
Development of a Digital System for the Traceability of the Horticultural Supply Chain Products.

J. Díez-Cascón Martínez

ICAI – Universidad Pontificia Comillas

ABSTRACT The development of a cloud-based platform for traceability of fruits and vegetables [1] throughout the food supply chain is presented. The main objective of this platform is to improve food safety, control product quality and conditions, and ensure transparency of the collected data. To achieve this, laser branding technology, QR codes, and RFID tags are used to track transport pallets from their origin to the final consumer. This platform is based on Amazon Web Services (AWS) and targets various industry stakeholders, such as agricultural cooperatives, logistics companies, large retailers, insurance companies, exporting firms, and end consumers. In the logistics of perishable products, it is essential to manage and monitor temperature and humidity using suitable equipment. The platform allows for analysis, storage, and access to data generated during processing, transportation, and commercialization. For this purpose, the Amazon RDS relational database is utilized to store and manage product information.

KEY WORDS Traceability, Fruits, Vegetables, RFID, IoT, QR.

I. INTRODUCTION

In today's society, there is a growing concern about the origin and quality of food products. Consumers seek transparency and safety. Therefore, they want to know where their food comes from, how it has been produced, and what treatments it has undergone. Recognizing this need, this Master's Thesis proposes a project aimed at developing a cloud-based platform that allows traceability of fruits and vegetables from their origin to the final consumer.

To achieve this goal, laser technology, QR codes, and RFID tags are used to identify and gather relevant information about each piece of fruit throughout the entire supply chain. The platform, based on Amazon Web Services (AWS)[2], targets two types of users: the platform's clients (agricultural cooperatives, logistics companies, large retailers, exporting firms, etc.) and the end consumers of the products. Depending on the user type, detailed information will be provided about the fruits and vegetables, from their place of origin to the treatments, processes, and transportation methods they have undergone, including quality certifications from the orchards where they were cultivated.

The main objective is to ensure transparency and food safety while providing greater control over the quality and conditions of fruits and vegetables. The implementation of a traceability system allows for the identification of any problems that may arise during the logistics process. For example, if there is a break in the cold chain, it will be

possible to determine where and why it occurred, which could lead to obtaining more cost-effective insurance for farmers or large retailers, as they can determine the cause and the responsible party for the deterioration. Additionally, products that have ripened more due to a temperature maintenance failure can be prioritized for sale. Therefore, this project represents a significant advancement in consumer protection and the sustainability of the agri-food sector.

It is important to note that recent bans on plastic packaging will require producers and suppliers to adopt plastic-free solutions, which will drive the demand for laser labeling (Natural Branding) and biodegradable packaging.

II. LASER ENGRAVING

In Europe, approximately 25.8 million tons of post-consumer plastic waste are generated each year, of which only 30% is recycled. This means that between 5 and 13 million tons of plastic end up in the environment, causing serious environmental impacts and endangering ecosystem health. Faced with this alarming issue, it is essential to seek innovative solutions that help reduce the amount of waste generated and promote more sustainable practices in the agri-food industry.

In this context, laser labeling on the skin[3] of fruits and vegetables emerges as a promising alternative. This revolutionary technology allows for direct marking on the skin of products, eliminating the need for additional labels,

wrappers, or plastic materials. Instead, a carbon dioxide laser is used to perform a smooth and precise engraving on the surface of the skin without affecting the quality or taste of the food.



FIGURE 1. LASER ENGRAVINGS ON FRUITS [4]

Spain positions itself as a pioneering country in the development and application of this technique, thanks to the Valencian company Laserfood, which has developed all the necessary technology to carry out laser labeling of fresh fruits and vegetables. Through this process, a durable and legible marking is achieved without compromising the quality of the product or food safety.

Laser labeling is beneficial because it significantly contributes to the reduction of plastic waste in the agri-food chain. By eliminating the need for additional wrappers and labels, the amount of waste generated is reduced, helping to preserve natural resources and mitigate the environmental impacts associated with excessive plastic use.

In terms of costs, laser labeling is estimated to cost approximately €0.90 per thousand marks, compared to the €15 required for conventional labeling. This represents significant savings for companies while promoting more sustainable practices and reducing the carbon footprint by 99.5%.

Another important advantage is the possibility of engraving QR codes directly on the skin of fruits and vegetables. These codes provide an ID for each fruit, linking it to detailed information about the product, such as its origin, harvest date, cultivation method, supplier, and certifications. This facilitates consumer access to relevant information, promoting greater transparency and trust in the food production and distribution process.

The legibility of QR codes engraved through laser labeling is high, thanks to the precision and durability of this technology. Mobile devices can easily scan the codes, allowing consumers to obtain nutritional information, storage recommendations, and even culinary recipes

associated with the product, which positively influences consumer decision-making.

Since fruits are not always sold in bulk, despite that being the current legal requirement, other marketing methodologies such as selling individual consumer packaging should also be considered, which must also be labeled. Therefore, to ensure safety and comply with regulations in fruit and vegetable packaging plants, it is essential to have coding and marking systems that can adapt to different packaging formats and carefully select the inks used, taking into account regulations and recommendations that prohibit the transfer of unauthorized substances from the packaging to the food.



FIGURE 2. CONTAINER ENGRAVED WITH FOOD GRADE INK [5]

III. RFID TECHNOLOGY

RFID (Radio Frequency Identification) technology has significantly evolved since its invention during World War II. Nowadays, its widespread adoption has led to a considerable cost reduction, with adhesive tags available at affordable prices. This technology offers numerous advantages in terms of product traceability and tracking, making it highly beneficial across various sectors.

RFID technology enables remote identification of products without human intervention, streamlining processes and minimizing errors. This technology has different variants, depending on factors such as operating frequency and standards used.



