

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE TELECOMUNICACIÓN  
UNIVERSIDAD POLITÉCNICA DE CARTAGENA



Proyecto Fin de Carrera

**Desarrollo de técnicas RFID para trazabilidad de productos alimentarios**



AUTOR: Juan Antonio Hernández Pedreño

DIRECTOR: Alejandro Álvarez Melcón

DIRECTOR: Fernando D. Quesada Pereira

COORDIRECTOR: Alfredo Parreño Marchante, EMPRESA: SIDCOM

Cartagena, Marzo 2013

## Índice:

1. Introducción.
  - 1.1. Tag.
  - 1.2. Lector.
  - 1.3. Middleware.
  - 1.4. Trazabilidad.
  - 1.5. Proyecto europeo FARM2FORK.
  
2. Revisión del estándar EPC.
  - 2.1. Introducción.
  - 2.2. Eventos y composición.
  - 2.3. Vocabulary.
  - 2.4. Tipos.
    - 2.4.1. Object Event.
    - 2.4.2. Aggregation Event.
    - 2.4.3. Quantity Event.
    - 2.4.4. Transaction Event.
  
3. Descripción del proceso de producción de la granja.
  - 3.1. Engorde.
  - 3.2. Comercial.
  - 3.3. Pesca.
  - 3.4. Envío.
  - 3.5. Recepción.
  
4. Diseño del sistema de trazabilidad.
  - 4.1. Engorde.
  - 4.2. Comercial.
  - 4.3. Pesca.
  - 4.4. Empaquetado.
  - 4.5. Envío.
  - 4.6. Recepción.
  - 4.7. Desempaquetado.

5. Descripción de los equipos empleados y software desarrollado.
  - 5.1. Hardware.
    - 5.1.1. Handheld.
    - 5.1.2. Tag.
  - 5.2. Software.
    - 5.2.1. Engorde.
    - 5.2.2. Comercial.
    - 5.2.3. Pesca.
    - 5.2.4. Empaquetado.
    - 5.2.5. Envío.
    - 5.2.6. Recepción.
    - 5.2.7. Desempaquetado.
  
6. Pruebas y resultados obtenidos.
  - 6.1. Prueba de laboratorio.
  - 6.2. Primera prueba de campo.
  - 6.3. Segunda prueba de campo.
  
7. Conclusiones.
  
8. Bibliografía.

## Introducción.

La palabra RFID es un acrónimo inglés de *Radio Frequency Identification* (Identificación por radiofrecuencia) e ilustra de una manera sencilla su cometido: ser capaz de reconocer un objeto inequívocamente, que previamente ha sido catalogado, y procesar esa información para un fin determinado (control de inventariado, asistencia, pagos electrónicos,...) [7].

El sistema de forma básica puede ser descompuesto en tres partes:

- El *tag*, un chip semiconductor con una antena integrada y, según si es pasivo o activo, una batería.
- El lector, que consta de una antena, módulos RF y un mínimo de componentes para procesar la información que recibe.
- El middleware, encargado en muchas ocasiones de controlar el lector y que opera la base de datos del sistema, tiene entre sus cometidos el de procesar la información que llega desde los lectores de acuerdo a la función que se persiga en el sistema.

La Ilustración 1 ayuda a ver de una manera sencilla y gráfica lo expuesto en el párrafo anterior.



Ilustración 1: Sistema de comunicaciones RFID básico

Una descripción sencilla del sistema en operación sería:

1. Se asigna un identificador al tag de acuerdo a las especificaciones del proyecto en el que esté enmarcado, fijándolo en el objeto que se pretenda posteriormente identificar. Se debe tener en cuenta el posicionamiento de éste en el objeto así como la composición, lugar donde estará el objeto, etc. Según la complejidad de éstos se puede almacenar gran cantidad de parámetros ya que pueden ir acompañados de sensores adicionales que monitoricen parámetros del entorno que sean necesarios conocer (temperatura, humedad, localización,...).
2. Se sitúa el lector en el emplazamiento que se deseé controlar, de manera que cuando el objeto pase por esa zona, se capture su paso.

