



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Implantación de un sistema de etiquetas electrónicas en una empresa de venta on- line

Trabajo Fin de Grado

Grado en Ingeniería Informática

Autor: Vicent Llopis Castell

Tutor: Sergio Sáez Barona

Curso 2020-2021

Resumen

En la empresa donde he desarrollado el proyecto los pedidos se sacan en colecciones, es decir, siempre se sacan conjuntos de pedidos al mismo tiempo. Una colección de pedidos se saca utilizando un carro con tres baldas, de forma que en cada balda se añaden los productos de un pedido distinto. Teniendo esto en cuenta, cada vez que se prepara una colección de pedidos se imprime una etiqueta de papel por cada pedido, con datos del propio pedido, que se pega en la balda del carro donde se van a añadir los productos del pedido correspondiente.

El objetivo de este proyecto es sustituir estas etiquetas impresas en papel por etiquetas electrónicas. La puesta en marcha de las etiquetas electrónicas constará principalmente de dos pasos. Primeramente, habrá que vincular las etiqueta electrónica a las baldas de los carros de forma permanente. El siguiente paso ya será mostrar en la pantalla de las etiquetas electrónicas la información del pedido que se está preparando.

Para cumplir este objetivo se ha desarrollado una API REST que realizará la función de cliente/servidor. Esta API se conecta a una interfaz gráfica, para obtener datos proporcionados por el cliente y almacenarlos en la base de datos, y viceversa. La API desarrollada también realiza peticiones a otra API de una empresa subcontratada de etiquetas electrónicas. Las herramientas utilizadas han sido NodeJs para la capa del servidor, HTML y JavaScript para la parte del cliente y DBeaver para la administración de la base de datos.

Palabras clave: etiquetas electrónicas, base de datos, NodeJS, empresa

Abstract

In the company where I have developed the project, orders are taken out in collections, in other words, sets of orders are always taken out at the same time. A collection of orders is taken out using a trolley with three shelves, so that products from a different order are added to each shelf. Taking this into account, each time a collection of orders is prepared, a paper label is printed for each order, with data from the order itself, which is pasted on the shelf of the cart where the products of the corresponding order will be added.

The objective of this project is to replace these labels printed on paper with electronic labels. The implementation of electronic labels will consist mainly of two steps. First, the electronic labels must be permanently attached to the shelves of the trolleys. The next step will be to show the information of the order that is being prepared on the screen of the electronic labels.

In order to fulfil this objective, a REST API has been developed that will perform the client / server function. This API connects to a graphical interface, to obtain data provided by the client and store it in the database, and vice versa. The developed API also makes requests to another API from an electronic tag outsourcing company. The tools used have been NodeJs for the server layer, HTML and JavaScript for the client side and DBeaver for the database administration.

Keywords: electronic tag, database, NodeJS, company.

RESUM

En l'empresa on he desenvolupat el projecte les comandes es trauen en col·leccions, és a dir, sempre es trauen conjunts de comandes al mateix temps. Una col·lecció de comandes es trau utilitzant un carro amb tres baldes, de manera que en cada balda s'afigen els productes d'una comanda diferent. Tenint això en compte, cada vegada que es prepara una col·lecció de comandes s'imprimeix una etiqueta de paper per cada comanda, amb dades de la pròpia comanda, que es pega en la balda del carro on s'afegiran els productes de la comanda corresponent.

L'objectiu d'aquest projecte és substituir aquestes etiquetes impreses en paper per etiquetes electròniques. La posada en marxa de les etiquetes electròniques constarà principalment de dos passos. Primerament, caldrà vincular les etiqueta electrònica a les baldes dels carros de manera permanent. El següent pas ja serà mostrar en la pantalla de les etiquetes electròniques la informació de la comanda que s'està preparant.

Per a complir aquest objectiu s'ha desenvolupat una API REST que realitzarà la funció de client/servidor. Aquesta API es connecta a una interfície gràfica, per a obtenir dades proporcionades pel client i emmagatzemar-los en la base de dades, i viceversa. La API desenvolupada també realitza peticions a una altra API d'una empresa subcontractada d'etiquetes electròniques. Les eines utilitzades han sigut NodeJs per a la capa del servidor, HTML i Javascript per a la part del client i DBeaver per a l'administració de la base de dades.

Paraules clau: etiquetes electròniques, base de dades, NodeJS, empresa.

Tabla de contenidos

Contenido

1.	Introducción.....	10
1.1	Motivación	11
1.2	Objetivos.....	12
1.3	Impacto Esperado	12
1.4	Metodología.....	13
1.5	Estructura.....	14
2.	Estado del arte.....	15
3.	Análisis del problema.....	19
3.1	Requisitos funcionales	20
3.2	Requisitos no funcionales	20
3.3	Casos de Uso	21
3.4	Análisis de riesgos.....	23
3.5	Solución propuesta	24
3.6	Plan de trabajo	25
3.7	Presupuesto.....	26
4.	Diseño de la solución	27
4.1	Arquitectura del sistema	27
4.2	Diseño detallado	28
4.2.1	Capa de persistencia	28
4.2.2	Capa de presentación	31
4.3	Tecnología utilizada	34
5.	Desarrollo de la solución propuesta.....	38
5.1	Preparación del entorno.....	38
5.2	Archivo de configuración	39
5.3	API REST	40
5.4	Flujo de control.....	40
5.5	Authorization	48
5.6	Diseño pantalla etiqueta electrónica	49
6.	Implantación	50
7.	Pruebas.....	51
7.1	Inicio.....	51
7.2	Caso de Uso 1: Vincular etiqueta con carro	52

7.3	Caso de Uso 2: Insertar tique en etiqueta.....	53
7.4	Caso de Uso 3: Ver etiquetas vinculadas	54
7.5	Caso de Uso 4: Ver registro de tiques asignados	55
7.6	Caso de Uso 5: Encender LED	56
8.	Conclusiones	57
8.1	Relación del trabajo desarrollado con los estudios cursado.....	57
8.2	Trabajos a futuro	57
9.	Referencias.....	58
10.	Términos.....	59

Índice de figuras

Ilustración 1. Carro con etiquetas electrónicas	11
Ilustración 2. Etiqueta electrónica	13
Ilustración 3. Metodología del proyecto.....	14
Ilustración 4. Evolución ventas VerdNatura	15
Ilustración 5. Ranking de VerdNatura	15
Ilustración 6. Presupuesto SES imagoTag	16
Ilustración 7. Mantenimiento Telefónica	17
Ilustración 8. Presupuesto Telefónica	17
Ilustración 9. Presupuesto Aeme group	18
Ilustración 10. Casos de uso	21
Ilustración 11. Diagrama de Gantt.....	26
Ilustración 12. Presupuesto total etiquetas electrónicas	26
Ilustración 13. Componentes arquitectura sistema.....	28
Ilustración 14. Vista SQL	29
Ilustración 15. Modelo de la base de datos	30
Ilustración 16. Prototipo “Inicio”	31
Ilustración 17. Prototipo “Seleccionar carros”.....	32
Ilustración 18. Prototipo “Vincular etiqueta con carro”	32
Ilustración 19. Prototipo “Ver etiquetas vinculadas”	33
Ilustración 20. Prototipo “Registro tiques asignados”	33
Ilustración 21. Prototipo “Encender LED”	34
Ilustración 22. Logo JavaScript.....	34
Ilustración 23. Logo NodeJS	35
Ilustración 24. Logo Express	35
Ilustración 25. Logo DBeaver	36
Ilustración 26. Logo MariaDB.....	36
Ilustración 27. Logo Visual Studio Code	37
Ilustración 28. Logo Git.....	37
Ilustración 29. Logo HTML.....	37
Ilustración 30. Logo Postman	38
Ilustración 31. Archivo información proyecto.....	39
Ilustración 32. Archivo configuración	40
Ilustración 33. Rutas API	40
Ilustración 34. Pantalla “Vincular etiqueta con carro”	41
Ilustración 35. Petición HTTP para insertar artículo.....	42
Ilustración 36. Petición HTTP para vincular una etiqueta con un artículo	42
Ilustración 37. Pantalla “Mostrar información en una etiqueta”	43
Ilustración 38. Vista SQL “collection_newSmartTag”	44
Ilustración 39. Petición actualizar artículo	44
Ilustración 40. Pantalla “Encender LED”	45
Ilustración 41. Petición HTTP para encender LED	45
Ilustración 42. Pantalla “Ver etiquetas vinculadas”	46
Ilustración 43. Pantalla “Registro tiques asignados”	47
Ilustración 44. Petición HTTP con “Authorization”	48

Ilustración 45. Obtención de la “Authorization”	48
Ilustración 46. Diseño pantalla etiqueta electrónica.....	49
Ilustración 47. Archivo para crear servidor.....	50
Ilustración 48. Servidor corriendo	51
Ilustración 49. Prueba pantalla “Inicio”	51
Ilustración 50. Prueba pantalla “Vincular etiqueta con carro”	52
Ilustración 51. Etiquetas mostrando el carro y nivel.....	52
Ilustración 52. Prueba pantalla “Mostrar información en una etiqueta”.....	53
Ilustración 53. Etiquetas mostrando tiques	53
Ilustración 54. Prueba pantalla “Ver etiquetas vinculadas”	54
Ilustración 55. Prueba pantalla “Registro tiques asignados”	55
Ilustración 56. Prueba pantalla “Encender LED”	56
Ilustración 57. Etiqueta con el LED encendido.....	56

1. Introducción

La tecnología como se viene viendo durante los últimos años no ha parado de evolucionar y ampliarse a prácticamente cualquier sector y dispositivo. En este proyecto nos vamos a enfocar en un dispositivo que no lleva mucho tiempo en el mercado, las etiquetas electrónicas.

Una etiqueta electrónica es un sistema usado por los vendedores al por menor para mostrar el precio de los productos en sus comercios. Sin embargo, para nuestro proyecto su objetivo es ligeramente distinto, el objetivo es mostrar la información útil del pedido que se está preparando para ser enviado a un cliente. Las etiquetas electrónicas no son sólo una herramienta de gestión de información, sino también de eficiencia y productividad.

El proyecto lo voy a realizar en colaboración con la empresa en la que estoy realizando las prácticas denominada VerdNatura S.L. El proyecto dentro de la empresa se dividió en dos partes:

1. Desarrollo de una API REST. Esta es la parte del proyecto que se me ha encargado.
2. Desarrollo del frontend correspondiente en la aplicación Android actual de la empresa. De esta parte se encarga otro empleado de la empresa. Esta parte no se tratará en mi proyecto TFG.

La distribución del trabajo ha sido decisión de la empresa. Con el fin de seguir aprendiendo y mejorando mis conocimientos, decidí realizar por mi parte una interfaz gráfica con la interactuará la API REST desarrollada.

Las etiquetas se conectan a través de bluetooth 5.0 a unos Puntos de Acceso previamente instalados en la nave, a través de los cuales se transmite la información.



Ilustración 1. Carro con etiquetas electrónicas

1.1 Motivación

En la empresa donde estoy realizando las prácticas tenían la necesidad de implantar etiquetas electrónicas en su proceso de producción. Era un proyecto que tenían ya hace tiempo en mente.

Cuando me incorporé a la empresa me ofrecieron este proyecto e inmediatamente acepté por la motivación de tener mi primer proyecto laboral, además de poder ayudar a la empresa a lograr su objetivo.

Las etiquetas electrónicas las queremos incorporar en los carros de preparación de pedidos (picking), sustituyendo a unas etiquetas impresas en papel. La fase de la preparación de pedidos que nos interesa conocer es la siguiente, un trabajador coge tantos carros como desee, a continuación, entra en el almacén donde están todos los productos, cada trabajador tiene una PDA que le indica por pantalla que productos debe coger y colocar en los carros que acaba de coger.

Cada carro tiene tres baldas con lo que cada balda tendrá incorporada una etiqueta electrónica. Es decir, un total de tres etiquetas electrónicas por carro. Cada etiqueta electrónica muestra la información de un pedido diferente, o lo que es lo mismo, cada balda del carro se corresponde con un pedido diferente. De este modo se soluciona el problema de tener que estar pegando y despegando constantemente las etiquetas en los carros, esto implica un aumento de productividad ya que el empleado no pierde tiempo.

Como ya hemos mencionado los empleados que sacan los pedidos llevan siempre en mano una PDA. De forma que cuando tienen que sacar un producto antes lo escanean con la PDA. Esta información es importante porque todas las etiquetas tienen incorporadas una luz LED. De modo que cuando un trabajador escanee un producto con la PDA, la etiqueta electrónica que contenga la información del pedido al que corresponde ese producto, se iluminará.

1.2 Objetivos

El objetivo del proyecto es lograr la puesta en marcha de las etiquetas electrónicas. De forma que muestren la información deseada de cada pedido, y se encienda el LED cuando corresponda.

Con la interfaz gráfica, podremos transmitir la información a la API, así como visualizar distintos datos de interés.

Para ello se ha decidido crear una aplicación web que necesitará alcanzar los siguientes objetivos:

- Implementar una API capaz recibir peticiones de un cliente, conectada con la API de las etiquetas electrónicas y con la BBDD.
- Implementar una interfaz gráfica, que actúe como cliente para poder realizar peticiones a la API.

1.3 Impacto Esperado

Este proyecto supone diferentes ventajas para la empresa VerdNatura, las ventajas son:

- Con el LED de las etiquetas electrónicas, se reducirán notablemente los errores de los trabajadores a la hora de colocar los productos en los carros.

- Los empleados ya no perderán tiempo pegando y despegando las etiquetas en papel, ya que una vez instalas las etiquetas electrónicas será automático.
- Ahorro económico en impresoras y papel.