FIGURE 3. RFID[6]

In logistics and retail, high-frequency (UHF 860-920MHz) RFID technology is commonly used as it allows for reading distances of up to 20 meters with the appropriate equipment. These tags consist of a thin aluminum antenna and a memory chip attached to it. There are various chip manufacturers that comply with the EPC Class1 Gen2 [7] standard, providing flexibility in the development of projects based on this technology.

The existence of standards in RFID technology ensures compatibility between tags and readers, making it easier to implement in different environments. The size, material, and shape of the tags are related to their application and the material they will be applied on. For example, in logistics, tags that maximize reading distance and minimize errors are used, while in other cases, high-quality reading with a limited range is desired.

In conclusion, the use of RFID tags on pallets and the boxes contained within them offers an efficient and accurate solution for product traceability. Its capability for remote identification, the various options of tags, readers, and antennas available make this technology a valuable tool in inventory management and logistics.

IV. PROPOSAL

This section describes the general process of handling a fruit or vegetable, such as an apple, in a cooperative, with the aim of providing a detailed explanation of the steps involved. It is important to note that, in the context of a cooperative, typically one or two varieties are managed per season. For example, apples may be harvested in summer and oranges in winter.

Each cooperative includes several farmers, each owning their own orchard. At the beginning of the harvest season, farmers must provide the cooperative with certificates related to their orchard, such as origin certificates, pesticide usage certificates, fertilization certificates, etc. The administrative part of the cooperative integrates this information into a database, generating a unique identification (ID) for each farmer. When a farmer delivers their harvest, they only need to enter their cooperative ID on the weighing machine's screen and unload their fruits from the truck onto the weighing area. The weighing machine stores the date, farmer's ID, weight contributed, fruit type, and assigns a batch ID that is the same for all apples coming from the same truck.

Once all the fruits or vegetables in the truck have been weighed, the "end of weighing" button is pressed. At this point, the weighing machine stores in the database all the weights contributed by the farmer with their corresponding batch ID. If QR codes are present on the fruits at that time, it would be possible to scan them to identify the farmer,

where the fruit or vegetable was grown, and the certifications it has.

After the weighing process is completed, the washing, disinfection, and drying process, known as "line entry," begins. During these operations, new data is generated by sensors and machines connected to the platform, such as water temperature, disinfectant usage, and drying temperature. This data is stored in the database and associated with the last stored batch ID. The code programmed in AWS receives a series of parameters measured by the machine's sensors and creates an operation with its corresponding operation ID. The only field to be completed is the batch ID. To determine which batch is being worked on, a query is made to the database to find the last batch that doesn't have a cleaning/disinfection/drying operation in the last hour, and its batch ID is copied.

Once the handling process is completed, the selection and classification phase take place. Some cooperatives perform manual selection with workers, while others use machines. In both cases, certain fruits or vegetables are removed from the process due to being in poor condition or other reasons, and the rest are divided based on their size (caliber).

In the case considered in this proposal, three processing lines for fruits with different calibers are considered: small, medium, and large. It is important to note that the caliber is categorical and can only take one of these three values. Depending on the cooperative's configuration or the type of fruit or vegetable being processed, these values may vary. Each line is directed to a laser engraving machine to apply QR codes on the skin of the fruits or vegetables, which requires a connection to the database.

This process is automated through AWS. The machine generates a new QR code containing a new ID and stores it in the database to later be engraved on the fruit. At the same time, the system assigns that QR code to the last created batch ID in the database and the corresponding sub-batch ID for the machine. The sub-batch ID indicates the caliber of the fruit and is previously configured by an operator at the start of the day or season. This configuration determines which size of fruits is assigned to each of the three production lines.

Once the fruits are labeled with the QR codes, they can be scanned to obtain information about their origin, the cultivating orchard, or the pesticides used.



FIGURE 4. APPLE WITH LASER ENGRAVED QR [8]

Next, a differentiation is made based on the marketing format of the fruits or vegetables. Those sold in bulk are placed in collapsible plastic crates and an RFID tag is added to facilitate their tracking in the logistics chain. On the other hand, those sold in individual packages are incorporated into the packaging, which is screen-printed with a printer using food-grade ink. These packages are then placed in similar boxes as mentioned earlier for easier transportation. Both the bulk and individually packaged fruits are placed on a pallet, which also has an RFID tag.

This entire process occurs simultaneously with the laser engraving of the fruits. While they are being engraved with the laser, they are placed into the individual packages, which are then placed in the boxes. The ID_Fruit, ID_Package, and ID_Box are assigned. In the case of bulk packaging, the fruits are directly placed into the boxes with the RFID, which is detected by the packaging machine, assigning them the ID_Fruit and ID_Box. In both cases, the ID_Box is assigned to the ID_Pallet so that during the journey from the cooperative to the point of sale, the entire pallet can be identified as soon as any of the RFID tags contained within it are detected. Thanks to the assigned IDs, the entire contents of the pallet can be quickly identified by detecting any RFID associated with it.

So far, the explanation has focused on the location of the pallet along the logistics chain but does not address the assignment of atmospheric values throughout that chain. To achieve this, an automated system must be implemented in AWS that periodically collects the atmospheric condition values from the different logistics centers through which the fruit passes. This system assigns each measurement an ID_Measurement and a measurement date, allowing the data on the physical position of the pallet to be cross-referenced with the measurements taken at that location. In this way, a complete and detailed record of the environmental conditions under which the apples have been transported along the logistics chain can be obtained.

Regarding the data from trucks and containers, the operation is slightly different because the data provider is the carrier who monitors the atmospheric conditions inside the container and provides them through an API that connects the Remote Container Management (RCM) [8] with the AWS platform.

V. PLATFORM

Each of the AWS services proposed for the development of the platform architecture is described and a sketch of it is presented.

AWS RDS: It is a fully managed relational database service that allows users to easily create, operate, and scale relational databases in the cloud. RDS supports a wide range of relational databases, including MySQL, PostgreSQL, Oracle, Microsoft SQL Server, and Amazon Aurora.