3. Una vez es registrada la lectura del objeto y conformado el fichero de datos requeridos para el proceso, se envía a la estación base, quien procesará esta información y la almacenará según las necesidades.

## 1.1 Tag.

Su cometido fundamental es el de almacenar información que identifique al objeto al que esté adherido, además de otra que pueda ser crítica para el control que se pretenda hacer, un ejemplo de esto es la cantidad de trasposos de una jaula a otra que se hayan dado, etcétera. Debe ser capaz de transmitir estos datos al lector cuando éste se los requiera [7].

Su composición más elemental consiste en un chip electrónico junto con una antena transmisora y todo ello encapsulado en un objeto con forma de pegatina (también adopta otras formas según sea el dúo superficie-entorno en el que se vaya a situar). Como se aprecia en la Ilustración 2, se muestra un ejemplo de tag RFID, distinguiéndose los componentes anteriormente citados (chip electrónico y antena RF).

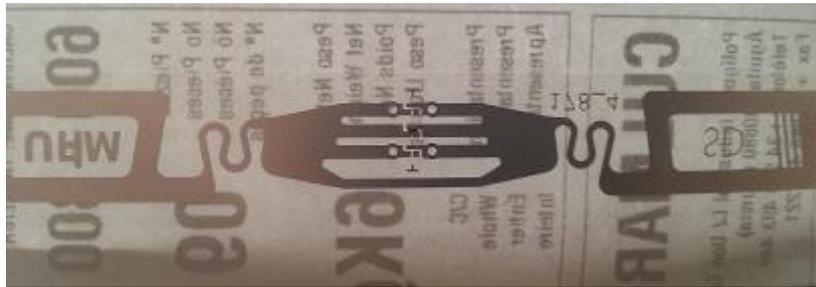


Ilustración 2: Ejemplo de tag pasivo y de tipo adhesivo

Según procedan para el envío de la respuesta al lector, se pueden clasificar en tres grandes familias:

- Pasivos, que emplean la energía que proviene de la interrogación del lector y del que físicamente podemos ver un ejemplo en la *Ilustración 2*.
- Activos, que cuentan con una batería en su estructura que provee de la energía necesaria para alimentar el chip y de esta manera mejorar el alcance con el que se puede interactuar con él. Son generalmente más complejos que los pasivos y su batería suele durar entre dos y siete años.
- Semipasivos, que se encuentran a mitad de camino entre los anteriores, teniendo una batería que en unos casos se emplea para alimentar el chip y en otras para mejorar la respuesta de lectura, con lo que aumenta el rango de uso.

Además de lo antes mencionado, debe considerarse el lugar en el que será fijado el tag, de manera que los sistemas de lectura sean capaces de interactuar con él.

No sólo la posición física respecto al elemento, sino también la composición de éste (madera, cartón, plástico,...) y de las condiciones a las que estará sometido (humedad, frío, calor,...), pues de ello depende que la comunicación entre lector y chip se lleve a cabo satisfactoriamente.

## 1.2 Lector.

Encargados de la identificación de los tags que están dentro de su cobertura, están compuestos de diferentes partes:

- Sistema de lectura, compuesto por una o varias antenas y lector.
- Módulo electrónico de RF.
- Módulo de control.

Tal y como se aprecia en la Ilustración 3, se muestra una antena empleada para la toma de datos y su correspondiente lector (en este caso de la marca Impinj).



Ilustración 3: Ejemplo de sistema lector con su antena (a la izquierda) y lector (a la derecha)

Esta disposición está sujeta a cambios dependiendo del entorno en el que se maneje, ya que en entornos móviles será preferible, por ejemplo, el empleo de un aparato a baterías y deberá contener algún sistema de comunicaciones sin cables para transmitir todo lo leído (tecnología WiFi, GSM,...) o poder conectarse físicamente al servidor que en el que esté el middleware instalado, una vez finalizadas todas las lecturas.

