de video) interactúen con Internet” [3]. La tecnología de comunicación más utilizada tanto en ámbito doméstico, como empresarial, educativo o industrial. El 92% de los hogares de Europa tienen conexión a internet [4], y por tanto, en su mayoría emplearán el Wifi para conectar los distintos dispositivos que posean para su salida a internet.

Con los estándares actuales proporciona buenas velocidades y seguridad avanzada, si bien las interferencias con un entorno saturado y los obstáculos físicos son un gran impedimento a la hora de dar cobertura en algunos emplazamientos.

2.2.2 Zigbee

La tecnología Zigbee, definida por el CSA (Connectivity Standards Alliance) como una tecnología de comunicación de baja potencia, implementado por diferentes fabricantes, bajo el estándar IEEE 802.15.4 [5]. Es conocido por sus numerosas aplicaciones en domótica, pues la gran mayoría de estos productos, implementan esta tecnología, además de por las características antes mencionadas, para no saturar la red Wifi doméstica donde trabajan. Una de sus ventajas es que los dispositivos que tienen alimentación permanente funcionan como repetidores conformando una red mallada.

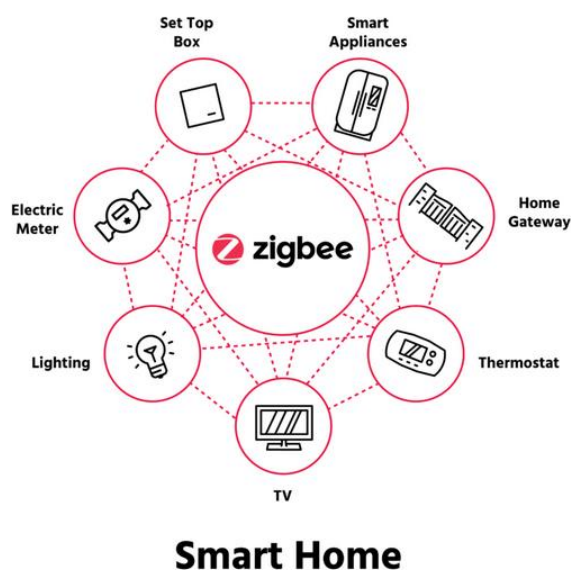


Ilustración 2. Diagrama Zigbee

2.2.3 Bluetooth

El Bluetooth “es una tecnología inalámbrica de corto alcance que permite la comunicación inalámbrica de datos entre dispositivos digitales, como un ordenador o una cámara digital. La tecnología inalámbrica BLUETOOTH funciona en un rango de aproximadamente 10 metros.” [6]

Comúnmente sirve para conectar 2 dispositivos, pero con las últimas revisiones, permite conexión simultánea entre varios dispositivos. También incluye otras funciones como BLE, que gasta menos energía en las comunicaciones, utilizado en dispositivos *wereables* y auriculares inalámbricos. Una de sus desventajas es el rango, que únicamente es de varios metros.

2.2.4 GSM

La tecnología GSM un estándar para la comunicación inalámbrica entre teléfonos móviles completamente digital. Esto permite que además de voz, puedan intercambiarse mensajes (SMS) o paquetes TCP/IP con una conexión a internet. [7]Comúnmente, la comunicación se realiza desde un dispositivo móvil equipado con una tarjeta SIM a una antena o repetidor, y de ahí viaja hasta alcanzar su objetivo, ya sea una web o la conexión para una llamada con otro dispositivo.

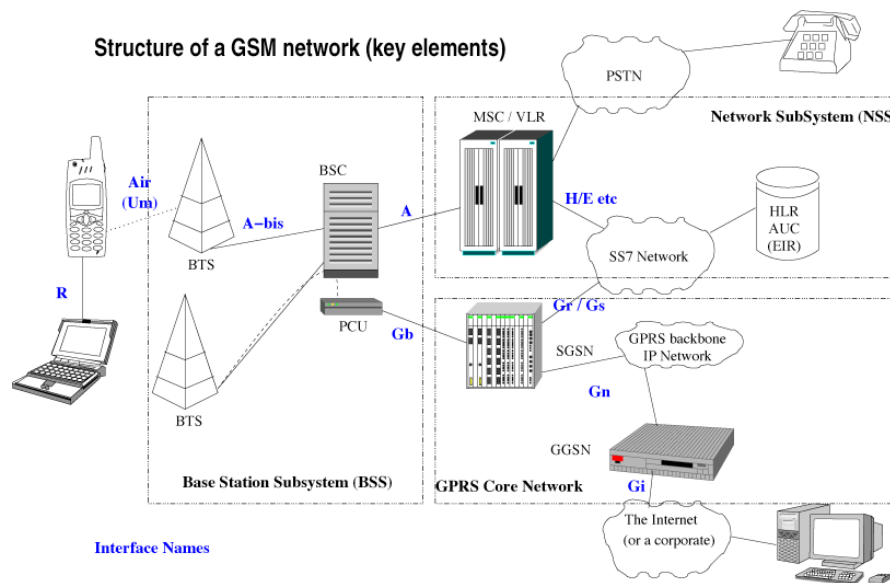


Ilustración 3. Esquema básico GSM

Su principal ventaja es que hay cobertura (en mayor o menor medida) en casi cualquier parte del territorio, en este caso nacional; mientras que, en otros territorios más remotos, no la hay, pues necesita de una infraestructura por detrás para que funcione correctamente.

2.3 Intercambio de mensajes.

En la transmisión e intercambio de paquetes se observa también una gran variedad de protocolos disponibles, desde los más conocidos como los WebSocket, IRC (usado antiguamente en chats y foros), MQTT (usado en domótica) y AMQP. Se van a analizar estos 2 últimos dado que son los más adecuados para la solución.

2.3.1 MQTT.

Es un protocolo creado por IBM en el año 1999 para el intercambio de mensajes en entornos de red con poco ancho de banda disponible. Su explosión de popularidad se vio con la llegada del IoT al público general [8]. Es un protocolo de comunicación bidireccional muy ligero y eficiente sobre el que se han implementado diferentes *brokers* como Mosquitto, el más usado para aplicaciones domóticas ya que cuenta con una integración para Home Assistant.



Ilustración 4. Logo Mosquitto

2.3.2 AMQP.

En este caso, creado por JPMorgan, mejora la interoperabilidad entre sistemas y las características de MQTT. Está pensado para un uso más generalizado, y sus mejoras se centran en la seguridad y en la transacción de los mensajes, implementando un QoS [8]. Estas mejoras hacen que sus principales usuarios e interesados sean el sector bancario y empresas de telecomunicaciones. Como MQTT, está basado en publicadores y consumidores de mensajes, que leen de las colas que se crean en un bróker, quien gestiona las conexiones y los mensajes. Hay dos versiones actualmente, la 0-9-1, utilizada por RabbitMQ y la 1.0 que emerge de OASIS.



Ilustración 5. Logo RabbitMQ

2.4 Bases de datos.

En cuanto a bases de datos, se observan 2 vertientes mayoritarias en el ámbito de uso informático. Encontramos bases de datos Relacionales de tipo SQL y No-SQL.

2.4.1 SQL.

Aunque en realidad SQL no es el tipo de base de datos (que en este caso sería relacional), es el lenguaje de base de datos creado por IBM que nutre la mayoría de las bases de datos utilizadas hoy en día [9]. Se basan principalmente en órdenes, ya sean de consulta, inserción, actualización o eliminación; las cuales se componen de unas expresiones o acciones, que se realizarán en caso de cumplirse unas condiciones evaluables concretas. Hoy en día ha sido desarrollado por diferentes empresas como Oracle (SQL Oracle y MySQL), Microsoft (Microsoft SQL) y otros. Su uso tan extendido aporta una gran comunidad de usuarios.

2.4.2 No-SQL.

Empezaron a ser populares a principios del siglo XXI gracias a la proliferación de la Web 2.0, las grandes empresas tecnológicas como Amazon, Facebook, Google y Netflix requerían una escalabilidad y una fiabilidad que no encontraban con las clásicas bases de datos SQL [10].

También encontramos diferentes visiones e implementaciones para este tipo de base de datos, siendo las más conocidas en este caso MongoDB (base de datos Documental con esquema dinámico), Cassandra (descentralizada, escalable y con alta tolerancia a fallos) e InterSystems Caché (estructura en forma de árbol inverso).

2.5 Soluciones ya existentes

Los gigantes del software empresarial como son Sage, Oracle y SAP tienen software adaptado e interconectado con sus ERP's que nutren de esta funcionalidad a las empresas de hoy en día.