AWS Lambda: It is a serverless computing service that allows you to run code without provisioning or managing servers. It can be used to create highly scalable and available applications, automate tasks, process real-time data, or integrate with other AWS services.

AWS IoT Core: It is a service that allows you to connect IoT devices to the cloud and process data generated by them. It can be used to collect data from sensors, control devices, send commands to devices, analyze real-time data, etc.

AWS Cognito: It is a service that allows you to add authentication, authorization, and user management to web and mobile applications. It can be used to enable users to log in with their own user credentials, use social identity providers, manage users and groups, etc.

AWS Amplify: It is a cloud development service that simplifies the creation and deployment of web and mobile applications. Amplify offers a wide range of functionalities to accelerate application development, providing developers with the necessary tools to create scalable, secure, and high-performance applications.

AWS API Gateway: It is a service that allows you to easily and quickly create, publish, maintain, monitor, and protect your own APIs.

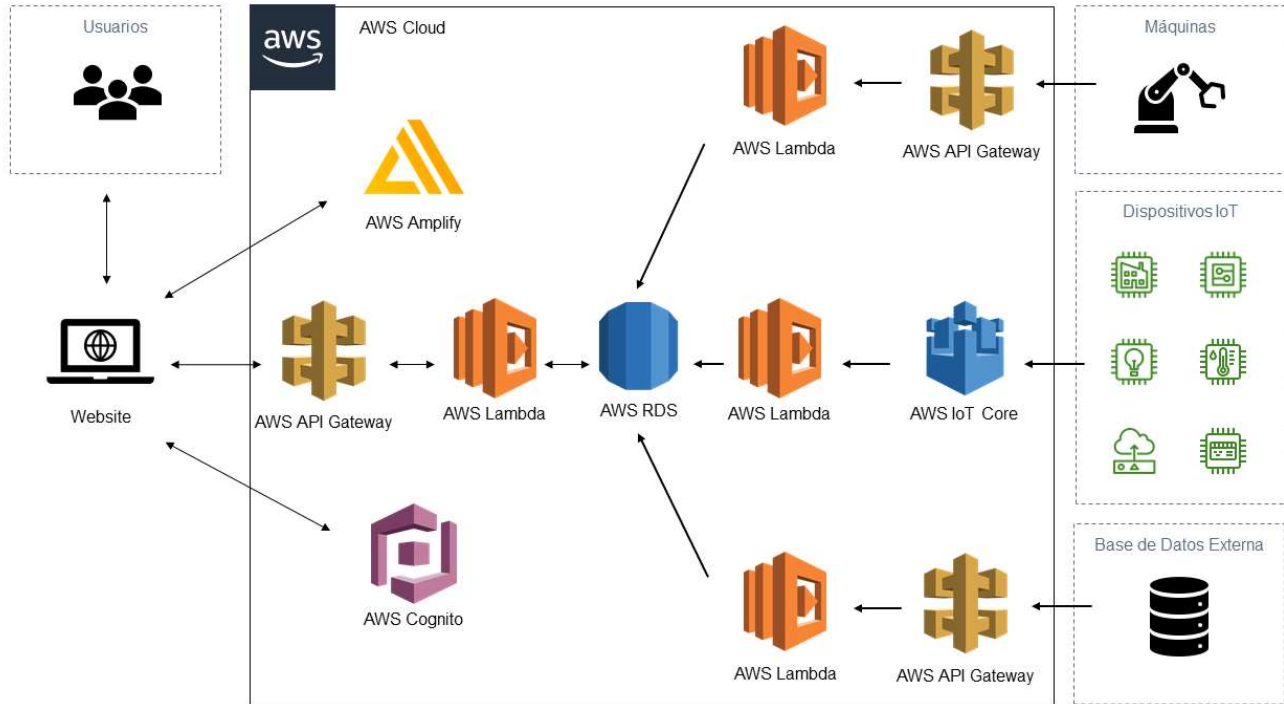


FIGURE 5. AWS ARCHITECTURE

From this database, four different modules branch out, each serving specific functions:

The first module is intended for end users, whether they are consumers or clients. These users can access the platform by scanning the QR code on the fruit. Upon doing so, they are redirected to a web page that utilizes AWS Cognito for user identification and AWS Amplify to host the web page. Additionally, the web page connects with the database through AWS API Gateway and AWS Lambda to request specific information about the scanned fruit.

The second module is responsible for the interaction between the machines and the database. For example, the machine that engraves QR codes on the fruits generates a new code and saves it in the database. To perform this task,

it utilizes AWS API Gateway and AWS Lambda as intermediaries between the laser machine and the database. The third module is responsible for receiving data from IoT devices used by cooperatives, logistics centers, suppliers, or supermarkets. This module uses AWS IoT Core and AWS Lambda to receive data from the devices, process it, and store it in the database.

Finally, the fourth module retrieves data from external databases, such as those used by carriers utilizing the Remote Container Management (RCM) software to collect and store data from their trucks or shipping containers. It uses AWS API Gateway and AWS Lambda as intermediaries between the external database and the platform's database. The platform's database makes use of Amazon RDS (Relational Database Service)[9], a SQL-based database.

VI. DATABASE

Below, a schema of the relational database developed in MySQL Workbench[9], a software for database creation, is presented. Each of the tables that comprise this database will be explained.