Ilustración 2. Etiqueta electrónica

1.4 Metodología

Una vez se ha establecido cual es el problema por resolver, y analizado cual es la solución que se intentará dar al problema, la metodología que se ha seguido para realizar el proyecto es un desarrollo en secuencia, concretamente el modelo en cascada:

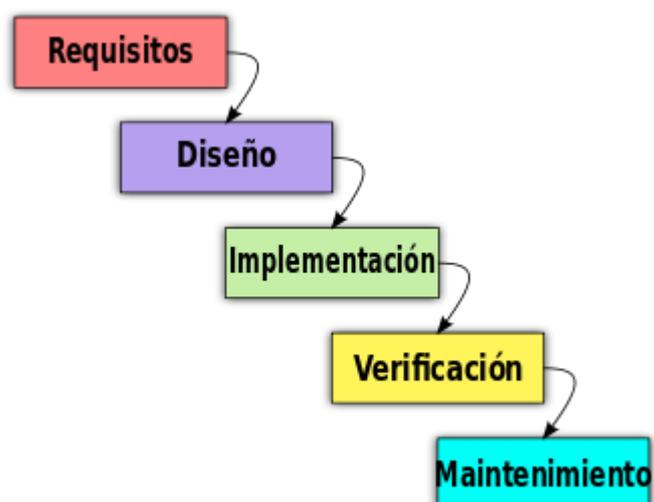


Ilustración 3. Metodología del proyecto

1. **Análisis de requisitos del software:** en esta fase se analizan las necesidades que se quieren suplir, las necesidades determinarán los objetivos que se deben cumplir.
2. **Diseño:** esta fase sirve para formular una solución específica en base al análisis de requisitos de la fase anterior. En esta fase, se ha desarrollado la arquitectura del software y un plan de diseño detallado del mismo.
3. **Implementación:** esta fase incluye la programación del software, la búsqueda de errores y las pruebas.
4. **Verificación:** como su propio nombre indica, una vez se termina la fase de implementación se verifica que todos los componentes del sistema funcionen correctamente y cumplen con los requisitos.
5. **Mantenimiento:** una vez se han desarrollado todas las funcionalidades del software y se ha comprobado que funcionan correctamente, se inicia la fase de instalación y mantenimiento. Se instala la aplicación en el sistema y se comprueba que funcione correctamente en el entorno en que se va a utilizar.

1.5 Estructura

El actual documento se ha organizado en las siguientes partes:

1. **Introducción:** el primer capítulo corresponde a este mismo apartado, donde se introduce el trabajo a realizar, la motivación por la que se ha decidido realizar el TFG, los objetivos que se pretenden conseguir, las ventajas que supondrá el trabajo realizado, y la metodología que se ha utilizado para llevar a cabo el proyecto.
2. **Estado del arte:** en este punto se comenta brevemente la historia y evolución de la empresa donde he realizado en proyecto.
3. **Análisis del problema:** se describen los problemas actuales, y que pretende resolver mi proyecto.
4. **Diseño de la solución:** en este capítulo se explica la arquitectura del sistema, capa de persistencia, capa de presentación y las tecnologías utilizadas para desarrollar el proyecto.
5. **Desarrollo de la solución propuesta:** se describe detalladamente el desarrollo de la solución planteada.
6. **Implantación:** en el sexto apartado, se detalla cómo se ha desplegado el proyecto.
7. **Pruebas:** se describen las pruebas realizadas para comprobar que las funcionalidades funcionan correctamente, y se muestran los resultados de estas pruebas.
8. **Conclusiones:** se analiza si han alcanzado los objetivos, junto con una valoración del proyecto en su conjunto.

2. Estado del arte

Como ya he mencionado anteriormente, he realizado el TFG en conjunto a una empresa donde estaba realizando las prácticas. La empresa denominada, VerdNatura Levante SL, se dedica al comercio al por mayor de flores y plantas.

La historia de VerdNatura es la historia de un agricultor que no podía llegar a finde mes por la constante caída de precios de frutas y hortalizas. Este problema le llevo a buscar alternativas, y encontró una solución en el cultivo de plantas ornamentales que comenzó a vender a floristerías junto con un socio. Pronto comenzó a captar clientes a lo largo de España, y desde ese momento la empresa no ha parado de crecer. Hoy en día, VerdNatura dispone de dos sedes, una en Madrid y la otra en Holanda, y tiene clientes por todo el mundo. A continuación, podemos observar la evolución de ventas de los últimos años, y su posición en el ranking de empresas de toda España:



Ilustración 4. Evolución ventas VerdNatura



Ilustración 5. Ranking de VerdNatura

Implantación de un sistema de etiquetas electrónicas en una empresa de venta on-line

Seguidamente, vamos a analizar las diferentes compañías de etiquetas electrónicas que se han estudiado para la adquisición de las etiquetas electrónicas. A continuación, podemos ver en las imágenes los productos y servicios que ofrecía cada compañía junto con su presupuesto:

- **SES imagotag:** Esta empresa ofrecía 40 etiquetas electrónicas a 65,89€ la unidad, un Punto de Acceso, y todo lo relacionado con la instalación y formación sobre las etiquetas electrónicas. Todo esto por un total de 8.381,80€.

SISTEMA SES ETIQUETAS ELECTRONICAS PARA SUPERFICIES HASTA 6000 M ²				
Referencia	Descripción	Cantidad	Precio Unitario	Total
ETIQUETA ELECTRONICA DE GONDOLA	Etiqueta G1 Retail 7.4 NB NFC (Paquete de 40 etiquetas)	40	65,89 €	2 635,60 €
	Etiqueta G1 Retail 4.2 NB NFC (Paquete de 60 etiquetas)	0	41,68 €	0,00 €
SISTEMA DE TRANSMISION RADIO	AP 2010 Bundle	1	478,70 €	478,70 €
SISTEMA INFORMATICO	Servidor SES + Base de datos + Licencias	1	890,00 €	890,00 €
	Software Jeegy G1 Pricing&Analytics Mini (-500 ESLs)	1	948,50 €	948,50 €
TOTAL SISTEMA SES				4 952,80 €
FIJACIONES Y ACCESORIOS ----- A DEFINIR				
Referencia	Descripción	Cantidad	Precio Unitario	Total
TOTAL FIJACIONES				0,00 €
INSTALACION				
Referencia	Descripción	Cantidad	Precio Unitario	Total
INSTALACION Y FORMACION	Sistema de transmision radio Alta frecuencia (Configuracion & instalacion de los Aps, configuracion del sistema informatico)	1	875,00 €	875,00 €
	Enlace Informatica (desarollo, test y preparacion servidor en fabrica) + Gestion de proyecto (Solo una vez)	1	1 524,00 €	1 524,00 €
	Formacion	1	280,00 €	280,00 €
	Colocacion de las 100 primeras Etiquetas con sus soportes	1	GRATIS	0,00 €
	Contrato tele-asistencia anual	1	300,00 €	300,00 €
TOTAL INSTALACION				2 979,00 €
TRANSPORTE				450,00 €
TOTAL SISTEMA SES CON INSTALACION				8 381,80 €

Ilustración 6. Presupuesto SES imagotag

- **Telefónica:** La conocida empresa multinacional española, ofrecía 480 etiquetas electrónicas por 8,30€ la unidad, un Punto de Acceso, instalación del servicio, junto con soportes y accesorios para colocar las etiquetas. También incluye un mantenimiento de 12 meses por 562,72€. La suma de todo lo mencionado asciende a 14.968,29 €.

HW para 1 localización		5.367,00 €	
Etiquetas para las tiendas			
Etiquetas			
480	VUSION 2.6 BWR GL440 F/anthracite B/anthracite - 3342	8,32 €	3.993,86 €
Etiquetas			
24	Stock de seguridad del 5% - VUSION 2.6 BWR GL440 F/anthracite B/anthracite - 3342	8,32 €	199,69 €
Elementos de comunicación			
Elementos de comunicación			
1	Retail IoT Connector Huawei 2.4 GHz ESL, CE - "Dongle"	382,74 €	382,74 €
Stock de seguridad Dongle + Core Appliance			
1	Retail IoT Connector Huawei 2.4 GHz ESL, CE - "Dongle"	382,74 €	382,74 €
Soportes y accesorios			
40	Rail R6 Standard L1320 mm [transparent] - 4447	4,61 €	184,33 €
10	VUSION Clip removal tool - Nylon - with battery opener - 4612	8,32 €	83,21 €
3	VUSION Easylock tool V1 - 4611	38,27 €	114,82 €
1	Tornillos 4.2 X 16mm para fijación Rail (caja de 500) - 4552	25,60 €	25,60 €
1 Servicios Oneshot		9.448,96 €	
Instalación			
Jefatura de proyecto			
1	Jefatura, coordinación y seguimiento del proyecto para una tienda	2.587,77 €	2.587,64 €
1	Gestión PRL (única plataforma)	881,12 €	881,12 €
instalación			
2	Plataformado AP + Dongle y entrega en cliente	512,03 €	512,03 €
1	Precio por metro de cableado. Se cotizará aparte los metros necesarios por tienda tras visitas replanteo.	3,80 €	3,80 €
480	Instalación Nocturna de soportes, railes y etiquetas; desinstalación de railes de cliente anteriores y asignación de Etiquetas Electrónicas a producto	2,56 €	1.228,88 €
1	Transporte etiquetas por palé Península y Baleares	268,82 €	268,82 €
Configuración y puesta en marcha del servicio (Pago único para todo el proyecto)			
1	Licencia Spotshelf Onpremise Mini On-Premise (500 - 3000 ESLs) + Core service licence 3 AP/500 ESL	1.768,82 €	1.768,82 €
1	Diseño de template hasta 3 tamaños de etiquetas, con 4 escenarios posibles según condiciones lógicas (oferta, provincia...)	1.973,83 €	1.973,83 €
1	Formación uso básico de plataforma Jeegy - 3 horas online	224,01 €	224,01 €

Ilustración 8. Presupuesto Telefónica

N.º Unidades	Meses	Descripción	Precio final unitario	Precio final total
12		Servicios Recurrentes	562,72 €	
Mantenimiento y soporte				
Mantenimiento				
1	2	Envío de HW - (Retail IoT Connector Huawei) sin visita de técnico en tienda, con stock centralizado incluye: plataformado, logística, logística inversa y gestión de garantía con fabricante y validación remota	2,99 €	143,37 €
1	1	Expert Support Plan - Nano (up to 1000 ESLs) 8x5	8,74 €	419,35 €

Ilustración 7. Mantenimiento Telefónica

Implantación de un sistema de etiquetas electrónicas en una empresa de venta on-line

- Aeme group:** Esta es la empresa que se decidió contratar finalmente para la implantación de las etiquetas electrónicas. Ofrecen 500 etiquetas electrónicas por 10,82€ la unidad, cinco Puntos de Acceso, instalación, y accesorios por un total de 6.810€. El factor económico fue clave para decidir que empresa contratar, a esto hay que sumar que las características que poseen las etiquetas de Aeme group, que se comentarán más adelante.

Código	Foto	Nombre	Descripción	Neto Factura (sin IVA)	Unidades	Total Neto Factura (sin IVA)
ESL - ETIQUETAS ELECTRONICAS INTELIGENTES						
		ESL 1,5" Tricolor	ESL 1,5" Tricolor Led Indicator Tres colores 1,5", tamaño etiqueta 44.5*35.9*13.7mm	8,97 €		0,00 €
		ESL 2,1" Tricolor	ESL 2,1" Tricolor Led Indicator Tres colores 2,1", tamaño etiqueta 72*36.4*14.3mm	6,00 €		0,00 €
		ESL 2,1" B/N	ESL 2,1" B/N FREEZER AREA. IP67. Led Indicator Dos colores 2,1", tamaño etiqueta 64,7mm X 36,5mm X 12mm. Batería no intercambiable. Duración 5 años	12,42 €		0,00 €
		ESL 2,7" Tricolor	ESL 2,7" Tricolor Led Indicator Tres colores 2,7", tamaño etiqueta 77.71*50*14.3mm	13,52 €		0,00 €
		ESL 2,9" Tricolor	ESL 2,9" Tricolor Led Indicator Tres colores 2,9", tamaño etiqueta 90*41*14.5mm	10,82 €	500	5.410,00 €
		ESL 4,2" Tricolor	ESL 4,2" Tricolor Led Indicator Tres colores 4,2", tamaño etiqueta 105.3*94.5*12.3mm	30,99 €		0,00 €
		ESL 5,8" Tricolor	ESL 5,8" Tricolor Led Indicator Tres colores 5,8", tamaño etiqueta 138.3*111.8*13.8mm	53,29 €		0,00 €
		ESL 7,5" Tricolor	ESL 7,4" Tricolor Led Indicator Tres colores 7,4", tamaño etiqueta 185*124.3*13mm	83,63 €		0,00 €
		ESL 11,6" Tricolor	ESL 11,6" Tricolor Led Indicator Tres colores 11,6", tamaño etiqueta 266,3*190*9mm	188,59 €		0,00 €
		ESL 7,7" B/N	ESL 7,7" B/N Dual Parte para insertar papel impreso B/N 7,7", tamaño etiqueta 215,7*206,8*13mm	100,03 €		0,00 €
TOTAL ETIQUETAS ELECTRÓNICAS					500	5.410,00 €
TELECOM & IT HARDWARE						
		Access Point (AP)	Acces Point BLE. DUAL BAND WI-FI Access Point Wi-Fi. Dual Band. 2.4 / 5 Ghz. 48V POE. Dimensiones 183*183*35mm	200,00 €	5	1.000,00 €
TOTAL HARDWARE						1.000,00 €
SERVICIO ESL CLOUD						
		Servicio Cloud 1000 etiquetas / ANUAL	Servicio CLOUD / Cuota anual. Servicio de atención telefonica de Incidencias: Lunes a Viernes de 9,00 a 14,00 horas. Servicio de sustitución de etiquetas averiadas en 24 - 48 horas	150,00 €	1	150,00 €
TOTAL LICENCIAS PAGO ANUAL						150,00 €
INTEGRACION SOFTWARE & ERP Y PERSONALIZACIÓN						
		Integración ERP	Integración Software ESL CLOUD con ERP del cliente. Opcional.	600,00 €	0	0,00 €
		Personalización etiquetas	Diseño personalizado de las plantillas de etiquetas. Opcional	170,00 €	0	0,00 €
TOTAL INTEGRACIÓN ESL CLOUD & ERP Y PERSONALIZACIÓN						0,00 €
INSTALACIÓN SOLUCIÓN ESL EN EL CLOUD						
INST1		Instalación solución CLOUD	Dirección del proyecto e instalación física de las etiquetas en punto de venta por parte de departamento técnico. Confirmación después de estudio previo por parte del departamento técnico.	1.000,00 €		0,00 €
ACCS1		Ralles y Accesorios	Suministro de ralles y accesorios.	0,50 €	1	250,00 €
TOTAL DIRECCIÓN DE PROYECTO E INSTALACIÓN						250,00 €
TOTAL SOLUCIÓN ESL CLOUD						6.810,00 €

* Los precios no incluyen transporte ni impuestos tales como aduanas e IVA.