Sage por ejemplo comercializa SAGE X3. El software del gigante británico destaca por su gran interoperabilidad, siendo compatible con sistemas clásicos en los cuales se lleva un control de stock a papel y también con modernos sistemas de radiofrecuencia en los cuales, los productos actualizan su estado gracias a emisores que se encuentran en las cajas. Además, al estar conectado al ERP de SAGE, permite que cuando un producto se lleve a una zona de carga para su salida del almacén, se activen procesos para la generación y envío de documentación al cliente (Albaranes, facturas y seguimientos).

Por parte de SAP tenemos el Extended Warehouse Management integrables con SAP S/4HANA. Aboga por su rapidez y agilidad a la hora de mostrar y manejar datos. Se centra más en optimizar la cadena de suministro, optimizar stocks y pedidos de proveedores gracias a las métricas internas que calcula. Se centra en promocionar su apoyo a la sostenibilidad de los clientes que lo integran al mejorar sus procesos.

Por último, Oracle comercializa el Oracle Cloud Inventory Management. Mucho más centrado en la gestión del stock de los almacenes y dejando a otros servicios interconectados características que sí integran sus competidores como la gestión de la documentación o la optimización de la cadena de suministro. Se hace destacar por la agilidad que aporta a la hora de llevar el control de stock y servir los pedidos a tiempo. Se declaran ganadores del cuadrante mágico de Gartner de 2023 en sistemas de gestión de almacenes.

3. Análisis del problema

A continuación, se va a realizar un análisis del problema que se nos ha planteado, la situación, la casuística y los puntos clave analizados anteriormente en el estado del arte nos servirán de guía para decidir qué tecnologías serán las más adecuadas para cada ámbito.

El propósito general es dar un soporte a la localización en una instalación industrial de dimensiones considerables. Necesitaremos captar la información desde diferentes puntos, enviarla a un servidor que almacene estos datos para que posteriormente pueda ser tratada de manera óptima.

Para no mezclar los diferentes ámbitos que nos podemos encontrar a la hora de plantear la solución, se va a diferenciar entre problema físico y tecnológico. Además, al ser esta una solución alternativa a opciones que ya se comercializan por grandes tecnológicas del sector, se añadirá un análisis del coste de la solución.

3.1 Problema físico.

El emplazamiento piloto de la solución es una nave industrial de industria papelera ubicada en la localidad de Favareta. La extensión es de aproximadamente 5.000 metros cuadrados, siendo el lado más largo de la parcela de 100 metros y el más corto de 50m (tamaño algo inferior a un campo de fútbol).

El edificio de oficinas e instalaciones está en una de las esquinas, por tanto, podemos encontrarnos con distancias que salvar bastante grandes. Teniendo en cuenta las dimensiones y que nos encontramos en una instalación interior, algunas de las alternativas planteadas para la comunicación, quedan descartadas. Una conexión GSM, pese a ser fiable y darnos conexión en amplias áreas de territorio, puede mostrar comportamientos anómalos bajo estructuras metálicas, que podrían tener efecto de "caja de faraday". Por otro lado, las distancias y los posibles obstáculos interiores hacen que el Bluetooth, pese a su bajo consumo de energía, tampoco pueda ser apto para la solución.

En caso del Zigbee, pese a ser una tecnología usada en interiores en soluciones domóticas, su corto alcance entre nodos nos haría depender de que hubiera equipos conectados de manera permanente a la red eléctrica en unas zonas específicas para que estas actuaran de repetidores (Zigbee soporta una tipología MESH).

Finalmente, el Wifi, pese a tener impedimentos con obstáculos, al poder apoyarse en una estructura cableada para distribuir puntos de acceso por toda la instalación, daría una cobertura adecuada a todos los dispositivos que se empleen, sin que importe su ubicación y su movimiento.

3.2 Problema tecnológico.

No se encuentra ninguna trazabilidad automática del producto actualmente en las instalaciones. Cuando se descarga un pedido, el operario encargado registra manualmente en el ERP las partidas de compra que se han recepcionado, esto provoca que la tarea se alargue de manera inevitable, pues tiene que ir reconociendo cada código de cada partida.

Ocurre lo mismo cuando la mercancía se tiene que enviar. Pese al etiquetado con lotes, se desconoce la composición de estos, así como el descontado de producto de stock, que también se realiza a mano. Este proceso también se realiza a la vez que la operación de carga del transportista.

Los operarios del almacén lo ordenan bajo criterio propio, así pues, pese a haber una zona de entrada y salida, las localizaciones de producto, ya sea en bruto, proceso o listo para la salida, no tienen una posición definida. Cuando la mercancía debe moverse por la extensión del almacén, son ellos mismos los que la reubican cuando sale de las máquinas que realizan las tareas.

3.3 Seguridad:

Dentro del sistema que se va a crear, no hay información sensible de la empresa. Únicamente se almacena información de los productos y sus ubicaciones dentro del almacén. Esto no es motivo para no proteger la información que contiene la base de datos y se intercambia con softwares externos. Además, la no securización de este nuevo sistema de información dentro del ámbito de la empresa, entraría en conflicto con normativas como la ISO 27001.

3.3.1 Comunicaciones:

Todas las comunicaciones entre los diferentes dispositivos se realizan mediante Wifi. Estas redes que se creen deberán estar securizadas con protocolos actuales (WPA2-WPA3). Además de estar separadas de las redes Wifi ya existentes y que se utilicen para otras aplicaciones o para los empleados. Es recomendable que los nombres de estas redes se oculten para que sea más difícil su ataque.

Además, la comunicación de los puntos de acceso al router principal o switches de interconexión se realizará por una VLAN distinta para separar el tráfico que pueda venir de estos puntos y no se mezcle con el tráfico entre equipos de la fábrica.

3.3.2 Base de datos:

La base de datos deberá tener una copia de seguridad diaria fuera del sistema, ya sea en un NAS u otro servidor de la empresa. Esta copia diaria, para ofrecer economía de espacio se debería configurar como incremental. Es recomendable que haya otra que se haga fuera de la empresa, en la nube o un servidor externo a ella.

Para mayor seguridad, se debería configurar una copia de seguridad completa y externa a la empresa que se haga de manera semanal. Esta deberá ir fuera de la empresa directamente.

En el diagrama a continuación se puede ver que hay una duplicidad de las copias externas. Esto es recomendable en caso de que la información sea sensible y una de las copias de seguridad falle, ya sea por problemas con la propia copia o con el proveedor del servicio.

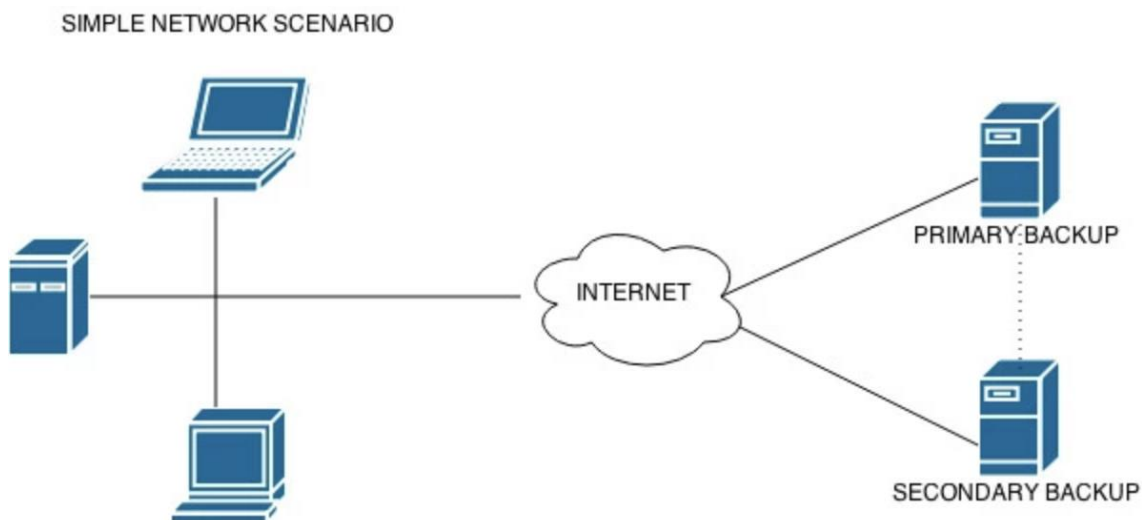


Ilustración 6: Esquema de seguridad

Además, los accesos a la base de datos que se hagan deben ser independientes. Tanto para la parte de inserciones de información como para las consultas. Esto aporta que en caso de que la contraseña del usuario sea descubierta o revelada, siempre mantengamos parte de la integridad de la información a salvo.