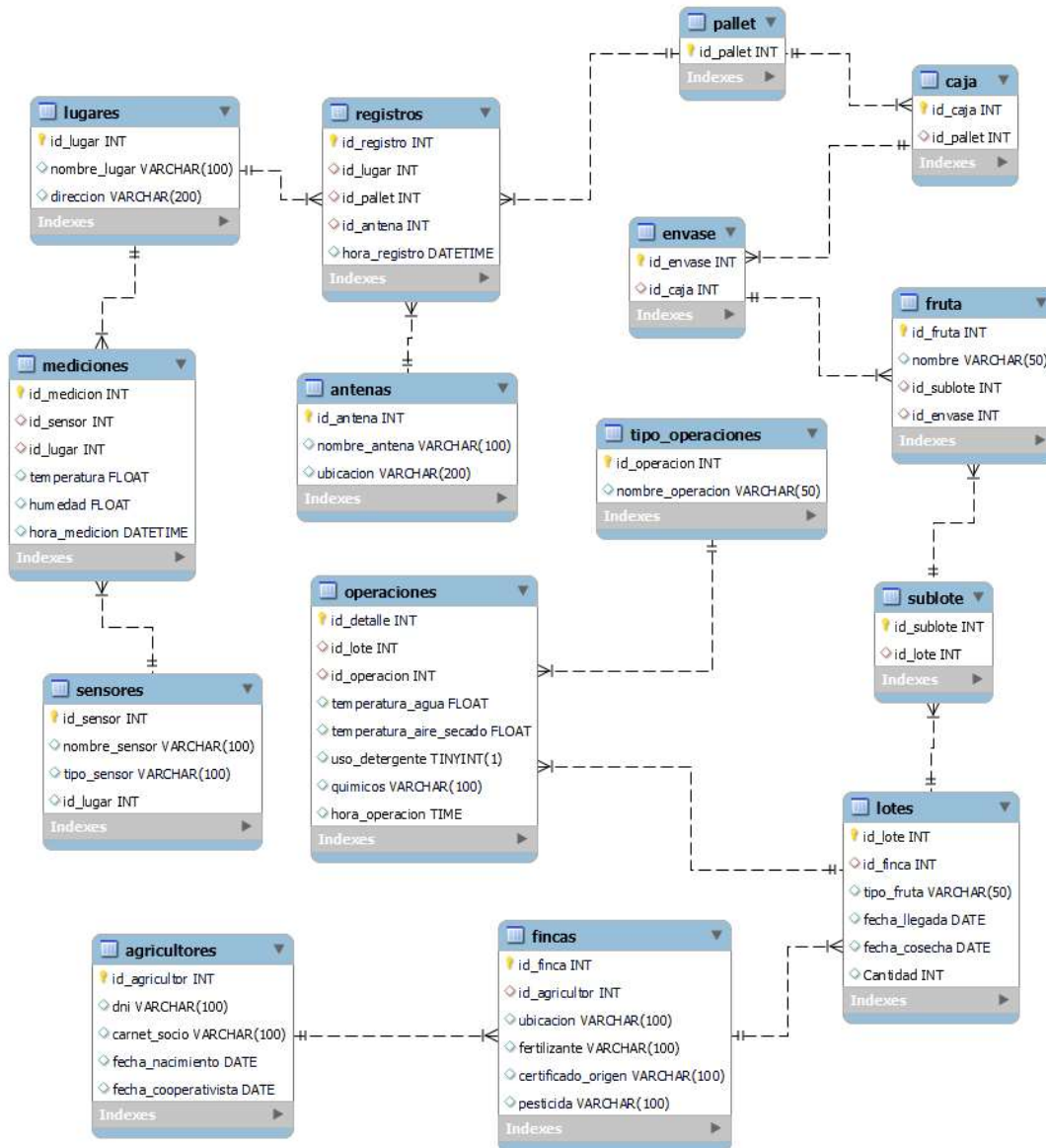


FIGURE 6. DATABASE

The "Agricultores" table stores information about the farmers participating in the system. Each farmer can have multiple farms, establishing a 1:N relationship between the "Agricultores" table and the "Fincas" table. The "id_agricultor" column is defined as the Primary Key (PK) in the table, which means it is a unique identifier for each farmer.

The "Fincas" table contains information about the farms associated with the farmers. Each farm is linked to a specific farmer through the "id_agricultor" column, which is set as a Foreign Key (FK) to establish the relationship

with the "Agricultores" table. The "id_finca" column is defined as the Primary Key (PK) in the table, making it a unique identifier for each farm.

The "Lotes" table contains records of the fruit batches. Multiple batches can belong to the same farm, establishing a N:1 relationship between the "Lotes" table and the "Fincas" table. The "id_lote" column is defined as the Primary Key (PK), making it a unique identifier for each batch. The "id_finca" column is set as a Foreign Key (FK) to establish the relationship with the "Fincas" table.

The "Operaciones" table stores records of the operations performed on the batches. Multiple operations can be associated with a single batch, establishing a N:1 relationship between the "Operaciones" table and the "Lotes" table. Additionally, multiple operations can have the same type of operation, establishing a N:1 relationship between the "Operaciones" table and the "Tipo_operaciones" table. The "id_detalle" column is defined as the Primary Key (PK), making it a unique identifier for each operation. The "id_lote" and "id_operacion" columns are set as foreign keys (FK) to establish the relationships with the "Lotes" and "Tipo_operaciones" tables, respectively.

The "Tipo_operaciones" table contains different types of operations. The "id_operacion" column is set as the Primary Key (PK) to uniquely identify each type of operation.

The "Sublote" table stores information about the sub-lots associated with the batches. Multiple sub-lots can belong to the same batch, establishing a N:1 relationship between the "Sublote" table and the "Lotes" table. The "id_sublote" column is set as the Primary Key (PK), making it a unique identifier for each sub-lot. The "id_lote" column is defined as a Foreign Key (FK) to establish the relationship with the "Lotes" table.

The "Fruta" table contains information about the fruits associated with the sub-lots and containers. Multiple fruits can belong to the same sub-lot, establishing a N:1 relationship between the "Fruta" table and the "Sublote" table. Additionally, multiple fruits can be in the same container, establishing a N:1 relationship between the "Fruta" table and the "Envase" table. The "id_fruta" column is set as the Primary Key (PK), making it a unique identifier for each fruit. The "id_sublote" and "id_envase" columns are defined as foreign keys (FK) to establish the relationships with the "Sublote" and "Envase" tables, respectively.