### 1.3 Middleware.

Es el cerebro del sistema RFID, quién almacenará todos los datos recibidos desde los lectores (pueden existir una gran cantidad de ellos según la complejidad y requerimientos del proyecto) y darle cohesión a la información obtenida (controlar pedidos, inventarios, asistencia,...). Suele ser un servidor o estación de trabajo en el que se aloja una base de datos y en el que se encuentra el *middleware* o sistema que coordina los sub-sistemas RFID con el resto de actividades que se lleven a cabo en la empresa, como pudiera ser el control de la cadena de suministros o la planificación de recursos en una determinada empresa.

De una manera más concisa, es posible definir las cuatro funciones más importantes del *middleware* [7]:

- Toma de datos: su cometido es el de filtrar toda la información en bruto que se pueda generar en los lectores, de manera que sólo se almacena aquella que pueda ser útil para los sistemas de producción de la empresa.
- Direccionado de datos: decide el destino de los datos que va tomando según qué sistema pueda necesitarlo.
- Manejo de procesos: en consecuencia de las anteriores capacidades, otra de sus características es la de poder alertar de eventos concernientes al proceso productivo (en algún punto de la cadena de producción un cierto lote), de logística (ya terminado un pedido), etc.
- Manejo de equipamiento: al igual que sucede con los datos, también se encarga de revisar y coordinar todos los lectores que están conectados a él.

Siguiendo la Ilustración 4, se distinguen los dos sentidos que puede seguir el flujo de información:

- El ascendente, en el que primero se toma una lectura de los datos en bruto por parte del sistema de lectura y que se envía al *middleware*, encargado de interpretar esa información y generar los avisos pertinentes para los programas y procesos propios de la empresa (planificador de recursos, almacenamiento de datos,...).
- El descendente, que sucede en sentido contrario partiendo desde los programas propios de la empresa que aportan unas reglas de negocio (un ejemplo podría ser una orden de compra compuesta por unos productos fijos) recogidas por el *middleware* que tiene el cometido de interpretarlas y cumplirlas, trasladando las órdenes necesarias a los elementos de entrada/salida (comprobar que todos los productos de un pedido están, escribir datos en un tag,...).