Ilustración 9. Presupuesto Aeme group

3. Análisis del problema

En los siguientes puntos se va a explicar con detalle los problemas que presenta VerdNatura con las actuales etiquetas de papel:

- Un problema constante es el fallo humano, y es que como ya se ha comentado en la introducción, los trabajadores se equivocan cuando colocan los productos en un carro, ya que a menudo lo colocan en una balda que no corresponde. Este problema se solucionará con el LED de las etiquetas electrónicas, que básicamente se encenderá indicando la balda a la que corresponde el producto.
- El otro problema es la forma en que se trabaja actual, concretamente cada vez que se saca un pedido el empleado tiene que coger un carro, ir a la impresora, coger la pegatina que se ha impreso como identificador de ese pedido, y a continuación pegar en el carro la etiqueta. Todo esto proceso, desembocaba en las quejas de los empleados sobre todo a la hora de despegar la etiqueta y pegarla, en propias palabras de los trabajadores: “Esto es desesperante”.
- Las actuales impresoras por las que se imprimen las etiquetas mencionadas en el punto anterior son otro problema que hay que tener en cuenta, ya que fallan a menudo por diferentes motivos.

En conclusión, la incorporación de etiquetas electrónicas eliminará todos los problemas ya mencionados. Se reducirá considerablemente el error de colocar un producto en una balda incorrecta, el problema de pegar y despegar etiquetas desaparecerá, ya que todo se hará automáticamente. Tampoco habrá que depender de las impresoras, para seguir trabajando.

3.1 Requisitos funcionales

A continuación, nos encontramos con los requisitos funcionales del sistema:

- El LED de una etiqueta electrónica deberá iluminarse cuando la API reciba un tique, junto a la petición de encender el LED.
- Cada vez que se proceda a vincular una etiqueta con un carro, se realizarán dos llamadas a la API de las etiquetas electrónicas. Una llamada para insertar el identificador del carro y otra llamada para vincular este carro con la etiqueta introducida. Además, en la BBDD tendrá que almacenarse esta vinculación, es decir, cada carro que etiquetas tiene vinculadas.
- Cada vez que se quiera mostrar la información de un pedido en una etiqueta, se realizará una llamada a la API de las etiquetas electrónicas para actualizar los datos que muestra la etiqueta. También, se almacenará en la BBDD la relación entre la etiqueta y el tique que ha mostrado.
- Mediante la interfaz gráfica, se debe permitir consultar las etiquetas electrónicas vinculadas.
- Mediante la interfaz gráfica, se debe permitir consultar un registro de los tiques que se han mostrado en etiquetas electrónicas.
- Mediante la interfaz gráfica, se debe permitir la vinculación de etiquetas con carros.
- Mediante la interfaz gráfica, se debe permitir el envío de la información que se quiere mostrar a las etiquetas.

3.2 Requisitos no funcionales

Los requisitos no funcionales que se pretenden conseguir son:

- La interfaz debe ser intuitiva y fácil de utilizar.
- La lista de las etiquetas vinculadas no debe tardar más de 3 segundos en cargar.
- La lista del registro de tiques no debe tardar más de 3 segundos en cargar.
- Todas las comunicaciones con la API de las etiquetas electrónicas se realizarán haciendo uso de la autenticación basada en tokens.

3.3 Casos de Uso

A continuación, se muestra la imagen de los casos de uso que se desean implementar:

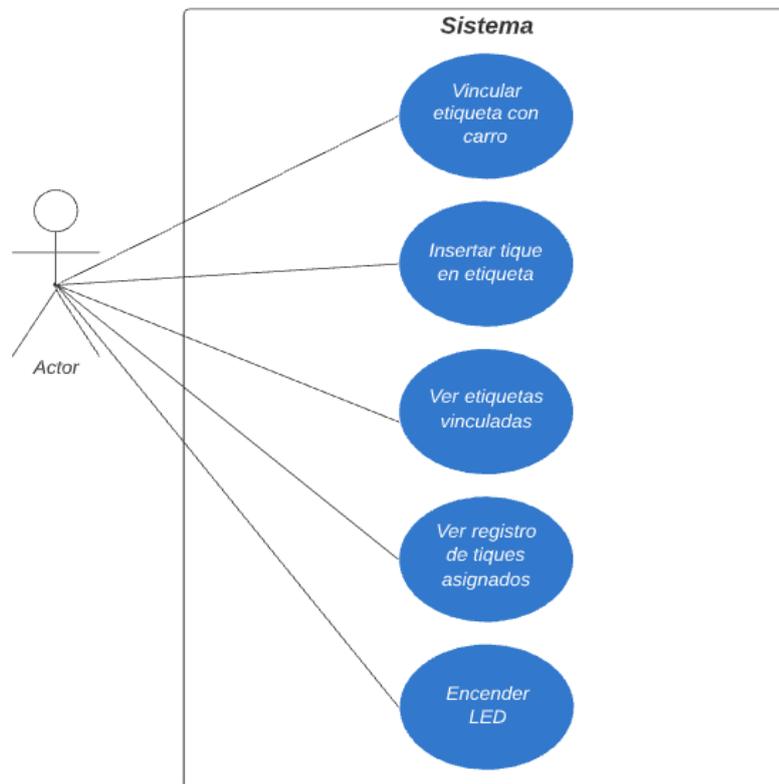


Ilustración 10. Casos de uso

Caso de Uso 1	Vincular etiqueta con carro
Objetivo	Vincular una etiqueta electrónica a un carro.
Descripción	Interfaz donde se deberán introducir tres datos, el identificador del carro, el nivel del carro (que es la balda del carro) y el identificador de la etiqueta, de esta forma la etiqueta introducida quedará vinculada a ese carro y nivel (balda).
Precondición	
Secuencia Normal	En la ventana principal: <ul style="list-style-type: none"> - Pulsar en el botón «Vincular etiqueta con carro». - Introducir datos solicitados.
Postcondición	

Caso de Uso 2	Insertar tique en etiqueta
Objetivo	Mostrar la información del tique en una etiqueta electrónica.
Descripción	Interfaz donde habrá que indicar cuantos carros vas a coger para sacar pedidos. Una vez introducidos los identificadores de los carros, las etiquetas electrónicas mostraran la información de los tiques correspondientes.
Precondición	
Secuencia Normal	En la ventana principal: <ul style="list-style-type: none"> - Pulsar en el botón «Seleccionar carros». - Introducir datos solicitados.
Postcondición	

Caso de Uso 3	Ver etiquetas vinculadas
Objetivo	Ver las etiquetas electrónicas que están vinculadas a un carro.
Descripción	Servirá para visualizar el identificador de las etiquetas junto al carro y nivel que están vinculadas.
Precondición	
Secuencia Normal	En la ventana principal: <ul style="list-style-type: none"> - Pulsar en el botón «Ver etiquetas vinculadas».
Postcondición	

Caso de Uso 4	Ver registro de tiques asignados
Objetivo	Ver todos los tiques que se han asignado a una etiqueta electrónica.
Descripción	Servirá para visualizar el identificador del tique, la etiqueta electrónica que ha mostrado dicho tique, el trabajador que ha sacado el pedido, y otros datos de interés.
Precondición	
Secuencia Normal	En la ventana principal: - Pulsar en el botón «Ver etiquetas vinculadas».
Postcondición	

Caso de Uso 5	Encender LED
Objetivo	Encender el LED de una etiqueta electrónica,
Descripción	Interfaz donde se deberá introducir un identificador de un tique, de manera que se encienda el LED de la etiqueta electrónica que este mostrando la información de dicho tique.
Precondición	
Secuencia Normal	En la ventana principal: - Pulsar en el botón «Encender LED». - Introducir datos solicitados.
Postcondición	

3.4 Análisis de riesgos

Como en todo proyecto de software aparece el riesgo de que no se acaben cumpliendo los requisitos funcionales. A fin de mitigar este efecto, nuestro proyecto se ha desarrollado para que en un futuro sea fácil de modificar y migrar.

Otro riesgo que hay que tener en cuenta es el tiempo que tardan las etiquetas electrónicas en mostrar la información. Desde que se envía la información hasta que se muestra en la etiqueta electrónica siempre pasan entre unos 5-15 segundos. Esta espera puede producir un efecto inverso al deseado ya que puede pausar demasiado tiempo el trabajo de los empleados. Este riesgo se ha valorado con el jefe del proyecto en su momento y se ha aceptado.

El riesgo económico que supone comprar todas las etiquetas electrónicas para poner en marcha su funcionamiento. Un pack de 100 cuesta alrededor de 1760€, y la empresa requiere mínimo 500 etiquetas electrónicas. Con todos estos datos, es un riesgo que hay que tener en cuenta debido a la gran cifra de dinero que asciende.

La batería de las etiquetas electrónicas es otro riesgo, ya que no son recargables, se trata de baterías de litio que hay que pagar por ellas cuando se quiera adquirir nuevas baterías.

3.5 Solución propuesta

La solución propuesta va a constar de tres partes principales:

1. Las etiquetas electrónicas, para tomar una correcta elección se realizó un estudio de las principales empresas que ofrecen este servicio. Donde se valoró el precio de las etiquetas electrónicas, el precio de instalación, la atención al cliente, el rango de los Puntos de Acceso, y las características de las etiquetas. Finalmente nos decantamos por la empresa ETE la nº1 en la instalación de etiquetas electrónicas en España. Las características de estas etiquetas son:
 - Angulo de visión de 180°
 - Tres colores para la tinta
 - Respuesta rápida (menos de 15 segundos)
 - Posición vertical y horizontal
 - Tecnología E-Ink. Para una lectura similar al papel
 - Bluetooth 5.0 para el envío de información entre las etiquetas y los Puntos de Acceso
 - LED incorporado.
2. Para el desarrollo del backend se optó por realizar una API REST debido a sus ventajas:
 - Separación del cliente/servidor: si se quiere modificar, evolucionar, o refactorizar uno de los dos, se puede hacer de manera separada perfectamente.
 - Independencia de tecnologías / lenguajes: se puede desarrollar utilizando la tecnología y lenguaje que más se adapte a ti.
 - Fiabilidad, escalabilidad, flexibilidad: tanto la parte del cliente como la API puede crecer si las necesidades lo requieren, esto lo hace escalable. También la parte del front se puede enviar desde unos servidores y la API puede estar alojada en otro servidor totalmente independiente, esto lo hace flexible.
 - Experiencia de usuario: al hacer una solicitud al servidor lo que obtienes como respuesta son datos planos, que requieren tiempos de transferencia menores.
 - REST requiere menos recursos del servidor: ya que, al no mantener el estado, no requiere memoria.

Las tecnología utilizada para desarrollar la API han sido NodeJS ¹ que utiliza el lenguaje JavaScript.

3. Para el desarrollo del frontend, se ha decidido realizar una interfaz web, el principal motivo de esta decisión ha sido que me apasiona y me quiero dedicar en un futuro al desarrollar aplicaciones web. De esta forma, he mejorado y consolidado mis conocimientos.

3.6 Plan de trabajo

Como ya hemos visto en un apartado anterior, la metodología utilizada para desarrollar este proyecto es una metodología en cascada, compuesta por cinco etapas.

A continuación, veremos con detalle el trabajo que se ha realizado en cada etapa y lo representaremos en un diagrama de Gantt, que permite mostrar de una forma sencilla el tiempo de dedicación previsto para diferentes tareas o actividades a lo largo de un tiempo total determinado.

1. Análisis de requisitos: se realizó una reunión de empresa con el jefe del proyecto para analizar los requisitos y así acordar las necesidades que debe cumplir el proyecto.
2. Diseño: se acordó realizar una API REST con NodeJs y el frontend con Android para poder probar el funcionamiento.
3. Desarrollo: comenzar a desarrollar la API que envía la información a una etiqueta electrónica.
4. Verificación y pruebas: se realizan pruebas para ver que el sistema funciona correctamente, se le enseña el resultado al jefe del proyecto para que lo verifique y recibir feedback.
5. Mantenimiento: nos quedaremos a la espera una vez se implante el proyecto en la empresa para comprobar que el sistema funciona correctamente y evitar posibles fallos no detectados en la fase de verificación y pruebas.