The "Envase" table stores information about the containers used in the boxes. Multiple containers can be in the same box, establishing a N:1 relationship between the "Envase" table and the "Caja" table. The "id_envase" column is defined as the Primary Key (PK), making it a unique identifier for each container. The "id_caja" column is set as a Foreign Key (FK) to establish the relationship with the "Caja" table.

The "Caja" table contains information about the boxes associated with the pallets. Multiple boxes can be associated with a single pallet, establishing a N:1 relationship between the "Caja" table and the "Pallet" table. The "id_caja" column is defined as the Primary Key (PK), making it a unique identifier for each box. The "id_pallet" column is set as a Foreign Key (FK) to establish the relationship with the "Pallet" table.

The "Pallet" table stores records of the pallets used in the operation. The "id_pallet" column is set as the Primary Key (PK) in the table to uniquely identify each pallet.

The "Registros" table stores records of events related to the locations, pallets, and antennas. Each record is associated with a location, pallet, and antenna, establishing a N:1 relationship between the "Registros" table and the "Lugares", "Pallet", and "Antenas" tables, respectively. The "id_registro" column is defined as the Primary Key (PK), while the "id_lugar", "id_pallet", and "id_antena" columns are set as foreign keys (FK) to establish the relationships with the "Lugares", "Pallet", and "Antenas" tables, respectively.

The "Antenas" table stores information about the antennas used in the records. The "id_antena" column is set as the Primary Key (PK) to uniquely identify each antenna.

The "Lugares" table contains information about the locations related to the records. The "id_lugar" column is set as the Primary Key (PK) to uniquely identify each location.

La tabla "Mediciones" sirve como un repositorio para almacenar las mediciones de sensores capturadas en varias ubicaciones. Cada entrada de medición en esta tabla está vinculada a un sensor y una ubicación específica, estableciendo una relación N:1 con las tablas "Sensores" y "Lugares", respectivamente. La columna "id_medicion" está designada como la Clave Primaria (PK), mientras que las columnas "id_sensor" e "id_lugar" se definen como claves foráneas (FK), estableciendo las relaciones con las tablas "Sensores" y "Lugares", respectivamente.

The "Sensores" table contains information about the sensors used in the measurements. The "id_sensor" column is set as the Primary Key (PK) to uniquely identify each sensor.

VII. QUERYS

In this section, several queries are performed on the database to demonstrate its functionality and usefulness in what could be a real-life scenario.

The first query shows the records associated with a specific pallet that contains the fruit with an ID. First, a subquery is executed to obtain the ID of the pallet to which the fruit belongs. Then, the "Registros", "Pallet", and "Lugares" tables are joined using the corresponding identifiers.

The result of the query includes all fields from the "Registros" table, along with the name of the place and its address. This provides detailed information about each record, such as the timestamp, the place's ID, and the location data of the place. By relating the records to the pallets and fruits, a comprehensive view of the places the

specific fruit has passed through is obtained, making it easier to track its journey in the supply chain.

```
SELECT R.*, L.nombre_lugar, L.direccion
FROM Registros R
JOIN Pallet P ON R.id_pallet = P.id_pallet
JOIN Lugares L ON R.id_lugar = L.id_lugar
WHERE P.id_pallet = (SELECT C.id_pallet
                    FROM Fruta F
                    JOIN Envase E ON
F.id_envase = E.id_envase
                    JOIN Caja C ON
E.id_caja = C.id_caja
                    WHERE F.id_fruta = 1);
```

	id_registro	id_lugar	id_pallet	id_antena	hora_registro	nombre_lugar	direccion
▶	29	1	1	3	2023-07-03 15:50:56	Lugar A	Dirección A
	30	1	1	2	2023-07-03 15:50:56	Lugar A	Dirección A
	50	1	1	3	2023-07-03 15:50:56	Lugar A	Dirección A
	92	1	1	3	2023-07-03 15:50:56	Lugar A	Dirección A
	94	1	1	3	2023-07-03 15:50:56	Lugar A	Dirección A
	97	1	1	1	2023-07-03 15:50:56	Lugar A	Dirección A
	1	2	1	1	2023-07-03 15:50:56	Lugar B	Dirección B
	42	2	1	3	2023-07-03 15:50:56	Lugar B	Dirección B
	44	2	1	3	2023-07-03 15:50:56	Lugar B	Dirección B
	46	2	1	2	2023-07-03 15:50:56	Lugar B	Dirección B
	47	2	1	2	2023-07-03 15:50:56	Lugar B	Dirección B
	73	2	1	3	2023-07-03 15:50:56	Lugar B	Dirección B
	6	3	1	2	2023-07-03 15:50:56	Lugar C	Dirección C
	9	3	1	3	2023-07-03 15:50:56	Lugar C	Dirección C
	12	3	1	1	2023-07-03 15:50:56	Lugar C	Dirección C
	19	3	1	2	2023-07-03 15:50:56	Lugar C	Dirección C
	21	3	1	3	2023-07-03 15:50:56	Lugar C	Dirección C
	24	3	1	2	2023-07-03 15:50:56	Lugar C	Dirección C
	28	3	1	1	2023-07-03 15:50:56	Lugar C	Dirección C
	41	3	1	1	2023-07-03 15:50:56	Lugar C	Dirección C
	49	3	1	2	2023-07-03 15:50:56	Lugar C	Dirección C
	55	3	1	1	2023-07-03 15:50:56	Lugar C	Dirección C
	61	3	1	2	2023-07-03 15:50:56	Lugar C	Dirección C
	62	3	1	1	2023-07-03 15:50:56	Lugar C	Dirección C
	88	3	1	2	2023-07-03 15:50:56	Lugar C	Dirección C
	93	3	1	2	2023-07-03 15:50:56	Lugar C	Dirección C

FIGURE 7. RESULT OF THE QUERY RECORDS

The second query provides information about the fruits that are present in a specific pallet, in this case, pallet number 2. The query joins multiple tables in the database to retrieve the details of each fruit. First, the fruits are related to their corresponding containers using a shared identifier. Then, the containers are related to the boxes through another shared identifier. Lastly, the boxes are related to the pallet using another shared identifier.