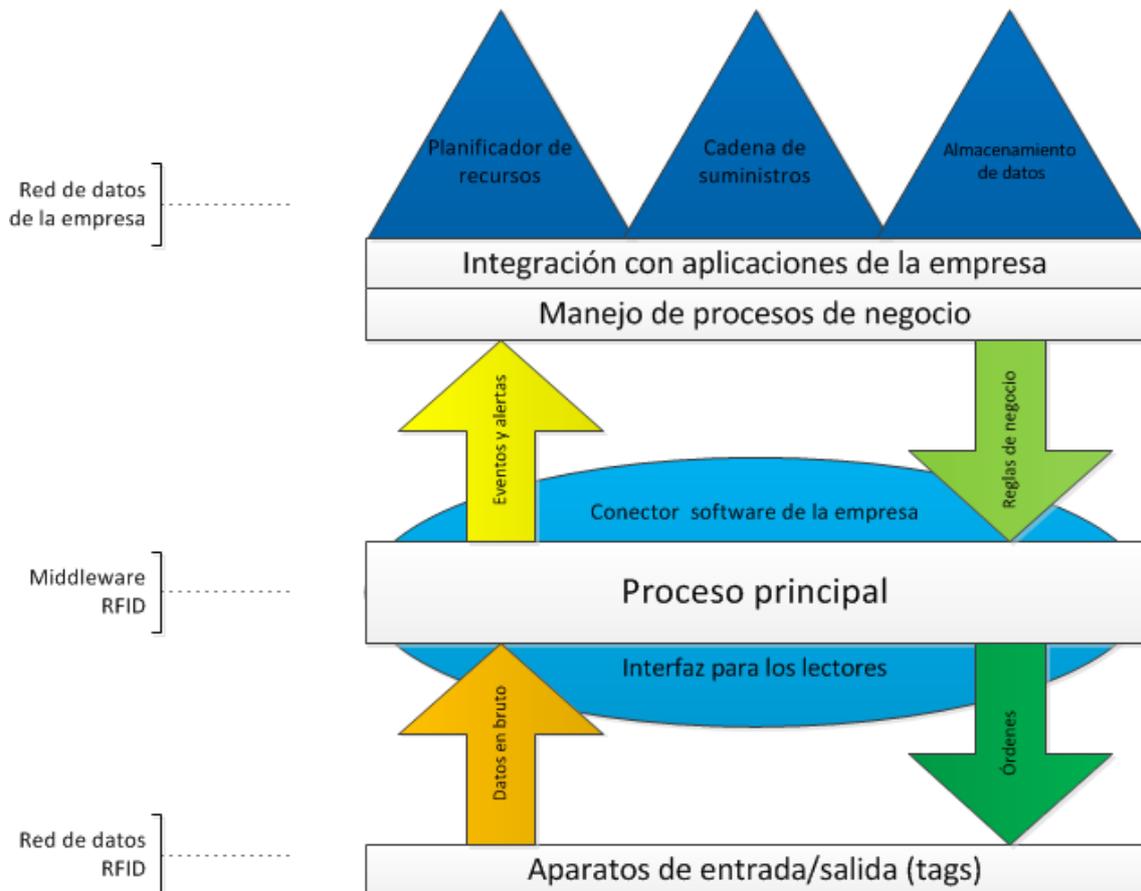


Ilustración 4: Middleware dentro de un sistema empresarial (traducción del original de Venture Development Corporation).

## 1.4 Trazabilidad.

Los sistemas, como han sido descritos previamente, se coordinan de manera que pueden producir tareas de más alto nivel dentro de una empresa u organización, desde controlar un proceso productivo y sus alertas en la producción, hasta la localización de lotes de productos ya terminados y listos para su posterior envío.

Uno de estos objetivos de alto nivel es la trazabilidad o conocer exactamente unos datos concretos sobre un elemento desde su origen, hasta que se transforma en un producto terminado y entregado. Un ejemplo sobre esta posibilidad se da en las industrias del sector de la alimentación, donde la trazabilidad es una obligación marcada por la ley.



Ilustración 5: Diagrama básico del pescado hasta un punto de venta.

En la *Ilustración 5* se puede observar una transición sencilla desde un pescado sin procesar, su paso por una factoría, que bien podría transformarlo en pescado en conserva, hasta un establecimiento de venta en el que el consumidor final podría adquirirlo. El objetivo de la trazabilidad es que el usuario final (desde empresas a particulares) sea capaz de obtener datos relevantes sobre todas las etapas por las que ha pasado ese producto:

- Día en que fue recibido en el establecimiento.
- Ruta que siguió.
- Temperatura del transporte.
- Día en que fue envasado.
- Etc.

De esta manera, se es capaz de trazar toda la vida del producto (*tracking*) desde el inicio del mismo, hasta que se compra, pudiendo ser el abanico de opciones tan amplio como la complejidad de los sistemas y datos que se deseen tener.

Los beneficios que aporta esta nueva herramienta redundan no sólo en el consumidor final que puede obtener de primera mano datos fiables y fehacientes sobre el producto que ha adquirido, sino también cara a las empresas que pueden optimizar sus procesos de producción con los datos que van recolectando o si existen fallos de seguridad, determinar qué productos están afectados.

## 1.5 Proyecto europeo FARM2FORK.

A la luz de las ventajas que podría traer la trazabilidad a las Pymes de la industria alimentaria en Europa, se impulsa este proyecto que pretende introducir los sistemas RFID para “*identificar y realizar la trazabilidad de la información sobre los alimentos que necesita la sociedad*” (Web oficial del proyecto - <http://www.rfid-f2f.eu/about.asp>).