3.3.3 Integridad de datos:

Con el fin de que los datos que se almacenan en la base de datos sean confiables y no provoquen un error o equivocación a la hora de su utilización y consulta, se van a implementar mecanismos para asegurar la integridad de los datos tanto de manera física como lógica.

En el apartado de la integridad física, como se ha comentado en el apartado anterior, se van a realizar copias de seguridad tanto en el emplazamiento principal como en la nube o en un emplazamiento secundario o externo. De esta manera, ante un problema en la base de datos original, se podría rescatar la primera copia de seguridad. En caso de un desastre en la ubicación principal, ya sea una inundación, incendio o fallo eléctrico, habría una segunda copia de seguridad en la ubicación externa separada o en la nube que se podría recuperar y poner en marcha.

Para que la copia de seguridad sea eficiente, no se hará una copia completa cada noche de la base de datos. Una política que equilibre esta integridad de datos con mover un volumen de datos asumible y que no suba el coste (de espacio y de tráfico en algunos servicios de almacenamiento) sería una copia completa cada semana. De manera diaria sí se haría una copia incremental-diferencial (sólo se subirían los cambios desde la última copia y no una imagen entera de la base de datos). Adicionalmente, para la ubicación externa se subirían las copias con una frecuencia menor, pudiendo subir las copias incrementales 2 veces a la semana y una completa a final de la semana.

Estas 2 copias de seguridad (tanto la principal como la externa) se pueden configurar con software de este estilo como Cobian Backup o con utilidades que integren los servicios de copias de seguridad (NAS de Synology o de QNAP cuentan con este tipo de utilidades y sincronizaciones con equipos externos).

Para una confiabilidad más extrema, se podría contar con una tercera copia, la cual se haría con una frecuencia mucho menor (mensual) como última opción en caso del fallo de la base de datos funcionando, y las 2 copias ya realizadas.

En el apartado de la integridad lógica, para asegurar que los registros tienen una trazabilidad correcta, podemos definir unas directrices:

- Todos los productos, cuando hayan finalizado su estancia en el almacén deberán tener como primer y último registro una zona de entrada/salida o una máquina de transformación. Esto es porque un producto en bruto que entre en el almacén entrará por una zona de E/S, y su último registro será una máquina de transformación. En el caso de un producto ya transformado, su primer registro corresponderá a la máquina, y su salida del almacén a una zona de E/S.
- Cualquier registro nuevo de la base de datos no podrá tener una marca temporal anterior al último registro existente.
- La clave primaria de la tabla de registros de localización será la tupla del ID del producto + la marca temporal.

3.4 Presupuesto:

Para dar solución a este problema se necesita de un soporte físico que se detalla por categorías.

Estas categorías separan la infraestructura para el montaje de la red de comunicaciones, por un lado, y los útiles que llevarán los operarios para llevar la trazabilidad de los productos por otro.

Infraestructura:

- Servidor para almacenamiento de BBDD y bróker comunicación. Raspberry Pi 4.
- Puntos de acceso Wi-Fi.
- Cableado e interconexión.

Herramientas para personal:

- Raspberry Pi-Zero W (o similar).
- Dispositivo lector de códigos de barras o QR.
- Baterías y carcasa.

Con el estado inflacionista actual, algunos de los artículos comentados anteriormente, sufren de una varianza de precios diaria constante y, por tanto, se va a tomar el precio MSRP que se anunció en su lanzamiento. Otros productos, como la carcasa y la batería se van a poner como referencia ya que su tipología o características puede variar en caso de las necesidades particulares.

Concepto	Precio
Raspberry Pi 4 4GB	50€
Ubiquiti U6-LR (x4)	840€ (210€ x 4)
Bobina cable FTP Cat6 305m	170€
TP-Link TL-SG1005P, 5 puertos con PoE	50€
Raspberry Pi-Zero W (x10)	100€ (10€ x 10)
Lector QR (x10)	200€ (20€ x 10)
Batería + carcasa (x10)	150€ (15€ x 10)
Total	1560€

No se suma el coste de licencia, ya que, tanto para la base de datos, como para el sistema operativo del servidor, hay opciones totalmente válidas que son de uso libre. Además, habría que sumar las horas de personal para el montaje de los aparatos y su testeo/formación.

Una solución comercial de este tipo, con lo que se conoce como pack en mano, multiplicaría el coste en 10 o 15 veces el presupuesto que se ha comentado anteriormente, y teniendo en cuenta la aplicación sobre una pyme, sería difícil su incorporación.

4. Modelo de localización

El problema de cómo organizar un almacén (SLAP por sus siglas en inglés) lleva siendo foco de diferentes algoritmos de ordenación desde que en 1976 Hausman, Schwarz y Graves introdujeron una taxonomía acertada sobre este problema. Esto provocó que, a partir de ese momento, se definieran distintas vías de exploración a la hora de adoptar una organización concreta. Hoy en día podemos ver que hay almacenes en los que se organizan por el tipo de producto, por la clase de producto, o que directamente se decide aleatorizar la ubicación de los productos [11]. Ésta última práctica es bastante famosa en los almacenes de Amazon, donde los productos solamente siguen algunas pequeñas reglas que definen su almacenamiento (por ejemplo, bultos voluminosos y pesados en zonas inferiores).

Aunque podamos pensar que esta última idea no tiene lógica alguna, (pues un almacén debe estar organizado) Amazon se escuda en que su aleatorización de la ubicación le aporta diversas ventajas:

- El espacio puede ser mejor aprovechado, ya que no hace falta que un bulto grande esté con otro bulto grande. Pueden llenar el hueco adyacente con otros productos más pequeños.
- Provoca menos errores humanos y la curva de aprendizaje es mejor; ya que es el sistema que organiza el almacén el que le dice al empleado dónde se encuentra el producto que tiene que recoger o dónde lo tiene que dejar.

En un almacén totalmente automatizado, que tenga vehículos autónomos inteligentes, y un gran volumen de productos, podremos llegar a un nivel de aleatorización en la ubicación de los productos, que, en caso de que el almacén sea totalmente llevado de manera totalmente manual por empleados, sería imposible. Evidentemente, todas estas tipologías de organización se deben particularizar cada caso, ya que ninguna será 100% correcta para el propósito final. Esta particularización u optimización harán que, una vez adoptado el modelo de ubicación, se pueda explotar con la mayor eficiencia y eficacia posible.

4.1 Modelos aleatorios para almacenes automatizados.

Dentro de los almacenes automatizados se pueden observar distintas etapas de automatización donde lo más básico que encontramos serían las zonas de *picking* de Amazon, donde el sistema tiene conocimiento de dónde se ubica cada elemento, pero es un operario el que tiene que desplazarse hasta la ubicación concreta y recoger el producto, ya sea manualmente o mediante algún vehículo industrial.

En el caso de automatizaciones más fuertes, como las que podemos ver en un almacén de productos farmacéuticos o hasta en una simple farmacia el camino que siguen los productos es diferente. Los empleados indican el pedido que hay que servir y las mercancías se dispensan automáticamente desde carriles o tolvas a una cinta que va

recorriendo el sistema y deposita toda la carga en una zona donde el empleado ya la tiene junta. Aquí el sistema es encargado de saber dónde se encuentra cada producto y de recogerlo para servirlo. En la mayoría de estos casos, mientras que la salida de productos o mercancía es automática o prácticamente automática, la entrada suele ser manual, y suelen ser los operarios los que rellenan las distintas zonas.

En un paso mayor de automatización nos podríamos encontrar situaciones como las que se dan en una fábrica de automóviles, donde los productos, mediante sistemas de robots autónomos que se desplazan por diversos medios, son capaces de recoger y llevar materiales a las distintas zonas según las necesidades. Cuando faltan unidades de cualquier producto, para seguir fabricando, el robot autónomamente es capaz de ir a la zona determinada donde se almacena ese producto, lo recoge y lo lleva a la cadena de montaje donde se requiere. [12]

Analizando el caso de Amazon, donde únicamente se aleatorizan y se automatizan las ubicaciones, podemos ver por imágenes, en un mismo estante, se almacenan productos que no tienen nada que ver uno con otro.



Ilustración 7: Estantería en Amazon

Podemos ver una máquina cortapelo justo debajo de unos puzzles, en el estante de al lado unos libros, y justo debajo una garrafa con líquido rojizo. Para el ojo humano puede parecer totalmente caótico pero el sistema que gestiona el almacén sabe dónde está cada elemento y no tendrá problema alguno en localizarlo en su estante concreto. Esto también evita equivocaciones, ya que, si hay 2 elementos muy parecidos en estantes contiguos, una confusión por parte del operario provocaría que el elemento recogido no fuera el adecuado, en cambio, con 2 elementos distintos, este error se reduce al mínimo.