The result of the query displays the ID and name of the fruits, along with the container ID and the box ID in which they are located. This allows us to understand how the fruits are organized and stored in the distribution process, facilitating the tracking and control of the supply chain.

```
SELECT F.id_fruta, F.nombre, E.id_envase,
C.id_caja
FROM Fruta F
JOIN Envase E ON F.id_envase = E.id_envase
JOIN Caja C ON E.id_caja = C.id_caja
JOIN Pallet P ON C.id_pallet = P.id_pallet
WHERE P.id_pallet = 2;
```

	id_fruta	nombre	id_envase	id_caja
▶	100	Naranja	5	7
	149	Naranja	5	7
	101	Naranja	7	10
	184	Naranja	11	8
	53	Naranja	12	6
	125	Naranja	12	6
	153	Naranja	14	8
	65	Naranja	17	7
	83	Naranja	32	6
	173	Naranja	45	9
	68	Naranja	64	6
	85	Naranja	64	6
	87	Naranja	64	6
	106	Naranja	68	10
	98	Naranja	83	9
	51	Naranja	97	10
	111	Naranja	97	10
	167	Naranja	97	10

FIGURE 8. RESULT OF THE QUERY PALLETS

Finally, the third query retrieves detailed information about the fruits that have been cultivated by farmers with the identifiers 2 and 3. It displays data such as the fruit type, arrival date, location of the farm where the fruits were grown, and personal details of the farmers such as their identification number, membership card, etc.

The query joins multiple tables, such as the fruit table, sublots, lots, farms, and farmers, using connecting information between them. This allows us to obtain a comprehensive view of the fruits along with the related details about the farmers and the farms where they were cultivated.

```
SELECT F.*, A.id_agricultor, A.dni,
A.carnet_socio, FC.id_finca, FC.ubicacion,
FC.certificado_origen
FROM Fruta F
JOIN Sublote SL ON F.id_sublote =
SL.id_sublote
JOIN Lotes LT ON SL.id_lote = LT.id_lote
JOIN Fincas FC ON LT.id_finca = FC.id_finca
JOIN Agricultores A ON FC.id_agricultor =
A.id_agricultor
WHERE A.id_agricultor IN (2, 3);
```

	id_fruta	nombre	id_sublote	id_envase	id_agricultor	dni	carnet_socio	id_finca	ubicacion	certificado_origen
▶	1	Naranja	23	31	2	987654321	EPGH456	3	Ubicación 3	Certificado Origen 3
	4	Naranja	22	62	2	987654321	EPGH456	3	Ubicación 3	Certificado Origen 3
	6	Naranja	11	52	2	987654321	EPGH456	3	Ubicación 3	Certificado Origen 3
	7	Naranja	26	95	2	987654321	EPGH456	4	Ubicación 4	Certificado Origen 4
	8	Naranja	25	24	2	987654321	EPGH456	4	Ubicación 4	Certificado Origen 4
	9	Naranja	29	74	3	123456789A	CS001	5	Ubicación 5	Certificado Origen 5
	10	Naranja	19	52	3	123456789A	CS001	5	Ubicación 5	Certificado Origen 5
	12	Naranja	25	61	2	987654321	EPGH456	4	Ubicación 4	Certificado Origen 4
	15	Naranja	28	2	3	123456789A	CS001	5	Ubicación 5	Certificado Origen 5
	17	Naranja	21	96	3	123456789A	CS001	5	Ubicación 5	Certificado Origen 5
	18	Naranja	30	60	3	123456789A	CS001	5	Ubicación 5	Certificado Origen 5
	19	Naranja	28	79	3	123456789A	CS001	5	Ubicación 5	Certificado Origen 5
	21	Naranja	30	86	3	123456789A	CS001	5	Ubicación 5	Certificado Origen 5
	23	Naranja	25	81	2	987654321	EPGH456	4	Ubicación 4	Certificado Origen 4
	26	Naranja	28	43	3	123456789A	CS001	5	Ubicación 5	Certificado Origen 5
	27	Naranja	19	47	3	123456789A	CS001	5	Ubicación 5	Certificado Origen 5
	30	Naranja	10	28	2	987654321	EPGH456	3	Ubicación 3	Certificado Origen 3
	32	Naranja	27	16	2	987654321	EPGH456	4	Ubicación 4	Certificado Origen 4
	33	Naranja	11	85	2	987654321	EPGH456	3	Ubicación 3	Certificado Origen 3

FIGURE 9. RESULT OF THE QUERY FARMERS

VIII. CONCLUSION

In conclusion, the study of the horticultural traceability platform project reveals significant technical findings that support its implementation. The main conclusions are as follows:

Reduction of unnecessary packaging and waste reduction: The implementation of technologies such as laser-engraved QR codes enables the individual identification of products, which in turn reduces the need for additional packaging. This contributes to both the reduction of unnecessary packaging and the decrease in waste generated in the horticultural supply chain.

Promotion of food safety and transparency: The platform provides detailed information to consumers, such as product origin, cultivation methods, and harvest dates. This promotes food safety and transparency, giving consumers the confidence to know the origin and quality of the products they consume.

Scalability thanks to AWS: The platform benefits from the scalability and flexibility provided by Amazon Web Services (AWS). This means that the platform can efficiently grow and adapt as demand increases and expands to new markets.

Strategic opportunity in Spain: Starting the business in Spain, as a member country of the European Union, provides a relevant market and a solid foundation to validate the platform before expanding to other countries. The horticultural industry in Spain offers a conducive environment to test and refine the platform, and EU membership provides access to a broad and regulated market.