El proyecto en su conjunto fue lanzado en Bruselas por la Universidad de Wolverhampton, coordinadora, y diez socios más que representan a seis Estados de la Unión Europea. Su propósito es la introducción de la tecnología RFID desde el inicio hasta el final de la cadena de producción y venta, de manera que toda la información que se genere desde el productor (*farm – granja*) hasta el consumidor (*fork – tenedor*) pueda ser consultada. Se autenticarían empresas productoras, procesadoras, de logística, vendedores para conseguir una solución estandarizada para la trazabilidad y autenticación de productos alimentarios.

Como parte de este proyecto, el cometido asumido por la UPCT es el de elaborar un piloto en una empresa, concretamente CULMAREX, en el que se estudiará su proceso de producción para conseguir sacar el máximo rendimiento de la tecnología a prueba sin que suponga una ruptura con los sistemas ya instalados.

A lo largo de los siguientes capítulos se proporcionará una visión global del proyecto, hasta centrarlo en la parte que sucede fuera de la planta de procesado de la empresa y que es la parte central de este Proyecto Fin de Carrera.

## Revisión del estándar EPC.

Esta sección tiene como objeto repasar las partes del estándar que son de utilidad para la comprensión del proyecto, de manera que se tenga un contexto con el que comprender mejor la parte técnica del mismo.

### 2.1 Introducción.

El estándar EPC fue desarrollado por EPCglobal en 2003, una *joint venture* entre U.C.C. (*Uniform Code Council*) y EAN International con el apoyo de la industria del RFID para el implante y mejora del estándar. El inicio de esta andadura se sitúa en el M.I.T., que inicia una investigación bajo el nombre *Auto-ID Center* en 1999, a la que posteriormente se añadirían más de cien compañías y cuatro universidades [7]:

- Cambridge (Reino Unido).
- Adelaida (Australia).
- Keio (Japón).
- St. Gallens (Suiza).

Su objetivo es la creación de una red de las cosas (*The internet of things*), es decir, una infraestructura que permita la identificación unívoca de objetos, que será conocida como E.P.C.N. (*Electronic Product Code Network*). Como parte de esta meta, surgen otros desarrollos que deben llevarse a cabo:

- E.P.C. (*Electronic Product Code*), para la designación de objetos.
- Especificaciones para tags y equipos de lectura que mejoren su precio.
- O.N.S. (*Object Naming Service*).
- P.M.L. (*Product Markup Language*).
- Software Savant, que ejerza de *middleware*.

### 2.2 Eventos y composición.

EPICIS (Servicio de Información EPC) tiene como cometido aportar una interfaz para el almacenamiento y consulta de datos EPC siguiendo unos mecanismos de seguridad y funciones de acceso que serán fijados con posterioridad, ya que el estándar de la interfaz sólo define la manera de compartir datos EPC entre la aplicación que los captura y aquella que los requiere.

De manera genérica, EPCIS divide la información en dos grupos [4]:

- Eventos, que son generados por el proceso productivo.
- *Master data*, datos que dan contexto (información añadida) para la interpretación de los eventos.