Otro tipo de almacenes más automatizados que se pueden encontrar son los de tipo carrusel. En ellos, hay un bucle cerrado de cajas en constante movimiento, que eventualmente se abre para dejar salir alguna caja del circuito. En estas cajas encontramos divisiones internas, que contendrán distintos elementos. Más tarde, será un operario quien con una ayuda lumínica sabrá qué tiene que coger. Esto se consigue porque el software que gestiona el almacén tiene conocimiento de qué elemento hay en

cada división del cajón, y cuántas unidades se almacenan. El operario únicamente tiene que coger las unidades que el software indica. Las cajas le llegan mediante una cinta y se vuelven al carrusel junto al resto de cajas.



Ilustración 8: Sistema carrusel Fedefarma

En este caso, el sistema está desarrollado por SSI Schaefer y el sistema implementado se denomina Schaefer Carousel System (SCS). Con este sistema aseguran que “la nueva logística de distribución de la instalación de Fedefarma, de casi 8.000 metros cuadrados, es capaz de almacenar 40.000 referencias y atender 125.000 líneas diarias con sólo 55 operarios, además de suministrar a su red farmacéutica 4 veces por día.” [13]

4.2 Modelos clásicos.

Dentro de un almacén clásico vamos a encontrar siempre una estructura bastante similar diferenciada en 2 o 3 zonas, que pueden ser las de entrada/salida y la de almacenaje puro.

Las de entrada y salida suelen ser zonas más pequeñas, donde la mercancía es descargada y preparada para su envío. Aquí no se siguen criterios de ordenación por tipo de producto o características. Suelen ordenarse dependiendo de la salida que vayan a tener (cliente o transporte) y las entradas según se vayan descargando.

En la zona de almacenaje es mucho más cotidiano encontrar ordenaciones que sean lógicas para el razonamiento humano. Esto hace que los operarios del almacén tengan facilidades para saber dónde se encuentran los productos que están buscando prácticamente de memoria. Siguiendo con el ejemplo de los tornillos, es lógico que todos los tornillos se almacenen en el mismo pasillo, clasificados por sus medidas y métricas, pero agrupados. No tendría sentido para la lógica y la “facilidad” de saber dónde está algo de memoria que estén los de métrica 2 y 3 en una zona del almacén y los de métrica 5 y 7 en otro extremo del almacén.



Ilustración 9: Almacén clásico. Tipos de áridos similares almacenados de manera contigua

En caso de un pedido, esto ayuda a los operarios a que, si tienen que coger distintas variantes de un mismo producto, las van a encontrar bastante agrupadas y no tendrán que recorrer grandes distancias. Además, en caso de ser un pedido de grandes dimensiones, varios operarios pueden ir simultáneamente recogiendo mercancías contiguas. [14]

4.3 Tabla comparativa:

A continuación, se comparan los puntos fuertes y débiles de ambas opciones haciendo hincapié en las diferencias.

Modelo automático caótico		Modelo clásico	
Pros	Contras	Pros	Contras
	Difícil localización	Facilita localización	
	Difícil control stock	Fácil control stock	
	Requiere comunicación o sistema apoyo	Fácil entrada/salida	
Aprovechamiento espacial			Espacio menos aprovechado
Mayor Flexibilidad			Menos flexibilidad

Como se puede observar, ambas opciones son totalmente complementarias, ya que los puntos débiles de una se ven reforzados en la otra.

Mientras que un modelo es mucho más intuitivo para el operario, el otro aprovecha mejor el espacio empresarial mientras que requiere una mayor carga tecnológica e inversión por parte de la empresa para implementar un modelo a priori más complejo pero que

Sistema de soporte a la localización de recursos en entornos industriales evolucionando hacia a un mundo digital. Una solución aplicada

ayuda a sacar más rendimiento a los recursos de la empresa potenciando una carga tecnológica asumible en esta época.



5. Diseño de la solución

En este capítulo se identificarán los distintos bloques que componen el sistema, profundizando en su funcionamiento y en la forma en la que se comunican entre ellos. Además, se seleccionarán las tecnologías que se emplearán para su implementación.

5.1 Arquitectura general del sistema.

La arquitectura para la solución y su funcionamiento será similar a la de un cliente-servidor, ya que los dispositivos que llevarán los empleados lanzarán datos contra el servidor, y en caso de querer realizar alguna consulta, ésta se realizará desde un puesto cliente también contra el servidor y la base de datos.

La particularidad, es que, para separar los flujos de información, dependiendo de su origen o su destino, se va a implementar sobre un bróker de comunicación.

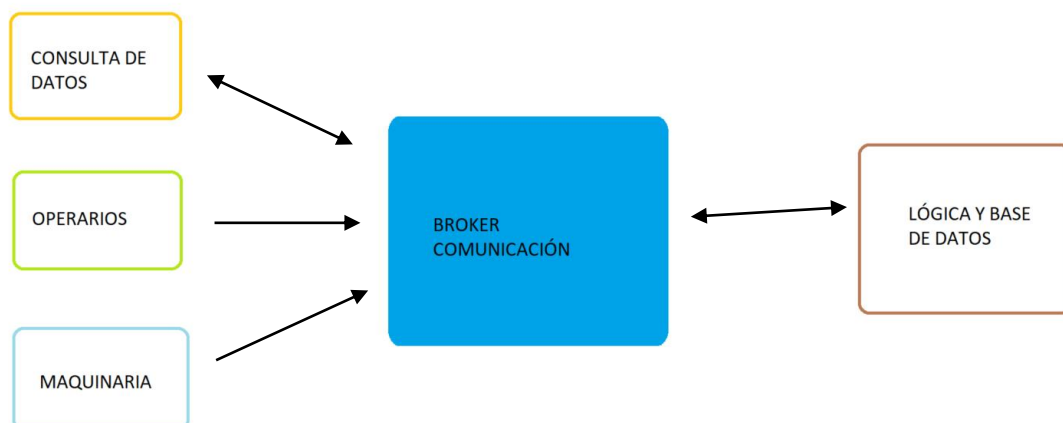


Ilustración 10: Diagrama de arquitectura

Por un lado, habrá 2 tipos de dispositivos finales, pudiendo diferenciarlos por el tipo de comunicación (unidireccional vs bidireccional) mientras que, en todo momento, el bróker es el que mantiene la comunicación bidireccional con la lógica y la base de datos.

Adicionalmente, para la conexión del sistema con un ERP o servicio externo, se habilitará un servicio mediante API REST para consultar la última posición de un producto.

5.2 Arquitectura detallada del sistema.

En este apartado se describirá de manera detallada el funcionamiento de cada componente, así como su tecnología escogida para cada uso.

5.2.1 Estructura de tablas SQL

Para dar soporte a la información se va a generar una estructura de tablas en la cual se almacenará el registro de localizaciones para los productos. Adicionalmente se va a generar una tabla de ocupación, que permitirá un control sobre la ocupación de los espacios y los que están libres para su próxima ocupación.

Tabla de Localizaciones:

Campos: *id*, *timestamp*, *location*, *worker*.

Claves primarias: *id*, *timestamp*.

Claves ajenas: *location* (tabla Ocupación)

Restricciones: *Not Null* para los 4 campos de la tabla de localizaciones.

Tabla de Ocupación:

Campos: *Location*, *occupied*.

Claves primarias: *Location*.

Restricciones: *Not Null* para los 2 campos de la tabla de Ocupación. El campo *occupied* es booleano.

Se define la clave ajena para asegurar que no hay 2 productos en la misma localización (se trabaja por palets enteros).

Adicionalmente, se define un índice a parte del creado automáticamente que trabaja con la tupla *id* y *timestamp* para que las búsquedas sean más ágiles.