¹ <https://nodejs.org/es/>

Implantación de un sistema de etiquetas electrónicas en una empresa de venta on-line



Ilustración 11. Diagrama de Gantt

3.7 Presupuesto

El desglose del tipo de trabajo y las horas que ha llevado cada uno es el siguiente:

1. Análisis de los requisitos: 20 horas.
2. Diseño: 25 horas.
3. Desarrollo: 235 horas.
4. Verificación y pruebas: 20 horas.
5. Mantenimiento: 0

Con la información que tenemos del salario medio de un informático en 2021, que son 20.400 €/año, obtenemos un precio de unos 11 €/hora. Por tanto, si el proyecto requiere una suma de 300 horas a 11€ por hora, el presupuesto del proyecto sería de 3.300€.

También habría que tener en cuenta el presupuesto de la empresa Aeme group de las etiquetas electrónicas, la suma de todos los productos y servicios ascendía a 6.810€, sin impuestos, con impuestos se quedaría un total de 8.240,10€.

CONCEPTO

500 etiquetas electrónicas de 4.2 + 5 APS repetidores + Accesorios	6.810,00 €
	<hr/>
	Importe base 6.810,00 €
	IVA 21% 1.430,10 €
	<hr/>
	TOTAL 8.240,10 €

Ilustración 12. Presupuesto total etiquetas electrónicas

Sumando ambos resultados el presupuesto total del proyecto sería de 11.540€.

4. Diseño de la solución

En este capítulo se explica la arquitectura del sistema, capa de persistencia, capa de presentación y las tecnologías utilizadas para desarrollar el proyecto. A continuación, se detalla cada uno de estos puntos:

4.1 Arquitectura del sistema

Relacionado a la arquitectura del sistema, se ha optado por desarrollar la parte del servidor (API) y la parte del cliente (frontend) individualmente, esta es la forma que aconseja el estilo arquitectónico en el que se basa la aplicación: REST. Esto permite a nuestra aplicación ser escalable y flexible.

La API se encarga de gestionar las peticiones del cliente, estará diseñada con varias rutas, dependiendo de la petición del cliente, esta entrará por la ruta correspondiente. La API se encargará de guardar la información necesaria en la BBDD, ya que se conecta a esta directamente y también se comunicará con la API de las etiquetas electrónicas para poder mostrar información en estas últimas.

Nuestra API mediante peticiones HTTP a la API de las etiqueta electrónicas le transmite la información al servidor de las etiquetas electrónicas. Este servidor a su vez está conectado con los Puntos de Acceso, y los Puntos de Acceso finalmente mandan la información a las etiquetas electrónicas.

El frontend, se comunica con la API a través de peticiones HTTP. Es el encargado tanto de enviar datos a la API, como de recibir datos para mostrarlos al usuario.

La encargada de almacenar los datos es MariaDB, que un sistema de gestión de bases de datos derivado de MySQL, se trata del sistema que tienen en la empresa. Contiene toda la información no volátil de la aplicación.

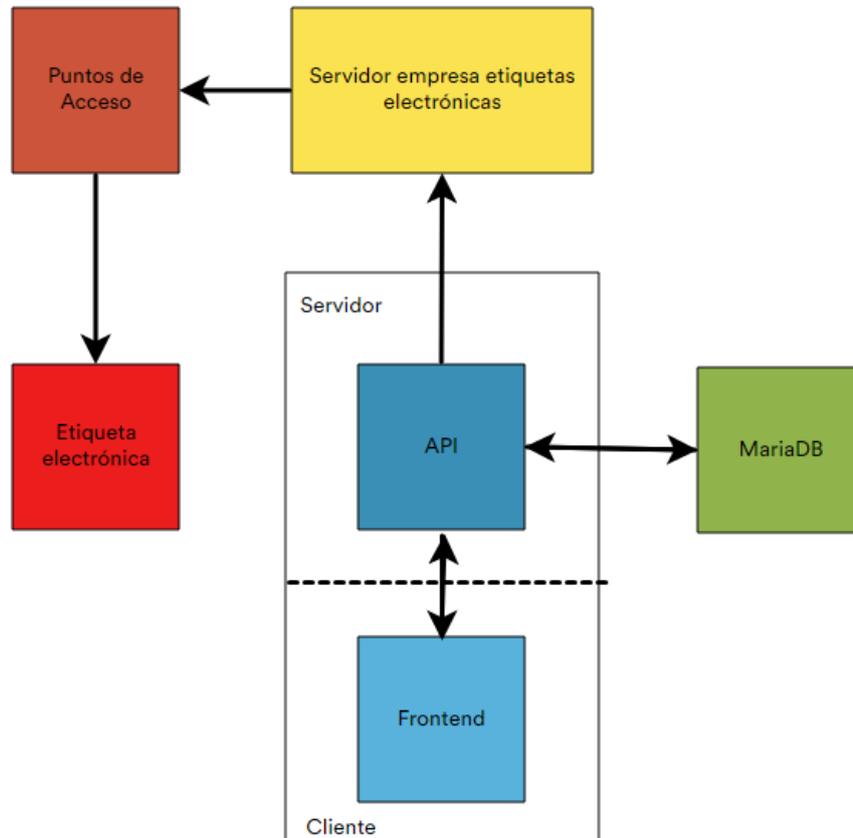


Ilustración 13. Componentes arquitectura sistema

4.2 Diseño detallado

Esta sección se divide en dos subsecciones, la capa de persistencia y la capa de presentación:

4.2.1 Capa de persistencia

En el siguiente punto se va a mostrar el modelo físico de la base de datos relacional. Debido a una cuestión de espacio únicamente se mostrarán las tablas y relaciones que sean necesarias para este proyecto, esto porque la BBDD de la empresa tiene un tamaño considerable y es innecesario mostrar todas las tablas que no vayan a tener relevancia para desarrollar el sistema.

- shelving (matrícula): tabla que almacena los identificadores de los carros. Un identificador esta formado por tres letras, por ejemplo: ABC.
- collection: una “collection” es un conjunto de “tickets”, es decir un conjunto de pedidos. La tabla que almacena los “collection” creados.

- ticket (pedido): un “ticket” representa un pedido de un cliente. De esta tabla sacaremos el cliente que ha realizado el pedido.
- ticketCollection: esta tabla almacena la relación entre “collection” y “ticket”. Es decir, que “tickets” forman “collection”. De esta tabla obtendremos el identificador del tique, el nivel del carro y el número de “wagon” donde debe ubicarse.
- agencyMode: representa la agencia encargada de transportar el pedido. Gracias a esta tabla podremos obtener la agencia que transporta el pedido y mostrarla en las etiquetas.
- smartTag: esta tabla no estaba en la BBDD, ha sido creada para representar la vinculación entre un carro y unas etiquetas. La tabla almacena el “code” que representa el identificador de la etiqueta, el “shelving” que representa el identificador del carro al que está vinculado la etiqueta y el “level”, que representa el nivel del carro en que está vinculada la etiqueta.
- worker: cada fila de esta tabla representa un trabajador de la empresa, y contiene datos personales como el nombre, apellidos, email, etc.
- workerShelving: esta tabla tampoco estaba creada en la BBDD, ha sido necesaria su creación para el proyecto. Esta tabla tiene tres columnas importantes, el “worker”, el “shelving” y el “collection”.

También se ha creado una vista SQL que básicamente es una tabla virtual cuyo contenido está definido por una consulta. La vista se llama collectionSmartTag y representa que etiquetas electrónicas han mostrado información de un tique y que información han mostrado.

collectionSmartTag

	ABC code	123 ticketFk	123 level	123 wagon	ABC shelvingFk	123 collectionFk	ABC agencyFk	123 clientFk
1	A0A3B8225B08	2.544.412	1	0	ZZZ	266	VN_REPARTO	4.870
2	A0A3B8224415	2.543.964	2	0	ZZZ	266	VN_REPARTO	8.794
3	A0A3B822475B	2.544.039	3	0	ZZZ	266	VN_REPARTO	5.852
4	A0A3B8224AEC	3.208.439	3	1	ABC	354.934	MRW 24h	4.080
5	A0A3B8224AEZ	3.209.193	2	2	ABD	354.934	MRW 24h	6.809
6	A0A3B8224AEX	3.209.626	3	2	ABD	354.934	MRW 24h	1.554
7	A0A3B8224DBF	3.211.326	1	1	ABC	354.934	MRW 24h	7.273
8	A0A3B8224AEB	3.211.370	2	1	ABC	354.934	MRW 24h	8.154
9	A0A3B822475B	3.228.222	3	1	ZZZ	368.091	MRW 24h	4.765
10	A0A3B8224415	3.234.669	2	1	ZZZ	368.091	MRW 24h	1.667
11	A0A3B8225B08	3.234.881	1	1	ZZZ	368.091	MRW 24h	3.649

Ilustración 14. Vista SQL

Por ejemplo, la etiqueta A0A3B8224DBF ha mostrado el tique 2544412 que ha sido pedido por el cliente 4870 y que repartirá la agencia de transporte VN_REPARTO. También muestra en que carro y nivel esta incrustada esa etiqueta electrónica, que es el carro con matrícula ABC y el nivel 1.

Implantación de un sistema de etiquetas electrónicas en una empresa de venta on-line

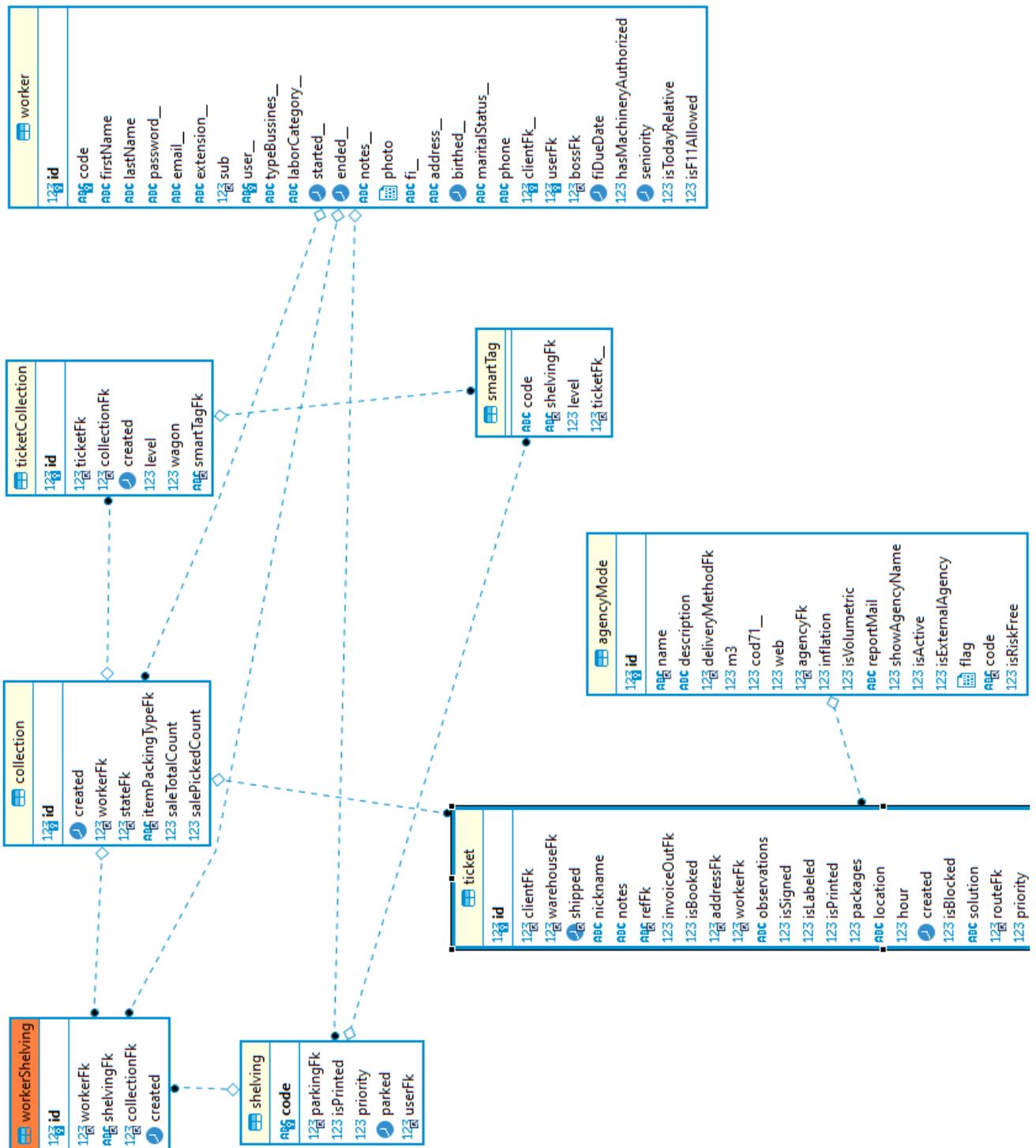


Ilustración 15. Modelo de la base de datos

4.2.2 Capa de presentación

Para la realización de los prototipos se ha decidido utilizar la página web “moqups.com”, aquí podemos observar los resultados:

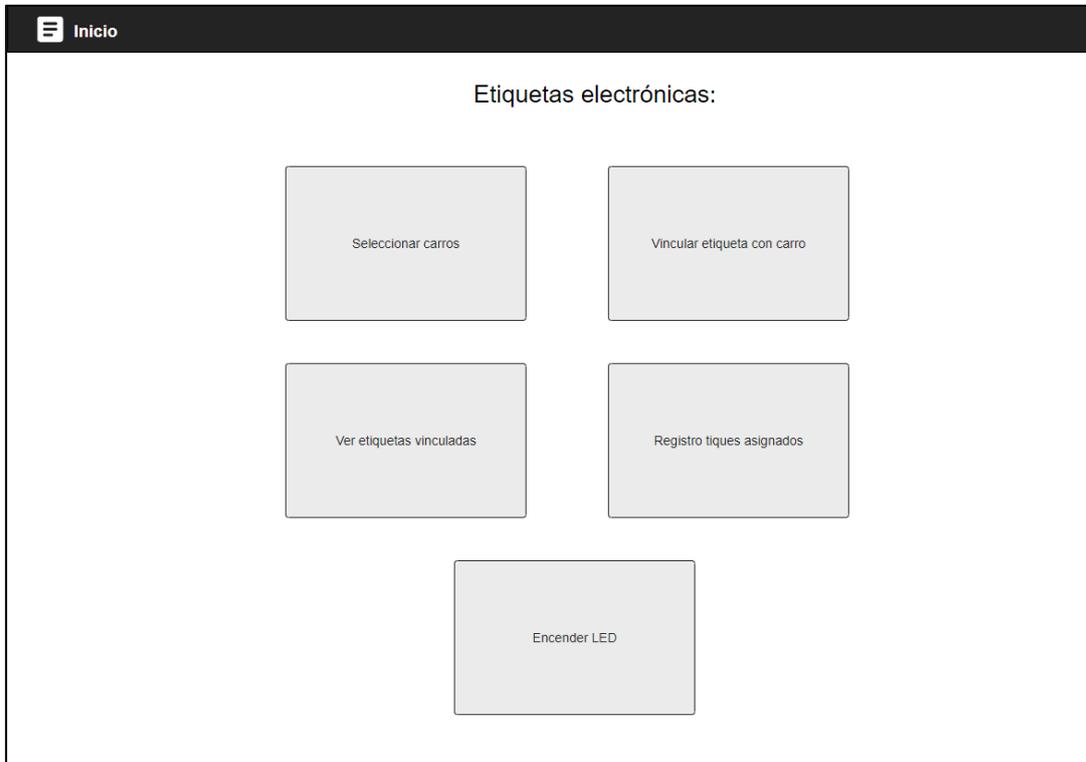


Ilustración 16. Prototipo “Inicio”

En la figura anterior, se presenta la página principal de la web. Tendrá una barra superior con un botón de inicio, para poder ir al inicio en cualquier momento. La página está compuesta principalmente por cinco botones:

1. Seleccionar carros
2. Vincular etiqueta con carro
3. Ver etiquetas vinculadas
4. Registro tiques asignados
5. Encender LED

Cada botón te redireccionará a otra página de la web.

Inicio

Introducir carros:

Carro 1:

Carro 2:

Carro 3:

...

Enviar

Detailed description: This is a web form titled 'Introducir carros:'. It features three text input fields labeled 'Carro 1:', 'Carro 2:', and 'Carro 3:'. Below these fields are three dots indicating a continuation of the list. At the bottom of the form is a button labeled 'Enviar'. The form is set against a white background with a dark header bar containing a menu icon and the word 'Inicio'.

Ilustración 17. Prototipo “Seleccionar carros”

Esta página se abrirá al hacer clic en “Seleccionar carros”. Podemos observar que hay tres campos de texto para introducir carros, sin embargo, el número de campos de texto puede variar, desde un carro hasta un máximo de cuatro carros. Donde ahora vemos tres puntos, habrá que añadir un botón para poder añadir más campos de texto, con el fin de introducir más carros. Por último, aparece un botón de “Enviar” para enviar los datos introducidos.

Inicio

Introducir datos:

Identificador etiqueta:

Matricula carro:

Nivel del carro:

Enviar

Detailed description: This is a web form titled 'Introducir datos:'. It contains three text input fields: 'Identificador etiqueta:', 'Matricula carro:', and 'Nivel del carro:'. Below the fields is a button labeled 'Enviar'. The form is set against a white background with a dark header bar containing a menu icon and the word 'Inicio'.

Ilustración 18. Prototipo “Vincular etiqueta con carro”

En la imagen anterior, observamos la página que se abrirá al hacer clic en “Vincular etiqueta con carro”. En esta página aparecerán tres campos de texto, una para la etiqueta electrónica, otro para el carro y finalmente uno más para el nivel del carro donde se va a ubicar la etiqueta electrónica. Por último, hay un botón de “Enviar” para enviar los datos que se han introducido.

Etiqueta electrónica	Matrícula carro	Nivel carro
A0A3B8224415	ABC	1
A0A3B8224420	ABC	2
A0A3B822475B	ABC	3

Ilustración 19. Prototipo “Ver etiquetas vinculadas”

Esta página hace referencia a “Ver etiquetas vinculadas”. Habrá tres columnas, una para la etiqueta electrónica, otra para la matrícula del carro, y una última para el nivel del carro.

Tique	Colección	Trabajador	Enviado	Entregado	Agencia Transporte	Cliente

Ilustración 20. Prototipo “Registro tiques asignados”

Cuando hagamos clic en el botón “Registro tiques asignados” de la página de inicio, nos redireccionará a esta página donde a través de una tabla. En las columnas mostraremos la información más importante de un tique, es decir: colección a la que pertenece, trabajador que ha preparado el pedido, fecha de envío, la fecha de entrega, agencia de transporte y cliente.



The image shows a web interface with a dark header bar containing a menu icon and the word 'Inicio'. Below the header, the main content area is white and contains the text 'Introducir tique:'. Underneath this text is a text input field with the label 'Identificador tique:'. Below the input field is a button with the text 'Enviar'.

Ilustración 21. Prototipo “Encender LED”

Esta página hace referencia a “Encender LED”, aparece un único campo de texto donde habrá que introducir el identificador de un tique. Por último, hay un botón de “Enviar” para enviar los datos que se han introducido.

4.3 Tecnología utilizada

La herramientas utilizadas para llevar a cabo este proyecto han sido:

JavaScript

JavaScript es un lenguaje de programación ligero, interpretado, o compilado justo-a-tiempo con funciones de primera clase. Si bien es más conocido como un lenguaje de scripting para páginas web, también es usado en muchos entornos fuera del navegador, tal como Node.js, Apache CouchDB y Adobe Acrobat. JavaScript es un lenguaje de programación basada en prototipos, multiparadigma, de un solo hilo, dinámico, con soporte para programación orientada a objetos, imperativa y declarativa.

Este lenguaje ha sido utilizado para desarrollar la parte del servidor, junto al entorno de desarrollo NodeJs. También se ha utilizado para desarrollar el frontend, para añadir funcionalidades a HTML.



Ilustración 22. Logo JavaScript

NodeJS

Node.js es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor (pero no limitándose a ello) basado en el lenguaje de programación JavaScript, asíncrono, con E/S de datos en una arquitectura orientada a eventos y basado en el motor V8 de Google. Algunas de las principales empresas del mundo utilizan Node en la producción, como Netflix, Paypal, Walmart y Uber

NodeJS junto a JavaScript, es la tecnología que se utiliza en VerdNatura para desarrollar su página web. Este ha sido el principal motivo de la utilización de estas herramientas.



Ilustración 23. Logo NodeJS

Express

Express ² es una infraestructura de aplicaciones web Node.js mínima y flexible que proporciona un conjunto sólido de características para las aplicaciones web y móviles.

Con miles de métodos de programa de utilidad HTTP y middleware a su disposición, la creación de una API sólida es rápida y sencilla.

Express es un framework de NodeJS que se encarga de gestionar las peticiones y enrutarlas. Estos han sido los motivos para su elección a la hora de desarrollar nuestra API.



Ilustración 24. Logo Express

² <https://expressjs.com/>

DBeaver

DBeaver ³ es una aplicación de software cliente de SQL y una herramienta de administración de bases de datos. Para las bases de datos relacionales, utiliza la interfaz de programación de aplicaciones JDBC para interactuar con las bases de datos a través de un controlador JDBC. Para otras bases de datos (NoSQL) utiliza controladores de bases de datos propietarios. Proporciona un editor que soporta el autocompletado de código y el resaltado de sintaxis.



Ilustración 25. Logo DBeaver

MariaDB

MariaDB ⁴ es un sistema de gestión de bases de datos derivado de MySQL con licencia GPL. Ha sido desarrollado por Michael Widenius, la fundación MariaDB y la comunidad de desarrolladores de software libre. Introduce dos motores de almacenamiento nuevos, uno llamado Aria y otro llamado XtraDB. Tiene una alta compatibilidad con MySQL ya que posee las mismas órdenes, interfaces, API y bibliotecas, siendo su objetivo poder cambiar un servidor por otro directamente.



Ilustración 26. Logo MariaDB

Visual Studio Code

Visual Studio Code es un editor de código fuente desarrollado por Microsoft para Windows, Linux y macOS. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código. También es personalizable, por lo que los usuarios pueden cambiar el tema del editor, los atajos de teclado y las preferencias. Es gratuito y de código abierto, aunque la descarga oficial está bajo software privativo e incluye características personalizadas por Microsoft. Visual Studio Code se basa en Electron, un framework que se utiliza para

³ <https://dbeaver.io/>

⁴ <https://es.wikipedia.org/wiki/MariaDB>

implementar Chromium y Node.js como aplicaciones para escritorio, que se ejecuta en el motor de diseño Blink.



Ilustración 27. Logo Visual Studio Code

Git

Git ⁵ es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia, la confiabilidad y compatibilidad del mantenimiento de versiones de aplicaciones cuando estas tienen un gran número de archivos de código fuente. Su propósito es llevar registro de los cambios en archivos de computadora incluyendo coordinar el trabajo que varias personas realizan sobre archivos compartidos en un repositorio de código.



Ilustración 28. Logo Git

HTML

Es el componente más básico de la Web. Define el significado y la estructura del contenido web. Además de HTML, generalmente se utilizan otras tecnologías para describir la apariencia/presentación de una página web (CSS) o la funcionalidad/comportamiento (JavaScript).



Ilustración 29. Logo HTML

Postman

⁵ <https://git-scm.com/>

Postman ⁶ es una aplicación para escritorio, que se utiliza para crear y utilizar APIs. Postman simplifica cada paso del ciclo de vida de la API y agiliza la colaboración para que pueda crear mejores API, más rápido.

Esta plataforma se ha utilizado para probar la API de la empresa contratada de etiquetas electrónicas. También nos ha sido útil para probar la API REST que hemos creado.



Ilustración 30. Logo Postman

5. Desarrollo de la solución propuesta

5.1 Preparación del entorno

La preparación del entorno es el primer paso a la hora de poner en marcha el desarrollo de nuestro proyecto, es más la preparación del entorno es importante en cualquier

⁶ <https://www.postman.com/>

proyecto de desarrollo de software, entonces lo primero que hay que instalar es NodeJS, y crear una carpeta donde trabajaremos. En esta nueva carpeta crearemos el archivo `package.json` el cual contiene la información básica de nuestro proyecto, incluyendo todas las librerías utilizadas.

```
package.json > {} dependencies > mysql
1  {
2    "name": "docapi",
3    "version": "1.0.0",
4    "description": "",
5    "main": "index.js",
6    > Debug
7    "scripts": {
8      "test": "echo \\\"Error: no test specified\\\" && exit 1"
9    },
10   "keywords": [],
11   "author": "",
12   "license": "ISC",
13   "dependencies": {
14     "eslint": "^7.22.0",
15     "eslint-config-google": "^0.14.0",
16     "express": "^4.17.1",
17     "got": "^11.8.2",
18     "mysql": "^2.18.1"
19   }
20 }
```

Ilustración 31. Archivo información proyecto

Las “dependencies” que vemos en la imagen de arriba son las librerías utilizadas en el proyecto. Estas librerías se han instalado gracias a npm que es un gestor de paquetes, este gestor permite tener cualquier librería disponible con solo una línea de código.

Las librerías que se han utilizado han sido:

4. eslint: se ha utilizado para la organización y formateo del código.
5. express: para creación y gestión de la API.
6. got: para las solicitudes HTTP.
7. mysql: para conectarse y realizar peticiones a la BBDD.

5.2 Archivo de configuración

Se ha creado el archivo `config.js`, esto con el fin de almacenar en este archivo la información confidencial como el usuario, password y storeId para realizar peticiones a la API de las etiquetas electrónicas. También el usuario y password para conectarse a la BBDD.

```
2 // /Datos esl
3 let password = 'XXXXXXXXXXXX';
4 let user = 'XXXXXXXXXXXX';
5 let storeId = 'XXXXXXXXXXXX';
6
7 // /Datos usuario
8 let userDev = 'XXXXXXXXXXXX';
9 let passwordDev = 'XXXXXXXXXXXX';
```

Ilustración 32. Archivo configuración

5.3 API REST

Como ya ha sido comentado a lo largo de la memoria, se ha desarrollado una API REST en NodeJS. De este modo, una vez se esté ejecutando el servidor en Node, la API empezará a funcionar y estará en todo momento escuchando cualquier petición que se realice a una de sus rutas.

Hay que mencionar que JavaScript es un lenguaje asíncrono, con las palabras reservadas “async” y “await”, podemos crear funciones asíncronas y esperar la respuesta en vez de seguir con la ejecución antes de que la llamada haya sido devuelta.