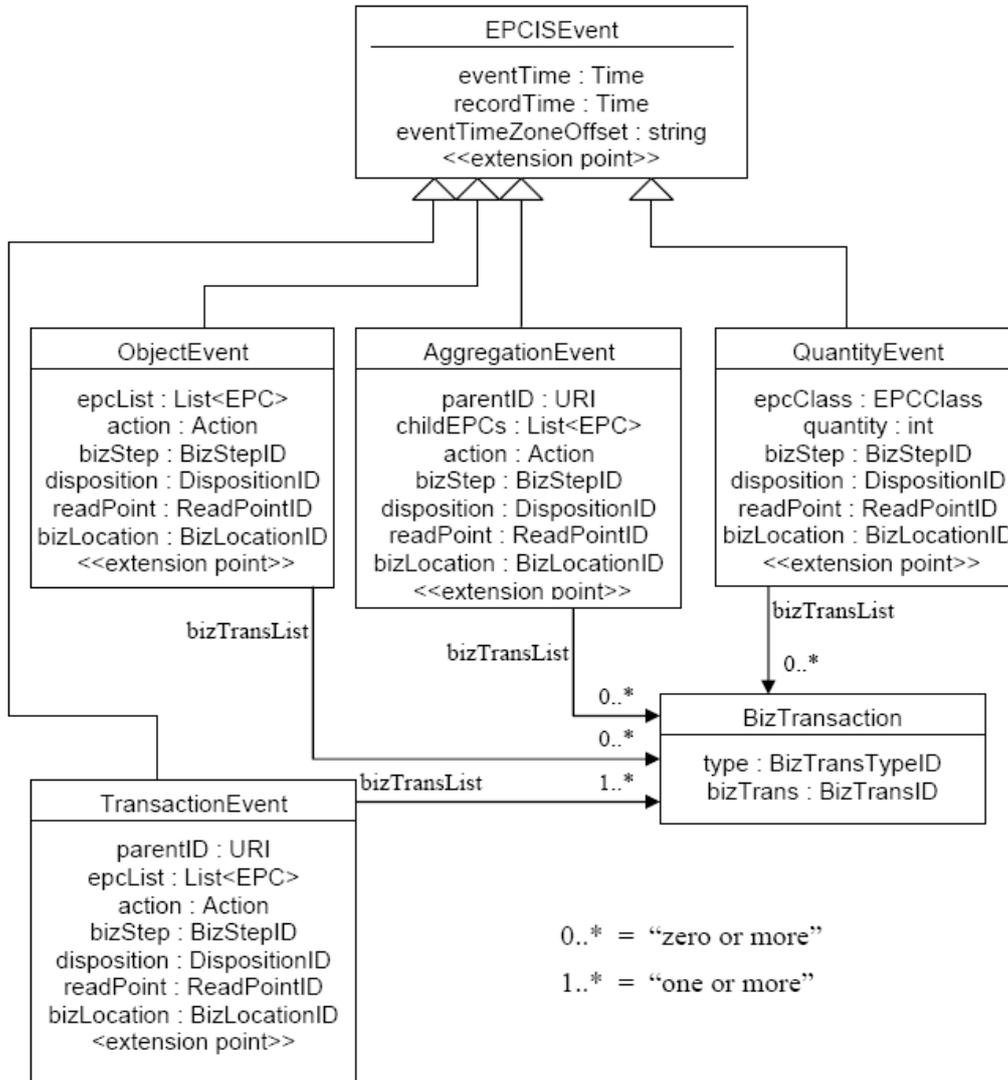


Ilustración 6: Diagrama en lenguaje unificado de modelado (UML por sus siglas en inglés) que recoge el tipo de eventos.

Como se observa en la *Ilustración 6*, existen cuatro tipos de eventos (sin contar el evento padre *EPCISEvent*) que se encargan de responder a las cuatro dimensiones importantes de un proceso, y que son:

- Objetos o entidades que provocan el evento (¿qué?).
- Fecha y hora del mismo (¿cuándo?).
- Lugar en el que se produce (¿dónde?).
- Contexto del negocio o proceso (¿por qué?).