APPENDIX A

This SQL code consists of a series of instructions for creating tables in a database. These tables are designed to store information about fruits and their relationships with boxes, containers, pallets, etc.

```
- Crear tabla "Agricultores"
```

```
CREATE TABLE Agricultores (
    id_agricultor INT PRIMARY KEY,
    dni VARCHAR(100),
    carnet_socio VARCHAR(100),
    fecha_nacimiento DATE,
    fecha_cooperativista DATE
);
```

```
-- Crear tabla "Fincas"
```

```
CREATE TABLE Fincas (
    id_finca INT PRIMARY KEY,
    id_agricultor INT,
    ubicación VARCHAR(100),
    fertilizante VARCHAR(100),
    certificado_origen VARCHAR(100),
```

```
    pesticida VARCHAR(100),
    FOREIGN KEY (id_agricultor) REFERENCES
    Agricultores(id_agricultor)
);
```

```
-- Crear tabla "Pallet"
```

```
CREATE TABLE Pallet (id_pallet INT PRIMARY
KEY);
-- Crear tabla "Caja"
CREATE TABLE Caja (
    id_caja INT PRIMARY KEY,
    id_pallet INT,
    FOREIGN KEY (id_pallet) REFERENCES
    Pallet(id_pallet)
);
```

```
-- Crear tabla "Envase"
```

```
CREATE TABLE Envase (
    id_envase INT PRIMARY KEY,
    id_caja INT,
    FOREIGN KEY (id_caja) REFERENCES
    Caja(id_caja)
);
```

```
-- Crear tabla "Lotes"
```

```
CREATE TABLE Lotes (
    id_lote INT PRIMARY KEY,
    id_finca INT,
    tipo_fruta VARCHAR(50),
    fecha_llegada DATE,
    fecha_cosecha DATE,
    Cantidad INT,
    FOREIGN KEY (id_finca) REFERENCES
    Fincas(id_finca)
);
```

```
-- Crear tabla "Sublote"
```

```
CREATE TABLE Sublote (
    id_sublote INT PRIMARY KEY,
    id_lote INT,
    FOREIGN KEY (id_lote) REFERENCES
    Lotes(id_lote)
);
```

```
-- Crear tabla "Tipo_operaciones"
```

```
CREATE TABLE Tipo_operaciones (
    id_operacion INT PRIMARY KEY,
    nombre_operacion VARCHAR(50)
);
```

```
-- Crear tabla "Operaciones"
```

```
CREATE TABLE Operaciones (
    id_detalle INT PRIMARY KEY,
    id_lote INT,
    id_operacion INT,
    temperatura_agua FLOAT,
    temperatura_aire_secado FLOAT,
    uso_detergente BOOLEAN,
    quimicos VARCHAR(100),
```

```

hora_operacion TIME,
FOREIGN KEY (id_lote) REFERENCES
Lotes(id_lote),
FOREIGN KEY (id_operacion) REFERENCES
Tipo_operaciones(id_operacion)
);

-- Crear tabla "Fruta"

CREATE TABLE Fruta (
    id_fruta INT PRIMARY KEY,
    nombre VARCHAR(50),
    id_sublote INT,
    id_envase INT,
    FOREIGN KEY (id_sublote) REFERENCES
Sublote(id_sublote),
FOREIGN KEY (id_envase) REFERENCES
Envase(id_envase)
);

-- Crear tabla "Antenas"

CREATE TABLE Antenas (
    id_antena INT PRIMARY KEY,
    nombre_antena VARCHAR(100),
    ubicacion VARCHAR(200)
);

-- Crear tabla "Lugares"

CREATE TABLE Lugares (
    id_lugar INT PRIMARY KEY,
    nombre_lugar VARCHAR(100),
    direccion VARCHAR(200)
);

-- Crear tabla "Registros"

CREATE TABLE Registros (
    id_registro INT PRIMARY KEY,
    id_lugar INT,
    id_pallet INT,
    id_antena INT,
    hora_registro DATETIME,
    FOREIGN KEY (id_lugar) REFERENCES
Lugares(id_lugar),
FOREIGN KEY (id_pallet) REFERENCES
Pallet(id_pallet),
FOREIGN KEY (id_antena) REFERENCES
Antenas(id_antena)
);

-- Crear tabla "Sensores"

CREATE TABLE Sensores (
    id_sensor INT PRIMARY KEY,
    id_lugar INT,
    nombre_sensor VARCHAR(100),
    tipo_sensor VARCHAR(100),
    FOREIGN KEY (id_lugar) REFERENCES
Lugares(id_lugar)
);

-- Crear tabla "Mediciones"

```

```

CREATE TABLE Mediciones (
    id_medicion INT PRIMARY KEY,
    id_sensor INT,
    id_lugar INT,
    temperatura FLOAT,
    humedad FLOAT,
    hora_medicion DATETIME,
    FOREIGN KEY (id_sensor) REFERENCES
Sensores(id_sensor),
FOREIGN KEY (id_lugar) REFERENCES
Lugares(id_lugar)
);

```

APPENDIX B

This code generates random examples of SQL queries for three different tables: "fruits", "measurements", and "records". Each table has different columns, and random values are generated to populate those columns in each query.

In the first code block, examples are generated for the "fruits" table. This table represents information about each fruit. 200 random examples are generated with values such as the fruit name (in this case, "Orange"), the subplot, the container number, and the box number. These values are stored in a text string called "value_text" and then printed.

In the second code block, examples are generated for the "measurements" table. This table could represent sensor data measuring temperature and humidity at different locations. 100 random examples are generated with values such as sensor ID, location ID, temperature, humidity, and measurement time. The measurement dates are generated within a specific range. The examples are stored in a text string called "measurements_sql" and printed.