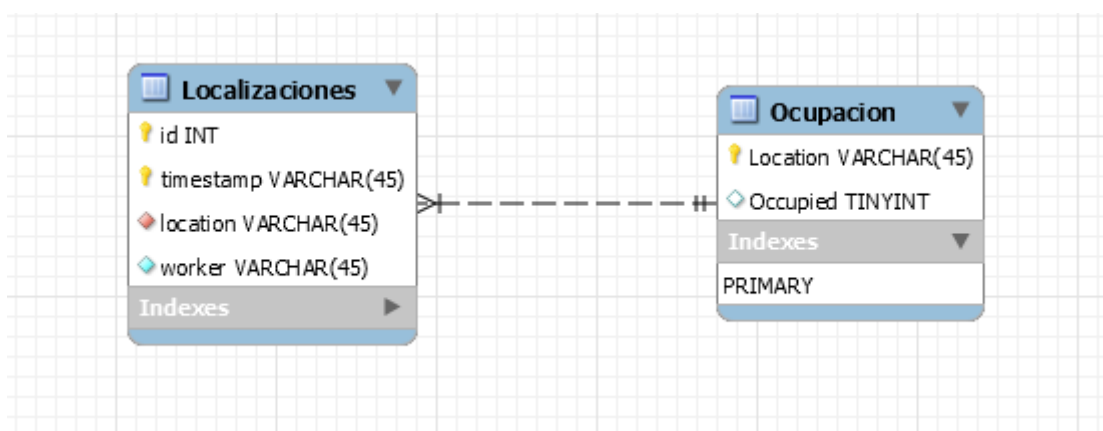


Ilustración 11: Diagrama de esquema tablas

5.2.2 Casos de uso.

En esta sección se van a definir los casos de uso del sistema, tanto de trabajo como de consulta de la información.

ID	1
Caso de uso	Registro de nueva ubicación
Propósito	Registrar la nueva ubicación de un producto en el sistema
Descripción	<ol style="list-style-type: none"> 1- Recibir ubicación libre 2- Liberar última localización del producto 3- Registrar nueva localización con marca temporal 4- Ocupar nueva localización 5-
Postcondiciones	Se crea un nuevo registro temporal de la ubicación del producto y ésta se marca como ocupada
Frecuencia esperada	Alta en caso de que haya muchos movimientos de producto en el almacén

ID	2
Caso de uso	Consultar última localización
Propósito	Consultar la última localización de un producto del almacén
Descripción	<ol style="list-style-type: none"> 1- Recibir el ID del producto 2- Consultar la última localización disponible del producto en la tabla de localizaciones 3- Devolver registro
Postcondiciones	Se devuelve un registro el cual contiene la última localización del producto solicitado, así como su marca temporal
Frecuencia esperada	Media, sólo utilizado como consulta para encontrar un producto.

ID	3
Caso de uso	Consultar histórico de localizaciones
Propósito	Consultar el registro histórico de localizaciones de un producto del almacén
Descripción	<ol style="list-style-type: none"> 1- Recibir el ID del producto 2- Consultar las localizaciones del producto dado un ID. 3- Devolver registros ordenados por tiempo
Postcondiciones	Se devuelve una serie de registros el cual contiene la localización del producto solicitado, así como su marca temporal ordenados de más recientes a más antiguos
Frecuencia esperada	Baja, sólo utilizado como consulta para conocer el histórico de un producto concreto.



5.2.3 Colas de mensajería.

El sistema va a emplear las colas de RabbitMQ para el intercambio de mensajes entre los terminales de los operarios y el bróker, así como entre el dispositivo de consultas y el bróker.

Con el fin de simplificar el funcionamiento y posteriores tareas de logging y consulta, se van a crear 3 colas separadas para distintas funciones.

- Cola para los mensajes de los trabajadores y máquinas
- Cola para entradas y salidas de material.
- Cola para consulta de información.

Ya que la gran mayoría de terminales pueden ir moviéndose por toda la extensión de la instalación, el sistema empleará conexiones no persistentes y puntuales para hacer cada envío, así en la transición de la conexión entre nodos Wifi, evitaremos la pérdida de conexión.

Esta repetición de tarea cada vez que se quiere enviar un mensaje para conectar al bróker y declarar la cola no es problema ya que solamente se creará si no existe.

Una vez el mensaje se haya enviado, la conexión se cerrará a la espera de que el operario ejecute de nuevo la acción y emita de nuevo la información de ubicación del producto en cuestión.

La única diferencia en las colas será que en la de obtención de información, la comunicación será de manera bidireccional, ya que el terminal de consulta realizará la petición a través de la cola y el propio backend del sistema le responderá por la misma cola. En el resto de los casos, la información será unidireccional de los terminales al backend.

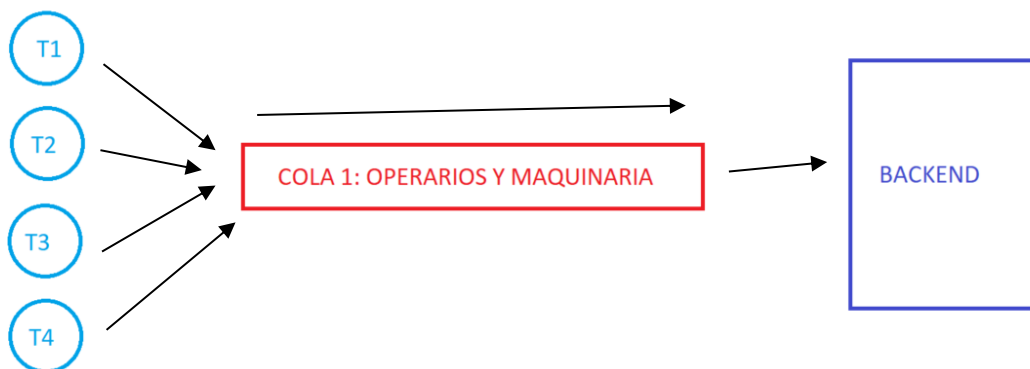


Ilustración 12: Ejemplo tráfico para cola de operarios

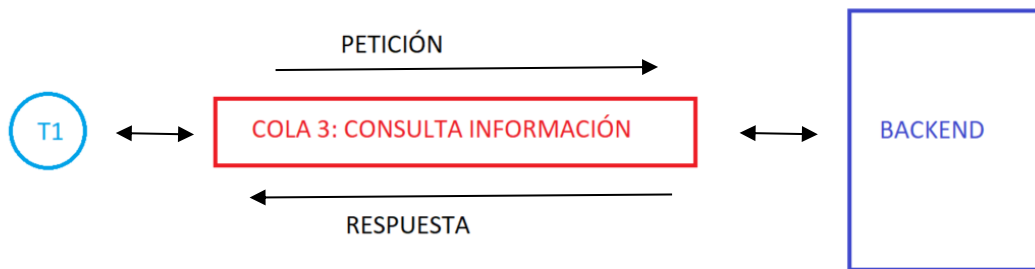


Ilustración 13: Ejemplo tráfico para consultas

En este último caso, el terminal, cuando realice la petición de la información lanzará un proceso que se pondrá en escucha sobre la cola de consulta para recibir la información proveniente de la base de datos.

5.2.4 Consulta de datos

La consulta de datos se basará en una serie de registros que se recuperarán según el código que tenga la etiqueta del producto. Estos datos se presentarán según el orden temporal de los eventos.

En el momento de hacer la consulta, en el cual lo que se enviará es el código que tenga el producto, el terminal de consulta se pondrá a la espera escuchando sobre la misma cola en la que ha hecho la petición para poder recibir la respuesta.

Esta respuesta estará compuesta por tantos registros como eventos de movimiento haya sufrido la referencia consultada ordenada de manera temporal de más reciente a más antigua.

Las tuplas de información serán Fecha-Ubicación-Operario.

```
SELECT TIMESTAMP, LOCATION, WORKER FROM dbo.Localizaciones WHERE
ID='EJEMPLO' ORDER BY TIMESTAMP DESC
```

5.2.5 Consulta de datos desde fuentes externas.

Con el fin de dotar que la información del sistema pueda ser consultada desde otras fuentes o plataformas (ERP de la empresa) se va a crear un servicio para que mediante el consumo de una API REST se pase como valor el identificador del producto y se devuelva la última posición que se tenga de este identificador.

Adicionalmente, se creará también un método para consultar el histórico de posiciones, siendo estas tuplas de la marca temporal con la posición en ese instante y qué operario lo movió.

Ambas consultas se desarrollarán dentro de la misma API REST y se podrá llamar a una u otra dependiendo del método a consultar.

Para simplificar la creación y el mantenimiento de esta API REST se va a utilizar Spring Framework.

```
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.RestController;

@RestController

public class ApiController {

    private ConsultaUltimaPosicion consultaUltimaPosicion = new
ConsultaUltimaPosicion();

    private ConsultaHistorico consultaHistorico = new ConsultaHistorico();

    @GetMapping("/api/consulta")
    public String consultarUltimaPosicion(@RequestParam String id) {
        String resultado = consultaUltimaPosicion.consultar(id);
        return resultado;
    }

    @GetMapping("/api/historico")
    public String consultarHistorico(@RequestParam String id) {
        String resultadoHistorico = consultaHistorico.consultar(id);
        return resultadoHistorico;
    }
}
```

Ilustración 14: API REST

El método `consultaUltimaPosicion.consultar` se apoyará en la misma query que se ha definido para el terminal de consulta, la cual obtiene la última localización de un producto concreto.