## 2.3 Vocabulary.

Adicionalmente, se recoge en cada uno de ellos otra información referente a otros campos para dar una imagen más completa a cerca del evento registrado [3]:

- **eventTime:** Fecha y hora en el que ocurre dicho evento.
- **recordTime:** (Opcional) Fecha y hora en el que es almacenado el evento por un repositorio EPCIS.
- **eventTimeZoneOffset:** La diferencia horaria de donde se captura el evento respecto de la hora U.T.C.
- **readPoint:** Identidad del lector que captura el evento y que tiene una fuerte asociación de lugar. Responde al concepto “como-donde fue detectado”.
- **bizLocation:** Lugar donde se encuentra un objeto cuando se produce el evento y que no variará hasta que no se produzca la siguiente captura o evento. De manera simple, indica “a donde sigue el objeto al producirse un evento EPCIS”.
- **bizStep:** Aporta información de la situación en el proceso productivo, es decir, indica que acción está siendo llevada a cabo durante la captura del evento.
- **disposition:** Estado en el que queda el objeto tras el evento y en el que permanecerá hasta que se vuelva a registrar el siguiente evento.
- **bizTransaction:** (Opcional) Identifica tanto el tipo de transacción de ese evento como su identificación unívoca. De esta manera es posible comprobar si un determinado evento está asociado a una determinada transacción (por ejemplo completar una orden de compra, etc.).

## 2.4 Tipos.

### 2.4.1 Object Event.

Captura información sobre un evento que referencia a uno o varios objetos identificados por EPCs, quienes no tienen otra relación más que estar en el mismo evento [3].

- Campos de acción:
  - **ADD:** El EPC(s) ha sido dado y asociado al objeto(s) por primera vez.
  - **OBSERVE:** EPC(s) que han sido detectados en el evento, pero que ya existían de antes.
  - **DELETE:** EPC(s) se elimina y no podrá volver a ser observado.

- Campos: Se recogen los campos tipo para un evento de tipo Object, siendo los campos resaltados en color verde aquellos que no dependen del tipo de evento.

eventTime
recordTime
eventTimeZoneOffset
epcList
action
bizStep
disposition
readPoint
bizLocation
bizTransactionList

Tabla 1: Datos recogidos para un Object event.

### 2.4.2 Aggregation Event.

Describe un evento en el que se produce una unión o agregación de un objeto a otro (generalmente más grande). Es importante resaltar que la relación física que los una debe ser fuerte, estando en el mismo lugar y al mismo tiempo hasta que se separen, de no cumplirse esta condición, conviene emplear *TransactionEvent*. Un ejemplo de esta unión física es la unión de varias cajas para conformar un pallet o cuando son almacenadas en un camión para su posterior transporte [3].

- Campos de acción:
  - **ADD:** Los EPC(s) hijos se unen al padre tanto si es la primera unión como si son nuevos hijos los añadidos.
  - **OBSERVE:** Se emplea en dos casos, bien cuando no están presentes todos los hijos o cuando no se conoce el padre.
  - **DELETE:** Se pone fin a la unión entre los objetos, pudiendo escoger uno, varios o, si se deja vacío el campo, todos.
- Campos: Se recogen los campos tipo para un evento de tipo Aggregation, siendo los campos resaltados en color verde aquellos que no dependen del tipo de evento.

eventTime
recordTime
eventTimeZoneOffset
parentID
childEPCs
action
bizStep
disposition
readPoint
bizLocation
bizTransactionList

Tabla 2: Datos recogidos para un Aggregation event.

### 2.4.3 Quantity Event.

Captura un evento en el que se da información sobre la cantidad o nivel de una clase objeto o producto, como podría ser, por ejemplo, llevar la cantidad de unidades que se tienen en inventario de un cierto ítem [3].

- Campos: Se recogen los campos tipo para un evento de tipo Quantity, siendo los campos resaltados en color verde aquellos que no dependen del tipo de evento.

eventTime
recordTime
eventTimeZoneOffset
epcClass
quantity
bizStep
disposition
readPoint
bizLocation
bizTransactionList

Tabla 3: Datos recogidos para un Quantity event.

### 2.4.4 Transaction Event.

Se usa para representar la unión o separación de transacciones a unos EPCs específicos, de una manera directa, al contrario del campo bizTransactionList en el que se da información extra a otros eventos. Al contrario que sucede con *AggregationEvent*, no existe ninguna relación física o padre-hijo entre ambos. Un ejemplo podría ser una orden de compra a la que se le asocian las cajas que conforman el pedido [3].