In the third code block, examples are generated for the "records" table. This table could represent records of products stored in a warehouse. 100 random examples are generated with values such as record ID, location ID, pallet ID (a type of platform for transporting goods), and antenna ID. A date and time of registration within a specific range are also generated. The examples are stored in a text string called "records_sql" and printed.

```

import random
import time
from datetime import datetime, timedelta

# Generar ejemplos aleatorios
texto_valores = ""
for i in range(1, 201):
    nombre = "Naranja"
    sublote = random.randint(1, 30)
    envase = random.randint(1, 100)
    envase = random.randint(1, 100)
    caja = random.randint(1, 40)

# Generar el contenido de la query

```

```

    texto_valores += f"({i}, '{nombre}',
{sublote}, {envase}, {caja}),\n"

# Imprimir la query SQL generada
print(texto_valores)

# Generar ejemplos aleatorios
mediciones_sql = ""

fecha_inicial = datetime(2023, 4, 1) #
Fecha inicial del rango
fecha_final = datetime(2023, 4, 30) # Fecha
final del rango

dias_totales = (fecha_final -
fecha_inicial).days + 1

for id_medicion in range(1, 101):
    id_sensor = random.randint(1, 10)
    id_lugar = random.randint(1, 5)
    temperatura = random.uniform(0, 100)
    humedad = random.uniform(0, 100)
    hora_medicion = fecha_inicial +
timedelta(days=random.randint(0,
dias_totales))

    # Generar el contenido de la query
    mediciones_sql += f"({id_medicion},
{id_sensor}, {id_lugar}, {temperatura},
{humedad}, '{hora_medicion}'),\n"

# Imprimir la query SQL generada
print(mediciones_sql)
# Generar registros aleatorios
registros_sql = ""

fecha_inicial = datetime(2023, 4, 1) #
Fecha inicial del rango
fecha_final = datetime(2023, 4, 30) # Fecha
final del rango

dias_totales = (fecha_final -
fecha_inicial).days + 1

for id_registro in range(1, 101):
    id_lugar = random.randint(1, 10)
    id_pallet = random.randint(1, 100)
    id_antena = random.randint(1, 5)
    hora_registro = fecha_inicial +
timedelta(days=random.randint(0,
dias_totales))

    # Generar el contenido de la query
    registros_sql += f"({id_registro},
{id_lugar}, {id_pallet}, {id_antena},
'{hora_registro}'),\n"

# Imprimir la query SQL generada
print(registros_sql)

```

REFERENCES

- [1] “Cuál es la diferencia entre verdura y hortaliza.” <https://www.bonviveur.es/preguntas/cual-es-la-diferencia-entre-verdura-y-hortaliza> (accessed Jul. 10, 2023).
- [2] “Cloud Computing Services - Amazon Web Services (AWS).” <https://aws.amazon.com/> (accessed Jul. 10, 2023).
- [3] “Natural Branding QR Code Video.” <https://viable.earth/plant-based-food/lasers-can-now-add-produce-branding-to-fruits-without-labels-or-packaging/> (accessed Jul. 14, 2023).
- [4] “Láser para grabar en la piel de las frutas la etiqueta y reducir los residuos del etiquetado convencional.” <https://efeagro.com/tecnologia-laser-etiquetas-fruta/> (accessed Jul. 09, 2023).
- [5] “Dos grandes innovaciones para un envase responsable con el medio ambiente para frutas y hortalizas exóticas.” <https://www.freshplaza.es/article/9187099/dos-grandes-innovaciones-para-un-envase-responsable-con-el-medio-ambiente-para-frutas-y-hortalizas-exoticas/> (accessed Jul. 09, 2023).
- [6] “¿Qué es el RFID? - Kiwanda.” <https://kiwandalabs.com/que-es-el-rfid/> (accessed Jul. 09, 2023).
- [7] Yago, “EPC Class1 Gen2,” 2011. <https://www.acore.es/blog/5/Escribir-y-leer-en-inventario-mas-de-96-bits-en-EPC-Class1-Gen2> (accessed Jul. 09, 2023).
- [8] “Lasers can now add produce branding to fruits without labels or packaging - Viable Earth.” <https://viable.earth/plant-based-food/lasers-can-now-add-produce-branding-to-fruits-without-labels-or-packaging/> (accessed Jul. 14, 2023).
- [9] “Remote Container Management (RCM) - YouTube.” <https://www.youtube.com/watch?v=-2oBE7QW7Qo> (accessed Jul. 09, 2023).



JAIME DÍEZ-CASCÓN MARTÍNEZ completed his Bachelor's degree in Industrial Engineering from Universidad Pontificia de Comillas in Madrid, Spain in 2021. He subsequently completed a double Master's degree in Industrial Engineering and Smart Industry at the same university in 2023. During his academic years, Jaime also served as a class representative and held positions in various university governing bodies.

In terms of work experience, Jaime interned as an Investment Analyst at Nobis Capital in Madrid. In this role, he analyzed companies in the Spanish and Portuguese markets with the objective of acquiring a €50 million enterprise value company in a fragmented and counter-cyclical sector. Prior to this, he worked as a Research Collaborator Intern at CIC LAB, where he designed a technical project using Machine Learning models to analyze data from a production line for a major automotive company. He also worked as a Summer Intern at the same company, developing a 3D visualization using Machine Learning techniques to optimize the logistics process for an international construction company. Currently, he works for Amazon in the strategic development of the air transportation network for Europe.

Beyond his professional experience, Jaime has demonstrated leadership in various organizations. He served as the Headquarters Coordinator at Bokatas, a non-governmental organization based in Madrid, where he managed a budget of €5,000 euros and led over 70 volunteers across different teams. Additionally, Jaime actively participated in CMU Loyola, where he held the position of Martial Arts Officer, teaching self-defense courses for women against gender-based violence and organizing Krav Maga classes.

This project has been recognized as the winner of the Innovation Challenge organized by AON, thanks to its innovative and pioneering approach within the horticultural sector. AON is a renowned global company specializing in professional services and risk management, with a broad presence in 120 countries.