Las rutas implementadas en la API son las siguientes:

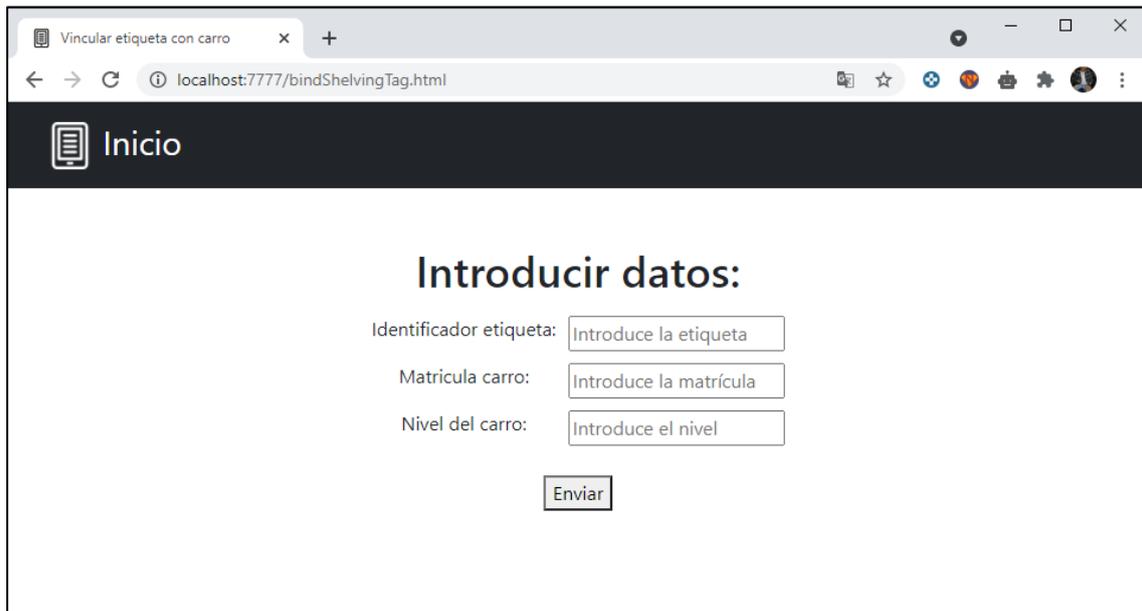
```
app.use('/bindShelving', require('./methods/bindShelvingTag')); // smartTagFk, sh
app.use('/flashOn', require('./methods/flashOn'));
app.use('/workerShelving', require('./methods/workerShelving'));
app.use('/collection_newSmartTag', require('./methods/collection_newSmartTag'));
app.use('/smartTag', require('./methods/smartTag')); // devuelve la tabla smarTag
app.use('/logTickets', require('./methods/logTickets'));
```

Ilustración 33. Rutas API

5.4 Flujo de control

Antes que nada y para que los siguientes puntos queden claros, vamos a describir cómo funcionará nuestra aplicación en su conjunto. La aplicación se divide en cinco partes principales:

- Vincular etiqueta con carro: Este paso se realizará al principio y una única vez por etiqueta electrónica. Este método se encarga de vincular una etiqueta con una balda (nivel) de un carro. Los carros suelen tener 4 baldas o niveles, con lo que cada carro tendrá 4 etiquetas electrónicas incrustadas. La solución propuesta es la siguiente:
 - Se ha desarrollado una interfaz web, donde un trabajador introducirá los datos necesarios para la vinculación de una etiqueta electrónica con un carro, que son identificador de una etiqueta, identificador de un carro y nivel del carro donde se incrustará la etiqueta.



The screenshot shows a web browser window with the title 'Vincular etiqueta con carro'. The address bar shows 'localhost:7777/bindShelvingTag.html'. The page has a dark header with a menu icon and the text 'Inicio'. The main content area is white and contains the heading 'Introducir datos:'. Below the heading are three input fields: 'Identificador etiqueta:' with the placeholder 'Introduce la etiqueta', 'Matricula carro:' with the placeholder 'Introduce la matrícula', and 'Nivel del carro:' with the placeholder 'Introduce el nivel'. Below these fields is a button labeled 'Enviar'.

Ilustración 34. Pantalla “Vincular etiqueta con carro”

- Una vez introducidos los datos, y se haya hecho clic en “Enviar”, se realizará una petición HTTP a la ruta **/bindShelving** de nuestra API REST que hemos creado previamente, pasándole los datos introducidos.
- Cuando nuestra API recibe la petición con los datos que ha introducido el trabajador, realiza los siguientes procesos:
 1. Comprueba que los tres datos recibidos tengan un formato correcto.
 2. Realiza una inserción en la BBDD concretamente en la tabla “smartTag” que vimos anteriormente en la capa de persistencia.
 3. Realiza dos peticiones a la API de las etiquetas electrónicas:

- La primera petición inserta lo que ellos llaman “artículo”, un “artículo” tiene varios campos como podemos ver en la imagen de abajo, el campo que nos interesa es el “barCode”, que será el identificador del carro junto al nivel, estos son los datos que ha introducido el trabajador anteriormente.

```
41 // Creamos un "artículo", que sera el que luego modificaremos para añadirle un ticket
42 async function insertShelving(shelvingId, smartTagFk) {
43   (async() => {
44     const info = await dataLogIn.dataLogIn;
45     let key = info.data.token;
46     let currentUser = info.data.currentUser;
47     (async() => {
48       const {body} = await got.post('http://app.etiquetaselectronicas.com:9999/item/batchImportItem', {
49         json: {
50           agencyId: currentUser.agencyId,
51           merchantId: currentUser.merchantId,
52           storeId: config.storeId,
53           unitName: currentUser.unitName,
54           itemList: [
55             {
56               attrCategory: 'verd natura', // Plantilla
57               attrName: 'withOutTicket', // Tipo de Plantilla
58               barCode: shelvingId, // Matricula + nivel --> Cod.Barras artículo
59               itemTitle: 'tagwithOutTicket',
60               qrCode: shelvingId
61             }
62           ]
63         },
64         responseType: 'json',
65         headers: {
66           'Authorization': key
67         }
68       });
69       await bindShelving(smartTagFk, shelvingId);
70       console.log('Insert Shelving:', body);
71     })();
72   })();
73 }
```

Ilustración 35. Petición HTTP para insertar artículo

- La segunda petición, que podemos ver en el código de abajo, realiza una vinculación de la etiqueta introducida con el “artículo” que acabamos de insertar arriba.

```
75 // Creamos la etiqueta con su vinculada con el ShelvingId al artículo
76 function bindShelving(smartTagFk, shelvingId) {
77   (async() => {
78     const info = await dataLogIn.dataLogIn;
79     let key = info.data.token;
80     (async() => {
81       const {body} = await got.post('http://app.etiquetaselectronicas.com:9999/bind/batchBind', { // eslint-disable-line
82         json: {
83           storeId: config.storeId,
84           tagItemBinds: [{
85             eslBarcode: smartTagFk,
86             itemBarcode: shelvingId
87           }
88         ]
89       },
90       responseType: 'json',
91       headers: {
92         'Authorization': key
93       }
94     });
95     console.log('Bind', body);
96   })();
97 })();
98 }
99 exports.bindShelving = bindShelving;
```

Ilustración 36. Petición HTTP para vincular una etiqueta con un artículo

- Mostrar información en una etiqueta: Este método se encarga de asignar un tique a una etiqueta electrónica para que se muestre la información deseada del tique. La solución propuesta es:
 - Se ha desarrollado una interfaz web, donde un trabajador introducirá el identificador de los carros que va a utilizar para sacar pedidos.

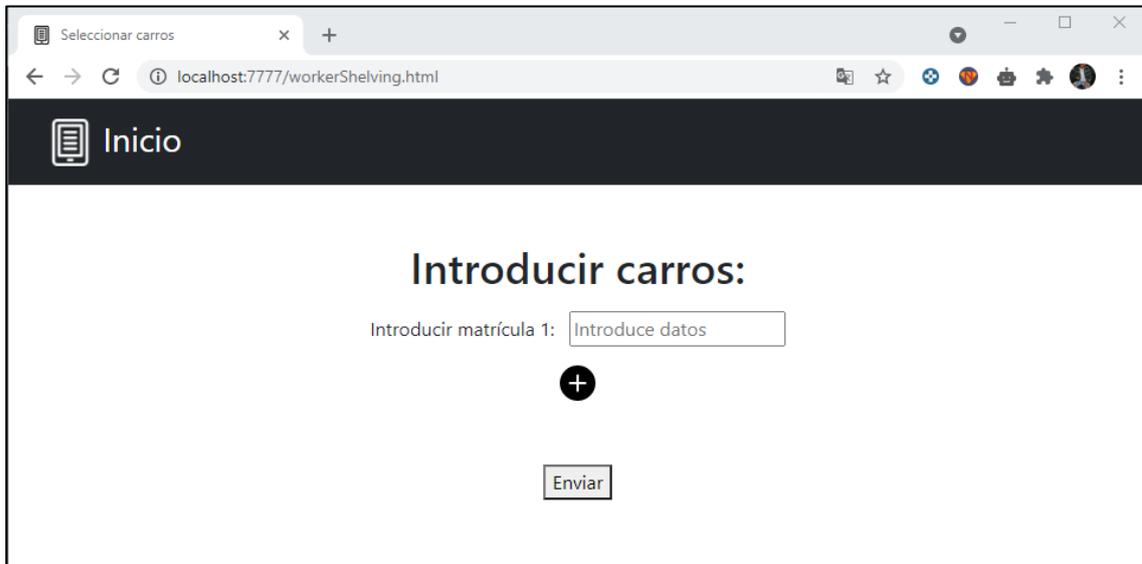


Ilustración 37. Pantalla “Mostrar información en una etiqueta”

- Una vez introducidos los datos, y se haya hecho clic en “Enviar”, se realizará una petición HTTP por cada identificador de carro introducido, a la ruta **/workerShelving** de nuestra API REST.
- También se realizará otra petición HTTP a la ruta **/collection_newSmartTag** de nuestra API.
- Cuando nuestra API recibe la petición a la ruta **/workerShelving** con los datos que ha introducido el trabajador, realiza el siguiente proceso:
 1. Realiza una inserción en la BBDD concretamente en la tabla “workerShelving” que vimos anteriormente en la capa de persistencia. Inserta el identificador de los carros que el trabajador ha introducido en la pantalla anterior.
- Cuando nuestra API recibe la petición a la ruta **/collection_newSmartTag** realiza los siguientes procesos:
 1. Realiza un procedimiento almacenado de la BBDD que se llama “collection_newSmartTag”, este procedimiento ha sido una modificación de un procedimiento anterior que ya existía en la BBDD de la empresa. El procedimiento calcula cuantos tiques caben en las tres baldas, según el número de carros que ha introducido el trabajador en la interfaz anterior. De esta forma, el

procedimiento devuelve una colección de tiques, y estos tiques serán los que se muestren en las etiquetas electrónicas.

2. El procedimiento “collection_newSmartTag” también rellena la vista “collectionSmartTag”, esta vista muestra los tiques que pertenecen a la colección que crea “collection_newSmartTag”.

	abc code	123 ticketFk	123 level	123 wagon	abc shelvingFk	123 collectionFk	abc agencyFk	123 clientFk
1	A0A3B8225B08	2.544.412	1	0	ZZZ	266	VN_REPARTO	4.870
2	A0A3B8224415	2.543.964	2	0	ZZZ	266	VN_REPARTO	8.794
3	A0A3B822475B	2.544.039	3	0	ZZZ	266	VN_REPARTO	5.852

Ilustración 38. Vista SQL “collection_newSmartTag”

3. Una vez ya tenemos los datos en la vista, se realiza una petición a la API de las etiquetas electrónicas. Esta petición lo que hace es actualizar el “artículo” que se había creado en método anterior insertando en el “artículo” todos los datos deseados. Una vez actualizado el “artículo” las etiquetas electrónicas ya mostrarán información por pantalla.

```
(async() => {
  for (let i = 0; i < consultaSql.length; i++) {
    console.log('SHELIVING: ', consultaSql[i].shelvingFk, consultaSql[i].level);
    const {body} = await got.post('http://app.etiquetaselectronicas.com:9999/item/batchImportItem', {
      json: {
        agencyId: currentUser.agencyId,
        merchantId: currentUser.merchantId,
        storeId: config.storeId,
        unitName: currentUser.unitName,
        itemList: [
          {
            attrCategory: 'verdnaturala',
            attrName: 'withTicket',
            barCode: consultaSql[i].shelvingFk + consultaSql[i].level, // Matricula + nivel
            itemTitle: 'tagWithTicket',
            productCode: consultaSql[i].ticketFk, // Ticket
            qrCode: consultaSql[i].ticketFk,
            custFeature1: consultaSql[i].clientFk, // Cliente
            custFeature2: consultaSql[i].agencyFk, // Agencia de transporte
            custFeature3: '', // consultaSql[i].workerFk, // Trabajador
            custFeature4: consultaSql[i].wagon, // Wagon
          }
        ]
      },
      responseType: 'json',
      headers: {
        'Authorization': key
      }
    });
    // insertTicketDB.insertTicket(consultaSql[i].ticketFk, consultaSql[i].code);
    console.log(i, body);
  }
})();
```

Ilustración 39. Petición actualizar artículo

- **Encender LED:** El uso del LED en mi proyecto será ligeramente distinto al mencionado anteriormente, debido a que no podemos saber a tique pertenece un producto. De forma que nosotros deberemos introducir directamente el identificador del tique, para encender el LED de la etiqueta electrónica. La solución se ha desarrollado:
 - Se ha desarrollado una interfaz web, donde se introducirá los datos necesarios para encender el LED de una etiqueta, en esta caso solo se necesita el identificador de un tique.