El método `consultaHistorico.consultar` se apoyará en una consulta que devuelva todas las filas que hay para un registro en concreto ordenadas por su marca temporal.

```
SELECT TIMESTAMP, LOCATION, WORKER FROM dbo.Localizaciones WHERE
ID='EJEMPLO' ORDER BY TIMESTAMP DESC
```

5.2.6 Terminales de operarios.

Los terminales de operarios presentan una única acción una vez iniciados para publicar la información del producto que se va a transportar en ese momento. Estos terminales transmitirán la información del código del producto, la posición donde se encuentran actualmente y quién ha realizado la acción. Como se ha comentado anteriormente, se incorpora también una marca de tiempo para saber en qué momento se ha realizado cada acción, que se calculará en el momento de grabar en la base de datos para prevenir diferencias de tiempo entre los dispositivos y la comunicación con el bróker.

La entrada del operario se realizaría al inicio de la jornada y se mantendría para todas las operaciones de ese operario. En caso de cambio, se asignaría el nombre de otro operario para que, a partir de ese momento, los movimientos de los productos estuvieran referenciados al nuevo empleado.

Para los datos de identificación del recurso, se pueden emplear 2 alternativas. Bien se pueden dotar a los distintos productos de códigos de barras o QR. En caso de ser un código de barras no nos hace falta irnos a tipos demasiado complejos ya que únicamente vamos a representar un identificador alfanumérico.

Para la identificación de las zonas que hay delimitadas en el almacén, tanto de máquinas como en el almacenaje o entrada y salida de material, también se pueden incluir códigos de este tipo en el suelo para que sean leídos y proporcionar la ubicación



Ilustración 15: Representación en código de barras y QR de un ID concreto

Para poder leer estos códigos se pueden emplear 2 dispositivos distintos. Por un lado, se pueden utilizar lectores específicos de QR o códigos de barras. Estos llevan un software interno en el propio lector que ya devuelve el código interpretado y lo muestra como texto plano, aunque, por el contrario, no suelen leer distintos tipos de códigos a la vez y hay que programarlos para que lean un tipo u otro. Estos dispositivos suelen ir conectados vía USB a los equipos que los controlan y mediante un botón o una orden de software, activan haz lector para interpretar el código en cuestión.

Para el uso de estos dispositivos necesitaremos tener los drivers del dispositivo para conocer la forma de acceso a los datos que nos proveen.

Sistema de soporte a la localización de recursos en entornos industriales evolucionando hacia a un mundo digital. Una solución aplicada



Ilustración 16: Tipos de pistolas lectoras de códigos de barra o QR

En caso de que no se pueda o no se quiera utilizar estos dispositivos, se pueden realizar también entradas manuales por teclado. Aunque para la operativa sería más manual y lenta, nos da mayor compatibilidad y menor complejidad a la hora de implementar dispositivos de terceros.

En este caso, no sería operativo llevar en el terminal un teclado de formato completo o reducido. Para el uso que se le quiere dar, nos tendremos que ajustar a un teclado únicamente numérico que incluya alguna tecla extra de carácter alfanumérico para poder denominar las zonas que se utilizarán.



Ilustración 17: Teclado numérico

En caso de este tipo de teclados, se podrían sustituir algunas fundas de teclas para que tuvieran otros usos (y mapear esas teclas para que realicen una función diferente a nivel de software). En otro caso, se podrían utilizar teclados programables donde se le puede asignar una acción concreta o una macro de acciones a cada tecla para que se realicen una serie de acciones de manera secuencial. Asimismo, este último caso conllevaría un coste extra de los dispositivos y de su programación.

5.2.7 Modelo de localización 1:

Uno de los modelos de localización que podemos emplear, dado que la localización en la que se ha utilizado como referencia para este proyecto encontramos una planta rectangular. Esta forma ayuda a que podamos ubicar los terminales gracias a beacons, ya sean Wifi o BLE (Bluetooth Low Energy)

Por un lado, con el Wifi que ya tenemos para la comunicación de los mensajes, nos encontramos que podemos gastar esta infraestructura para hacer un mapa de calor de toda la nave, teniendo como referencia la potencia de la señal que nos llegue de cada Punto de Acceso. Combinando las diferentes potencias, se puede calcular en qué zona concreta del almacén nos encontramos. El problema es que en el caso de que haya obstáculos o el terminal se encuentre apantallado por algún objeto, las señales se pueden alterar y no nos daría una lectura fiel, y por tanto una ubicación incorrecta.

Con los beacons Wifi tendríamos un montaje como el de la imagen:



Ilustración 18: Diagrama de disposición de AP Wifi para posicionamiento

Siendo el AP1 el superior izquierdo, el AP2 el superior derecho, AP3 el inferior izquierdo y AP4 el inferior derecho; en los diferentes puntos tendríamos las siguientes intensidades de señal (dBm):

- Punto rojo: 50/50/50/50
- Punto Azul: 30/70/20/40
- Punto Morado: 70/10/70/10

En el caso de montaje con Beacons BLE el alcance es menor y tendríamos que suplir esta falta de alcance con más unidades distribuidas por la localización:



Ilustración 19: Diagrama beacons BLE para posicionamiento

Aplicando un algoritmo de limpieza y postproceso de las señales [15], se puede extraer la posición real del terminal en el momento de enviar el mensaje para saber la ubicación de la mercancía marcada.

Una vez resuelta la posición real (física) de la mercancía, se convertiría esas potencias de señal en una ubicación sectorizada y humanamente entendible para los operarios.

La posición calculada se mostraría solo en caso de una presentación ya fuera mediante el terminal de consulta o la API REST, en la base de datos sólo se almacenaría la potencia de las 4 (o más) señales de las balizas Wifi para no entorpecer en cada entrada el cálculo de la posición.

5.2.8 Modelo de localización 2:

En este modelo de localización se encuentra el almacén sectorizado y clasificado por zonas. La dinámica de ubicación sigue siendo la misma pero la posición ya no se calcula. Encontramos una cuadrícula para almacenaje, así como las zonas de entrada y salida y las máquinas.



Ilustración 20: Diagrama ubicaciones Embalpack S.L.

En este caso la ubicación debe ser introducida por el operario en el momento que se realiza la operación de carga (en la carretilla) o descarga (en otra posición, máquina o en una zona de entrada-salida). Esta posición se enviaría directamente en el mensaje y se guardaría en la base de datos.

Con este modelo, se ahorra en espacio de la base de datos la tabla que transforma las señales de wifi en el lugar real, además de la recepción de la multitud de señales y su cálculo en el servidor.

5.2.9 Destino automático y control de la ocupación.

Cuando un producto salga de una zona de entrada/para ser depositado en una zona de almacenaje o cuando salga de ser procesado de una máquina se puede llevar de manera automática a una zona propuesta por el sistema.

Para ello se va a llevar el control de los espacios para saber si están ocupados. Con una tabla adicional se marcará mediante un campo "Ocupado" si esa localización contiene un producto. Para hacer uso de este destino automático, el terminal de los operarios deberá lanzar una consulta para emplazar el nuevo producto y recibirá la primera localización que no está ocupada.

Como la tabla ordena las ubicaciones mediante su nombre, siempre se recibirán de manera alfabética la primera que no esté ocupada. Esto no implica que siempre vaya a haber una zona más ocupada del almacén (la zona AXX) ya que en caso de que ésta se encuentre completa, se buscará otra localización.



Para poder hacer uso de esta utilidad, al desplazar un producto, antes de grabar la nueva localización habrá que consultar la última y liberarla en caso de que sea una zona de almacenaje.

```
SELECT FIRST LOCATION FROM dbo.Ocupacion WHERE OCCUPIED=0
```

Cuando se mueva un material, en el momento de la descarga de este en otra zona del almacén se producirán las siguientes órdenes en la base de datos. Donde primero se libera en la tabla de ocupación el último registro que se tenía. Después se registra el nuevo lugar, y posteriormente se marca como ocupado.

A continuación, se muestra un ejemplo del desarrollo de la transacción, en el cual se marca como libre la última localización ocupada por el producto 444777888 que ahora se reubica en la A08. Posteriormente se registra la zona A08 como ocupada.

```
INSERT INTO dbo.OCUPACION (OCCUPIED) VALUES ('0') WHERE
LOCATION=(Última localización en la tabla que se tenga para ese ID)

//Marcamos la última localización del producto como libre

INSERT INTO dbo.LOCALIZACIONES (ID,LOCATION,TIMESTAMP,WORKER)
VALUES ('444777888','A08','2023-08-05 12:00:00','WORKER1')

//Insertamos la nueva localización

INSERT INTO dbo.OCUPACION (OCCUPIED) VALUES ('1') WHERE
LOCATION='A08'

//Marcamos nueva localización como ocupada

COMMIT

//Se hace commit al final para asegurar que la transacción se realiza en bloque
```

6. Implementación prototipo y pruebas

Para finalizar este trabajo se va a implementar un pequeño desarrollo donde se va a comprobar la funcionalidad de registro de posiciones y el uso de la tabla ocupación. En este caso, sobre un mismo producto, se le van a aplicar 3 movimientos distintos pasando de la posición A1, a la A2, y de ésta a la A3.

Para que se pueda visualizar en el portal de gestión de RabbitMQ, los envíos se van a espaciar 10 segundos cada uno. Primero se pondrá en marcha el escuchador y posteriormente se lanzarán los registros uno a uno.

```
public class Principal {
    public static void main(String[] argv) throws Exception {
        Reciever1.Espera();
        String s = "123456789;A1;Worker1";
        emiter1.Enviamensaje(s);
        Thread.sleep(10000);
        String s2 = "123456789;A2;Worker1";
        emiter1.Enviamensaje(s2);
        Thread.sleep(10000);
        String s3 = "123456789;A3;Worker1";
        emiter1.Enviamensaje(s3);
    }
}
```

Ilustración 21: Secuencia ejecución registros

Cada emisor se conecta a la cola, en este caso a una cola de pruebas con los usuarios *guest* (único usuario dado de alta por defecto y con todos los permisos disponibles). Debemos tener los parámetros del bróker como IP y puertos para la conexión. Además, por precaución siempre se dará de alta la cola (en caso de que ya esté creada se omite esta orden por parte del bróker). Para enviar el mensaje se coge la cadena a enviar y se convierte en una cadena de bytes pasándole por parámetro la codificación que tiene esta cadena.

```
public static void EnviaMensaje (String msg) throws Exception {
    String USER = "guest";
    String PWD = "guest";
    String HOST = "localhost";
    String VHOST = "/";
    String QUEUE_NAME = "TEST";
    String message=msg;
    ConnectionFactory factory = new ConnectionFactory();
    factory.setHost(HOST);
    factory.setUsername(USER);
    factory.setPassword(PWD);
    factory.setVirtualHost(VHOST);
    try (Connection connection = factory.newConnection();
        Channel channel = connection.createChannel()) {
        channel.queueDeclare(QUEUE_NAME, false, false, false, null);
        channel.basicPublish("", QUEUE_NAME, null, message.getBytes(StandardCharsets.UTF_8));
    }
}
```

Ilustración 22: Emisor de información

Por parte del receptor tenemos una implementación similar a la que nos encontramos en el emisor. Primeramente, se conecta a la cola con los mismos parámetros; IP, puerto, usuario/contraseña y el nombre de la cola. El consumidor contiene una llamada de tipo *DeliverCallback* para que no se finalice la ejecución al leer el primer mensaje, sino que continúe recibiendo mensajes. Una vez capturado el mensaje éste se pasa como parámetro al método *Insert* de la clase *ToDB*.

```
public static void Espera() throws Exception {

    String USER = "guest";
    String PWD = "guest";
    String HOST = "localhost";
    String VHOST = "/";
    String QUEUE_NAME = "TEST";
    ConnectionFactory factory = new ConnectionFactory();
    factory.setHost(HOST);
    factory.setUsername(USER);
    factory.setPassword(PWD);
    factory.setVirtualHost(VHOST);
    Connection connection = factory.newConnection();
    Channel channel = connection.createChannel();

    DeliverCallback deliverCallback = (consumerTag, delivery) -> {
        String message = new String(delivery.getBody(), StandardCharsets.UTF_8);
        String[] s2 = message.split(";");
        ToDB.Insert(s2);
    };
    channel.basicConsume(QUEUE_NAME, true, deliverCallback, consumerTag -> { });
}
```

Ilustración 23: Receptor en servidor para procesado de mensajes

Por último, tenemos el método *Insert* que acabamos de citar. Este método se conecta a la base de datos con los datos del controlador y la dirección de ésta. Tras cargar el controlador se hace una primera consulta donde se rescata la última localización del producto que estamos tratando. En caso de que tenga una última posición (no sea su primera aparición en la tabla Localizaciones) se guarda esta última posición para liberarla en la siguiente consulta (este procedimiento se puede hacer en una sola consulta con una subconsulta que nos saque la última localización).

```
public static void Insert(String[] s2) {
    try
    {
        String myDriver = "org.mariadb.jdbc.Driver";
        String myUrl = "jdbc:mariadb://localhost/test";
        String[] reg=s2;
        Class.forName(myDriver);
        Connection conn = DriverManager.getConnection(myUrl, "root", "");
        String ultimalocalizacion = "";
        //BUSCAMOS ULTIMA LOCALIZACIÓN
        String query2 = "SELECT `location`, `timestamp` FROM `sga_table` WHERE `ID`='"+reg[0]+" ORDER BY `timestamp` DESC; ";
        PreparedStatement preparedStmt2 = conn.prepareStatement(query2);
        ResultSet rs = preparedStmt2.executeQuery();
        String result=rs.getString(1);
        ultimalocalizacion=result;

        //LIBERAMOS LA ÚLTIMA LOCALIZACIÓN SI LA HAY
        if (ultimalocalizacion!=""){
            String query4 = "UPDATE `ocupacion` SET `Ocupado`='0' WHERE `Localizacion`='"+ultimalocalizacion+"'";
            PreparedStatement preparedStmt4 = conn.prepareStatement(query4);
            preparedStmt4.execute();
        }
    }
}
```

Ilustración 24: Búsqueda último registro si lo hay y liberación de esa localización

Se procede a grabar el nuevo registro de la nueva ubicación con la marca temporal calculada en la base de datos. Por último, se ocupa la nueva ubicación que nos llega como parámetro en el método y cerramos la conexión con la base de datos.

```
//GRABAMOS NUEVO REGISTRO
String query = "INSERT INTO `sga_table`(`ID`, `location`, `worker`) VALUES ('"+reg[0]+"','"+reg[1]+"','"+reg[2]+"')";
PreparedStatement preparedStmt = conn.prepareStatement(query);
preparedStmt.execute();
//OCUPAMOS NUEVA LOCALIZACIÓN
String query3 = "UPDATE `ocupacion` SET `Ocupado`='1' WHERE `Localizacion`='"+reg[1]+'";
PreparedStatement preparedStmt3 = conn.prepareStatement(query3);
preparedStmt3.execute();

//CERRAMOS CONEXIÓN SIEMPRE
conn.close();
```

Ilustración 25: Grabación del nuevo registro y ocupación de la nueva ocupación

En el portal de gestión de RabbitMQ podemos observar que los mensajes llegan espaciados y son consumidos automáticamente por el receptor que hemos dado de alta. En caso de no dar de alta el receptor primero, los 3 mensajes se acumularían en la cola y el receptor los leería a la vez. Esto provoca que de manera visual no podamos comprobar los cambios de estado en la base de datos.

Sistema de soporte a la localización de recursos en entornos industriales evolucionando hacia a un mundo digital. Una solución aplicada



Ilustración 26: Grafica Llegada mensajes

Por último, vemos como en la base de datos han entrado los 3 registros que hemos lanzado al principio, siendo la secuencia de localizaciones por las que ha pasado el recurso A1, A2 y A3. Como se ha dado de alta en un inicio el receptor, las inserciones han sido espaciadas por el mismo motivo que hemos comentado anteriormente en la gráfica del portal de gestión.

<input type="checkbox"/>	Editar	Copiar	Borrar	123456789	2023-09-20 19:13:14	A1	Worker1
<input type="checkbox"/>	Editar	Copiar	Borrar	123456789	2023-09-20 19:13:24	A2	Worker1
<input type="checkbox"/>	Editar	Copiar	Borrar	123456789	2023-09-20 19:13:34	A3	Worker1

Ilustración 27: Resultado inserciones

Adicionalmente, vemos que la última posición ocupada es la A3 y las otras 2 han quedado libres.

Localizacion	Ocupado
Borrar A1	0
Borrar A2	0
Borrar A3	1

Ilustración 28: Resultado tabla ocupación

Vemos que tal y como hemos planteado la solución, podemos llevar a cabo un registro básico de las localizaciones de los productos que se trabajan en el almacén y que tanto el registro histórico de localizaciones como el registro de ocupaciones funcionan correctamente.