Ilustración 40. Pantalla “Encender LED”

- Una vez introducidos los datos, y se haya hecho clic en “Enviar”, se realizará una petición HTTP a la ruta **/flashOn** de nuestra API REST pasándole los datos introducidos.
- Cuando nuestra API recibe la petición con los datos que ha introducido el trabajador, realiza el siguiente proceso:
 1. A partir de identificador del tique, realiza una consulta a la BBDD con el fin de obtener la etiqueta electrónica que debe encender.
 2. Realiza una peticiones a la API de las etiquetas electrónicas para encender el LED de la etiqueta electrónica que se obtiene en el paso anterior:

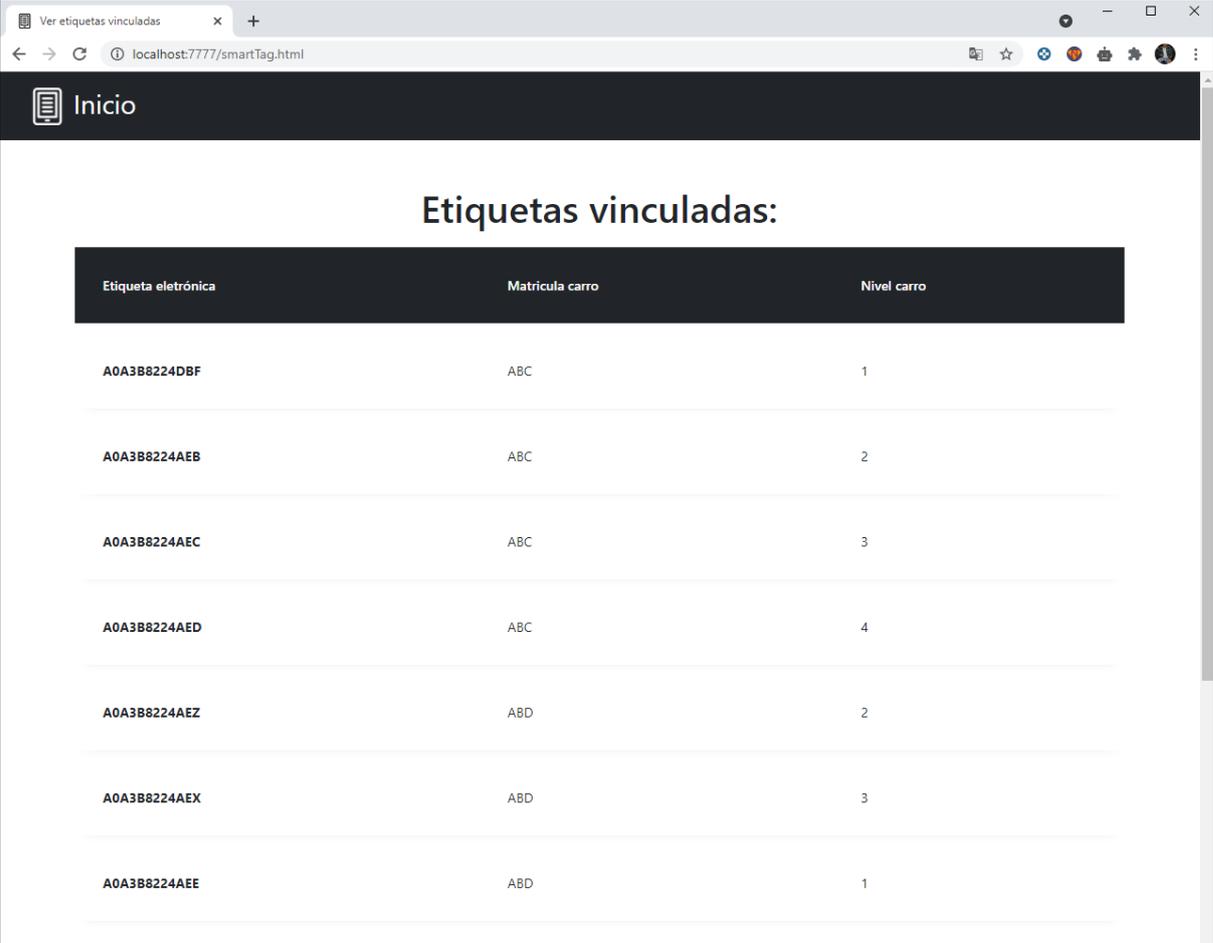
```
function insertFlash(qrySql) {
  // console.log('CONSULTASQL_TAMAÑO:', consultaSql.length);
  const dataLogIn = require('./server');
  const got = require('got');
  (async() => {
    const info = await dataLogIn.dataLogIn;
    let key = info.data.token;

    (async() => {
      const {body} = await got.post('http://app.etiquetaselectronicas.com:9999/led/excuteLed', {
        json: {
          storeId: config.storeId,
          lightColor: 8,
          lightTime: 30,
          targetExecution: qrySql[0].shelvingFk + qrySql[0].level + '/tagWithTicket',
          lightFrequency: 1
        },
        responseType: 'json',
        headers: {
          'Authorization': key
        }
      });
      console.log(qrySql[0].shelvingFk + qrySql[0].level);
      console.log(body);
    })();
  })();
}
```

Ilustración 41. Petición HTTP para encender LED

Implantación de un sistema de etiquetas electrónicas en una empresa de venta on-line

- Ver etiquetas vinculadas: Mediante una tabla representaremos las etiquetas electrónicas vinculadas a un carro. De esta forma podremos saber en todo momento a que carro y nivel está asociada una etiqueta electrónica. La solución propuesta se explica a continuación:
 - Se ha desarrollado una interfaz web, que al cargarse realiza una petición HTTP a la ruta **/smartTag** de nuestra API. A continuación, ya podremos observar los siguientes datos:



Etiqueta electrónica	Matricula carro	Nivel carro
A0A3B8224DBF	ABC	1
A0A3B8224AEB	ABC	2
A0A3B8224AEC	ABC	3
A0A3B8224AED	ABC	4
A0A3B8224AEZ	ABD	2
A0A3B8224AEX	ABD	3
A0A3B8224AEE	ABD	1

Ilustración 42. Pantalla “Ver etiquetas vinculadas”

- Cuando nuestra API recibe la petición, realiza los siguientes procesos:
 1. Realiza una consulta a la tabla “smartTag”.

- **Registro tiques asignados:** A través de una tabla, representaremos la un registro de todos los tiques que han sido mostrados en etiquetas electrónicas hasta la fecha actual. Para la solución se ha desarrollado:
 - Una interfaz web, que al cargarse realiza una petición HTTP a la ruta **/logTickets** de nuestra API. A continuación, ya podremos ver la siguiente tabla por pantalla:

The screenshot shows a web browser window with the URL 'localhost:7777/logTickets.html'. The page has a dark header with 'Inicio' and a main content area with the title 'Registro tiques asignados:'. Below the title is a table with the following data:

Tique	Colección	Etiqueta electrónica	Trabajador	Enviado	Entregado	Agencia Transporte	Cliente
2544412	266	A0A3B8225B08	pako	2020-01-13T23:00:00.000Z	2020-01-14T23:00:00.000Z	VN_REPARTO	4870
2543964	266	A0A3B8224415	pako	2020-01-13T23:00:00.000Z	2020-01-14T23:00:00.000Z	VN_REPARTO	8794
2544039	266	A0A3B822475B	pako	2020-01-13T23:00:00.000Z	2020-01-14T23:00:00.000Z	VN_REPARTO	5852
3208439	354934	A0A3B8224AEC	jubeda	2021-07-26T22:00:00.000Z	2021-07-27T22:00:00.000Z	MRW 24h	4080
3209193	354934	A0A3B8224AEZ	jubeda	2021-07-26T22:00:00.000Z	2021-07-27T22:00:00.000Z	MRW 24h	6809
3209626	354934	A0A3B8224AEX	jubeda	2021-07-26T22:00:00.000Z	2021-07-27T22:00:00.000Z	MRW 24h	1554

Ilustración 43. Pantalla “Registro tiques asignados”

- Cuando nuestra API recibe la petición, realiza los siguientes procesos:
 1. Realiza una consulta a la vista “collectionSmartTag”.

5.5 Authorization

Como se puede observar en la imagen de abajo, cada vez que se realiza una de las peticiones HTTP, mencionadas con anterioridad, a la API de las etiquetas, es necesario incluir en el “headers” una “Authorization”. Esta es una forma de verificar que quien realiza la petición al servidor realmente tiene los privilegios para llevarla a cabo.

```

async function bindShelving(smartTagFk, shelvingId) {
  (async() => {
    const info = await dataLogIn.dataLogIn;
    let key = info.data.token;
    (async() => {
      const {body} = await got.post('http://app.etiquetaselectronicas.com:9999/bind/batchBind', { // eslint-disable-line
        json: {
          storeId: config.storeId,
          tagItemBinds: [{
            eslBarcode: smartTagFk,
            itemBarcode: shelvingId
          }
        ]
      },
      responseType: 'json',
      headers: {
        'Authorization': key
      }
    });
    console.log('Bind', body);
  })();
})();
}

```

Ilustración 44. Petición HTTP con “Authorization”

Esta “Authorization” se consigue realizando otra petición a la API de las etiquetas para obtener una Public Key, una vez obtenida la Public Key y junto a una clave privada que ya había sido proporcionada con anterioridad, se encriptan ambas obteniendo la nueva “Authorization” que será la que se le envíe en cada petición a la API de las etiquetas electrónicas. La encriptación se realiza gracias a la librería crypto.

```

server.js  logIn.js  encrypted.js  bindShelvingTag.js  bindTicketShelving.js  selectCollection.js
utilities > encrypted.js > encryptPassword
1  const crypto = require('crypto');
2  const got = require('got');
3  // const config = require('../config');
4
5  const encryptPassword = async password => {
6    const {body} = await got.get('http://app.etiquetaselectronicas.com:9999/user/getErpPublicKey', {
7    });
8    const publicKey = `-----BEGIN PUBLIC KEY-----\n${JSON.parse(body).data}\n-----END PUBLIC KEY-----`;
9    const externKey = {
10     key: publicKey,
11     padding: crypto.constants.RSA_PKCS1_PADDING
12   };
13   let buffer = Buffer.from(password);
14   return crypto.publicEncrypt(externKey, buffer).toString('base64');
15 };
16
17 exports.encryptPassword = encryptPassword;

```

Ilustración 45. Obtención de la “Authorization”

5.6 Diseño pantalla etiqueta electrónica

El diseño de la información de las etiquetas electrónicas se ha realizado de forma visual a través de una página web que nos ofrecían la empresa de las etiquetas electrónicas. La información que muestra es la siguiente: el identificador de trabajador que va a preparar el pedido, el cliente que ha realizado el pedido, la agencia que va a transportar el pedido, y por supuesto el identificador del pedido que es el tique. También incluye un código QR y un código de barras con el identificador del pedido, para poder leerlo a través de un lector.

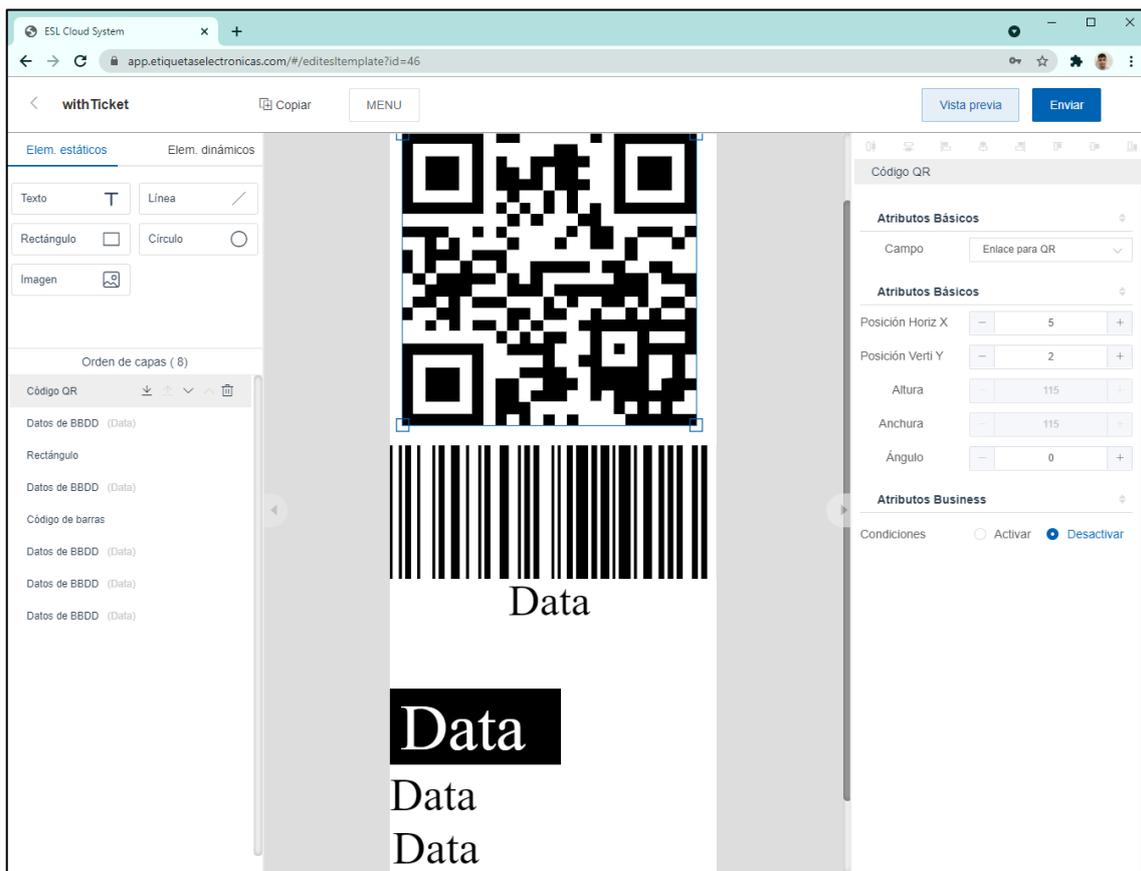


Ilustración 46. Diseño pantalla etiqueta electrónica

6. Implantación

El despliegue de nuestra aplicación web, se ha distribuido en dos servidores diferentes:

Por una parte, utilizando NodeJS junto con su framework más famoso denominado Express, se ha creado un servidor local donde está ubicada la API REST, junto con todos los archivos estáticos como archivos CSS, HTML e imágenes.

```
server.js > ...
1  const express = require('express');
2  const app = express();
3  const port = 7777;
4  const logIn = require('./methods/logIn');
5  const con = require('./db/connect');
6
7  const dataLogIn = logIn.info;
8  exports.dataLogIn = dataLogIn;
9
10 con.con.connect(function(err) {});
11
12 app.use('/bindShelving', require('./methods/bindShelvingTag')); // smartTagFk, shelving, level
13 app.use('/flashOn', require('./methods/flashOn'));
14 app.use('/workerShelving', require('./methods/workerShelving'));
15 app.use('/collection_newSmartTag', require('./methods/collection_newSmartTag')); // devuelve la vista collectionSmartTag
16 app.use('/smartTag', require('./methods/smartTag')); // devuelve la tabla smarTag
17 app.use('/logTickets', require('./methods/logTickets'));
18
19 app.use(express.static('front'));
20
21 app.listen(port, () => {
22   console.log(`Example app listening at http://localhost:${port}`);
23 });
```

Ilustración 47. Archivo para crear servidor

En la captura anterior podemos observar cómo se crea el servidor. La primera y segunda línea incluyen el módulo de Express, a través de la orden “require()”, y a continuación crean un aplicación de Express, llamada “app”. En las tres últimas líneas de código, se define y crea el servidor, que estará escuchando en puerto 7777. Cuando el servidor está en ejecución, se puede acceder a la dirección <http://localhost:7777> en un navegador para interactuar con el servidor.

En la línea diecinueve observamos una función de middleware, llamada “express.static”, esta función se utiliza para servir los archivos estáticos que acabamos de mencionar arriba. Mientras que de la línea doce a la diecisiete observamos las rutas que se llamarán cuando reciban una petición HTTP.

Por otra parte, la BBDD de MySQL está ubicada en el servidor de la empresa. Hay que tener en cuenta que para poder acceder a la BBDD necesitamos estar físicamente en la empresa o bien se puede acceder remotamente utilizando una VPN.

7. Pruebas

Para comprobar el correcto funcionamiento de la aplicación web desarrollada, primero se ha inicializado el servidor de NodeJS con el comando “node server.js”.

```
PS C:\Users\vicent\Documents\Proyectos\tfg2> node server.js
Example app listening at http://localhost:7777
```

Ilustración 48. Servidor corriendo

A continuación, se ha puesto en marcha un proceso donde navegamos por todas las secciones de la web y vamos comprobando que todas las funcionalidades de la aplicación funcionen correctamente.

7.1 Inicio

Hemos comenzado por la página de inicio comprobando que todos los botón te redireccionaban a la página correcta.



Ilustración 49. Prueba pantalla “Inicio”

7.2 Caso de Uso 1: Vincular etiqueta con carro

Seguidamente hemos probado el caso de uso “Vincular etiqueta con carro”, hemos vinculado tres etiquetas en el carro ABC, en los niveles 1,2 y 3. Comprobamos que efectivamente esta información se guarda en la BBDD, y que las etiquetas electrónicas muestran el identificador del carro y el nivel en el que se encuentran.



Inicio

Introducir datos:

Identificador etiqueta:

Matricula carro:

Nivel del carro:

Ilustración 50. Prueba pantalla “Vincular etiqueta con carro”



Ilustración 51. Etiquetas mostrando el carro y nivel

7.3 Caso de Uso 2: Insertar tique en etiqueta

A continuación, hemos comprobado la funcionalidad de “Insertar un tique en una etiqueta”, hemos seleccionado el carro ABC al que le hemos vinculado las etiquetas electrónicas en paso anterior. De modo que ahora las tres etiquetas muestran la información de un tique distinto cada una.



The screenshot shows a mobile application interface. At the top, there is a dark header with a menu icon and the word "Inicio". Below the header, the main content area is white and contains the following elements: the title "Introducir carros:" in bold; a text input field labeled "Introducir matrícula 1:" with the value "ABC"; a plus sign icon in a black circle; and a button labeled "Enviar".

Ilustración 52. Prueba pantalla “Mostrar información en una etiqueta”



Ilustración 53. Etiquetas mostrando tiques

7.4 Caso de Uso 3: Ver etiquetas vinculadas

La siguiente funcionalidad para comprobar es “Ver etiquetas vinculadas”, y comprobamos que las etiquetas que muestra por pantalla concuerdan con las que hemos vinculado anteriormente.

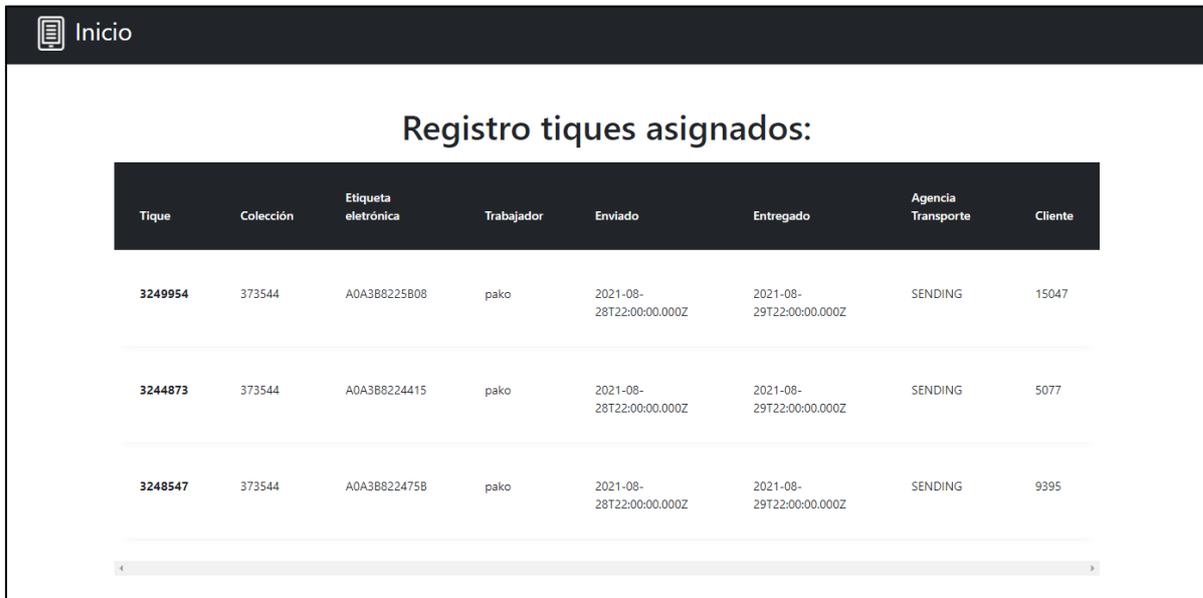


Etiqueta electrónica	Matricula carro	Nivel carro
A0A3B8225B08	ABC	1
A0A3B8224415	ABC	2
A0A3B822475B	ABC	3

Ilustración 54. Prueba pantalla “Ver etiquetas vinculadas”

7.5 Caso de Uso 4: Ver registro de tiques asignados

En la imagen de abajo comprobamos que el caso de uso “Ver registro de tiques asignados” funciona correctamente, ya que por pantalla podemos ver todos los tiques, junto algunos datos de interés, que se han mostrados en alguna etiqueta electrónica.



Inicio

Registro tiques asignados:

Tique	Colección	Etiqueta electrónica	Trabajador	Enviado	Entregado	Agencia Transporte	Cliente
3249954	373544	A0A3B8225B08	pako	2021-08-28T22:00:00.000Z	2021-08-29T22:00:00.000Z	SENDING	15047
3244873	373544	A0A3B8224415	pako	2021-08-28T22:00:00.000Z	2021-08-29T22:00:00.000Z	SENDING	5077
3248547	373544	A0A3B822475B	pako	2021-08-28T22:00:00.000Z	2021-08-29T22:00:00.000Z	SENDING	9395

Ilustración 55. Prueba pantalla “Registro tiques asignados”

7.6 Caso de Uso 5: Encender LED

La siguiente funcionalidad para comprobar es “Encender LED”, para ello hemos introducido el tique que está mostrando una etiqueta electrónica y pasados unos segundos comienza a parpadear el LED correctamente.



The screenshot shows a web application interface. At the top left, there is a dark header with a white icon of a document and the text "Inicio". The main content area is white and contains the text "Introducir tique:" in a large, bold font. Below this, there is a label "Identificador tique:" followed by a text input field containing the number "3249954". Below the input field is a button labeled "Enviar".

Ilustración 56. Prueba pantalla “Encender LED”



Ilustración 57. Etiqueta con el LED encendido

8. Conclusiones

Respecto a los objetivos que se pretendían conseguir para este proyecto, podemos concluir que se han cumplido satisfactoriamente. Ya que se ha conseguido implementar una API capaz de interactuar con un cliente, otra API y la BBDD. También se ha conseguido implementar una interfaz gráfica para mostrar datos y para poder interactuar con la API que he desarrollado. Todo esto en conjunto de objetivos ha permitido alcanzar el objetivo principal que era poner en funcionamiento unas etiquetas electrónicas.

Este proyecto me ha permitido conocer y aprender herramientas como NodeJS, Express y Postman, al mismo tiempo que reforzar mis conocimientos sobre otras herramientas y tecnologías que ya conocía como lo son Visual Studio Code, JavaScript, SQL, HTML y CSS.

También quiero comentar que este proyecto ha sido todo un reto para mí, debido a que nunca me había enfrentado solo al desarrollo de un proyecto de esta envergadura. Pero tras meses de trabajo duro y dedicación pienso que he cumplido con mis expectativas sobre el proyecto.

8.1 Relación del trabajo desarrollado con los estudios cursado

Este trabajo de final de grado ha sido sin lugar a duda una enorme oportunidad para poder aplicar mis conocimientos adquiridos a lo largo de la carrera de ingeniería informática. Algunas de las asignaturas que más útiles han sido para el desarrollo del proyecto han sido *Interfaces persona computador*, *Bases de datos y sistemas de información*, *Estructuras de datos y algoritmos*, *Ingeniería del software*, *Programación*.

8.2 Trabajos a futuro

Para obtener un buen feedback y un mejor desarrollo de la API hubiera sido perfecto poner en marcha las etiquetas electrónicas en funcionamiento en la empresa, sin embargo, debido a la falta de tiempo esto no ha sido posible.

Aunque como voy a continuar trabajando para esta empresa y en un futuro las etiquetas se pondrán en funcionamiento podré corregir todos los errores que vayan surgiendo, llevar el mantenimiento de la aplicación, así como aplicar nuevas funcionalidades que se requieran.

9. Referencias

- [1] *ETE Etiquetas Electrónicas*. <<https://etiquetaselectronicas.com/>> [Consulta: 20 de junio de 2021]
- [2] *Etiquetas electrónicas*. <https://es.wikipedia.org/wiki/Etiquetas_electr%C3%B3nicas> [Consulta: 20 de junio de 2021]
- [3] *Ventajas API REST*. <<https://desarrolloweb.com/articulos/ventajas-inconvenientes-apirest-desarrollo.html>> Consulta: [1 de julio de 2021]
- [4] *Salario promedio informático*. <<https://es.indeed.com/career/informatico/salaries>> Consulta: [5 de julio de 2021]
- [5] *Servicio de archivos estáticos en Express*. <<https://expressjs.com/es/starter/static-files.html>> Consulta: [20 de julio de 2021]
- [6] *Introducción a Express/Node*. <https://developer.mozilla.org/es/docs/Learn/Server-side/Express_Nodejs/Introduction> Consulta: [15 de julio de 2021]
- [7] *W3schools*. <<https://www.w3schools.com/>> Consulta: [16 de julio de 2021]
- [8] *Manual de HTML*. <<https://desarrolloweb.com/manuales/manual-html.html>> Consulta: [29 de julio de 2021]
- [9] *Javascript Asíncrono*. <<https://lemoncode.net/lemoncode-blog/2018/1/29/javascript-asincrono>> Consulta: [3 de julio de 2021]
- [10] *El señor de las flores da paso a su gran familia*. <<https://www.levante-emv.com/el-mercantil-valenciano/2021/03/28/or-flores-da-paso-gran-45941156.html>> Consulta: [3 de julio de 2021]
- [11] *Introducción a JSON*. <<https://www.json.org/json-es.html>> Consulta: [22 de junio de 2021]
- [12] *Criptografía asimétrica*. <https://es.wikipedia.org/wiki/Criptograf%C3%ADa_asim%C3%A9trica> Consulta: [19 de junio de 2021]
- [13] MATTHIAS, B (2016). *RESTful API Design: Best Practices in API Design with REST*.
- [14] RAUSCHMAYER, A (2019). *JavaScript for impatient programmers*

10. Términos

1. **Picking:** es un proceso básico en la preparación de pedidos en los almacenes que afecta en gran medida a la productividad de toda la cadena logística. Consiste en la recogida de material extrayendo unidades o conjuntos empaquetados de una unidad de empaquetado superior que contiene más unidades que las extraídas.
2. **API:** La interfaz de programación de aplicaciones, conocida también por la sigla API, es un conjunto de subrutinas, funciones y procedimientos que ofrece cierta biblioteca para ser utilizada por otro software como una capa de abstracción.
3. **PDA:** ordenador de bolsillo, organizador personal o agenda electrónica de bolsillo, es una computadora de mano.