7. Conclusiones

En el desarrollo de este trabajo hemos podido comprobar que aportar una solución de soporte a la localización de los recursos en un entorno industrial es posible sin realizar grandes inversiones tanto económicas como materiales. Las grandes empresas de software fabricantes de ERP's, llevan años con módulos que cubren estas necesidades para una amplia mayoría de empresas, la prueba la podemos encontrar en que las mayores corporaciones industriales tienen estos sistemas implementados. En cambio, en un ambiente más pequeño como pueda ser una PYME, con menor poder de capital y con unos recursos más limitados, se puede implementar un sistema que cumpla con unas necesidades básicas de localización instantánea, registro histórico y consulta de información.

La solución aportada es fácilmente adaptable por el personal de IT (que hoy en día encontramos en cualquier empresa), ya que no son necesarios amplios conocimientos de software específicos para llevar a cabo la solución. Además, permite el aprovechamiento de recursos ya existentes (como instalaciones de red o servidores), ya que la carga y el uso de estos sistemas es mínimamente invasivo y totalmente compatible con el resto de los servicios.

Alternativamente, y como se ha presentado al principio de este trabajo, la gran variedad de software para cada uno de los componentes de este sistema (desde la comunicación, la localización o el almacenamiento de la información registrada), permite una adaptación total a conocimientos ya existentes o preferencias/limitaciones de cualquier tipo.

Con relación a los estudios cursados, en este proyecto se han trabajado conocimientos vistos en la mayoría de las áreas que se imparten en la carrera. Se ha desarrollado una solución que implementa almacenamiento de información en una BBDD, así como la comunicación de diferentes elementos mediante 2 protocolos distintos y el desarrollo de software específico para la lógica interna. Además, también se han comentado aspectos de sistemas como la interconexión o la persistencia de los datos o la actuación ante desastres.

En conclusión, para la realización de este trabajo se han aplicado conocimientos adquiridos durante en transcurso de la carrera, e incluso en alguno de ellos, se han ampliado con el fin de buscar nuevas funcionalidades o características específicas que se buscaban para la solución planteada.



8. Referencias

- [1] B. Eastwood. [En línea]. Available: <https://www.northeastern.edu/graduate/blog/most-popular-programming-languages/>. [Último acceso: 28 07 2022].
- [2] Thales Group, «Thales,» [En línea]. Available: <https://www.thalesgroup.com/en/worldwide-digital-identity-and-security/mobile/magazine/5g-vs-4g-whats-difference>. [Último acceso: 28 07 2022].
- [3] Cisco, «What is Wifi,» [En línea]. Available: <https://www.cisco.com/c/en/us/products/wireless/what-is-wifi.html>. [Último acceso: 28 07 2022].
- [4] Eurostat, «Digital economy and society statistics - households and individuals,» [En línea]. Available: https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Digital_economy_and_society_statistics_-_households_and_individuals#Internet_access. [Último acceso: 28 07 2022].
- [5] CSA, «Zigbee | Complete IoT Soltions,» [En línea]. Available: <https://csa-iot.org/all-solutions/zigbee/>. [Último acceso: 28 07 2022].
- [6] Sony, « ¿Qué es la tecnología inalámbrica BLUETOOTH?,» [En línea]. Available: <http://download.sony-europe.com/pub/manuals/html/Z002678111/ES/contents/02/01/01/01.html>. [Último acceso: 28 07 2022].
- [7] VIU, «¿Qué es GSM y cómo funciona?,» [En línea]. Available: <https://www.universidadviu.com/es/actualidad/nuestros-expertos/que-es-gsm-y-como-funciona>. [Último acceso: 28 07 2022].
- [8] IBM, «Understanding enterprise messaging APIs and protocols,» [En línea]. Available: <https://developer.ibm.com/articles/messaging-protocols/>. [Último acceso: 28 07 2022].
- [9] S. R. Department, « Most popular database management systems worldwide 2022,» [En línea]. Available: <https://www.statista.com/statistics/809750/worldwide-popularity-ranking-database-management-systems/>. [Último acceso: 29 09 2022].
- [10] D. S. Block, « Why Amazon, Google, Netflix and Facebook Switched to NoSQL?,» 25 Julio 2018. [En línea]. Available: <https://www.linkedin.com/pulse/why-amazon-google-netflix-facebook-switched-nosql-shannon-block-cfe/>. [Último acceso: 29 09 2022].

- [11] C. Battista, A. Fumi, F. Giordano y M. Schiraldi, «Storage Location Assignment Problem,» 2011. [En línea]. Available: https://www.researchgate.net/publication/251734544_Storage_Location_Assignment_Problem_implementation_in_a_warehouse_design_optimization_tool.
- [12] W. Knight, «Wired,» [En línea]. Available: <https://www.wired.com/story/fords-smarter-robots-speeding-assembly-line/>. [Último acceso: 11 5 2023].
- [13] Interempresas, «Premio Icil para el almacén de Fedefarma, automatizado por SSI Schaefer,» 21 6 2012. [En línea]. Available: <https://www.interempresas.net/Logistica/Articulos/81409-Premio-Icil-para-el-almacen-de-Fedefarma-automatizado-por-SSI-Schaefer.html>. [Último acceso: 10 06 2023].
- [14] T. -. E. Team, «The manual sort order fulfilment,» TUP, 09 07 2013. [En línea]. Available: <https://logistikknowhow.com/en/picking/the-manual-sort-order-fulfilment/>.
- [15] R. A. a. Z. D. Song Chai, «An Indoor Positioning Algorithm Using Bluetooth,» 2016. [En línea]. Available: <https://www.atlantis-press.com/article/25858154.pdf>. [Último acceso: 23 08 2023].



ANEXO

OBJETIVOS DE DESARROLLO SOSTENIBLE

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

Objetivos de Desarrollo Sostenibles	Alto	Medio	Bajo	No Procede
ODS 1. Fin de la pobreza.				X
ODS 2. Hambre cero.				X
ODS 3. Salud y bienestar.		X		
ODS 4. Educación de calidad.				X
ODS 5. Igualdad de género.				X
ODS 6. Agua limpia y saneamiento.				X
ODS 7. Energía asequible y no contaminante.		X		
ODS 8. Trabajo decente y crecimiento económico.		X		
ODS 9. Industria, innovación e infraestructuras.	X			
ODS 10. Reducción de las desigualdades.		X		
ODS 11. Ciudades y comunidades sostenibles.		X		
ODS 12. Producción y consumo responsables.		X		
ODS 13. Acción por el clima.				X
ODS 14. Vida submarina.				X
ODS 15. Vida de ecosistemas terrestres.				X
ODS 16. Paz, justicia e instituciones sólidas.				X
ODS 17. Alianzas para lograr objetivos.	X			



Reflexión sobre la relación del TFG/TFM con los ODS y con el/los ODS más relacionados.

Con este trabajo centrado en la industria 4.0 y su alta evolución que estamos viviendo en los últimos años, se presenta una solución aplicada a una PYME local que requiere de un sistema de apoyo a la localización de sus productos.

Es importante plantear una solución que potencie o que apoye los ODS si queremos que ésta conviva con la sociedad que lucha por defender unos objetivos buscados por el bien de todos.

En relación con este trabajo, se aplican de manera muy notoria todos los objetivos centrados en la sostenibilidad del trabajo. Por un lado, la solución que se plantea está diseñada para que genere un mínimo impacto en la infraestructura empresarial ya existente, como por ejemplo, redes locales ya existentes, o servidores que ya existan. Esto se consigue utilizando aplicaciones ligeras que no supongan montar infraestructura específica para ellas con la nueva implantación.

Por otro lado, también se persigue una eficiencia y decencia en el trabajo, así como promover la salud y el bienestar de los trabajadores. El sistema simplifica y ayuda a que los trabajadores no tengan que ir recordando las ubicaciones de los productos, además, en caso de una nueva incorporación, ésta podría trabajar con la misma eficiencia que un operario con más experiencia.

En términos de sostenibilidad y consumo responsable, se conseguirá reducir el uso de papel y otros soportes desechables a la hora de hacer un inventariado de stock o seguimiento de las mercancías dentro de las instalaciones.

La solución se ha diseñado para que pueda ser interconectable. Esto aporta flexibilidad a la hora de su utilización, tanto por la propia empresa, como por otros servicios externos que se quieran vincular y, por tanto, aprovechen de esta solución para nutrirse de información que permita a la empresa evolucionar de una manera más rápida.

En resumen, guiar el trabajo con el objetivo de potenciar o perseguir estos Objetivos de Desarrollo Sostenible, aporta un extra a la hora de diseñar soluciones ya que además de que éstas, cumplan las premisas con las que han sido diseñadas, aprovechan para que, en un mundo globalizado, se potencien unos compromisos con el fin de mejorar e igualar las oportunidades.