- Campos de acción:
  - **ADD:** Los EPCs se unen a una o varias transacciones tanto si son creados por primera vez como si ya existieran y se unen en este momento a la(s) transacción(es).
  - **OBSERVE:** Se confirma que se mantienen dentro de la transacción aquellos EPCs que se detecten.
  - **DELETE:** Ruptura de los EPCs con la transacción, en caso de que estuviera vacía la lista, todos los EPCs quedarían fuera de la transacción.
  
- Campos: Se recogen los campos tipo para un evento de tipo Transaction, siendo los campos resaltados en color verde aquellos que no dependen del tipo de evento.

eventTime
recordTime
eventTimeZoneOffset
bizTransactionList
parentID
epcList
action
bizStep
disposition
readPoint
bizLocation

Tabla 4: Datos recogidos para un Transaction event.

## Descripción del proceso de producción de la granja.

El proceso de producción que presenta la empresa se puede dividir en dos fases bien diferenciadas [2]:

- Fase de cría y recolección, en la que los peces siguen su desarrollo y crecimiento hasta que son pescados y llevados a la planta de procesado.
- Fase de procesado y distribución, donde se clasifican los peces según su clase y se empaquetan en cajas para abastecer a los clientes según las peticiones de los mismos.

En este punto se desarrolla en profundidad la primera de las fases, de manera que contribuya a la comprensión del conjunto del proyecto.

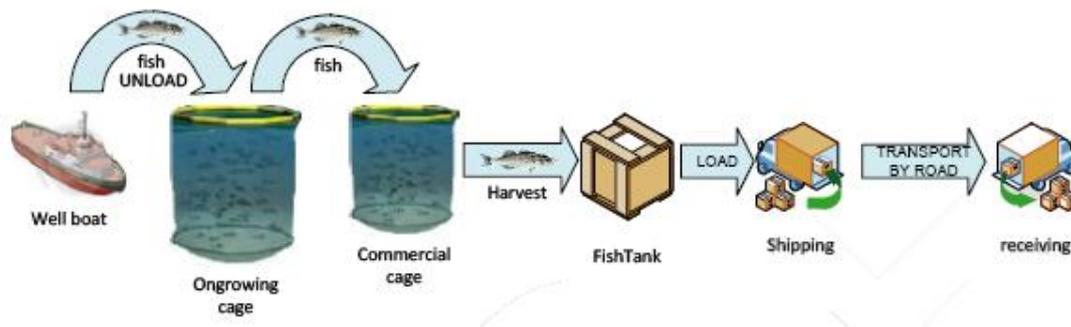


Ilustración 7: Diagrama de los pasos durante la cría en la granja.

Tal y como se refleja en la *Ilustración 7*, son cinco los estados por los que pasa el pez desde su recepción en la granja, hasta que comienza su procesado en planta, siendo cada uno de estos de diferentes duraciones de tiempo. Voy a detallar cada uno de ellos a continuación.

### 3.1 Engorde.



Ilustración 8: Desembarco de alevines en las jaulas de engorde.

Como se observa en la *Ilustración 8*, los peces no son criados en la granja desde su nacimiento, sino que son traídos desde otra distinta de la compañía y depositados a su llegada en jaulas de cría o engorde según el tipo de pez: dorada o lubina.



Ilustración 9: Jaula tipo.

Tal y como se aprecia en la *Ilustración 9*, estas jaulas son de un tamaño considerable, y albergarán a los alevines hasta que se produzca el traspaso a la siguiente etapa.

No se producen rellenados de las jaulas hasta que el lote que se depositó en ella haya salido completamente de la misma, con lo que todos los peces que encontremos dentro serán siempre del mismo lote, con las ventajas respecto a trazabilidad que esto conlleva.

### 3.2 Comercial.

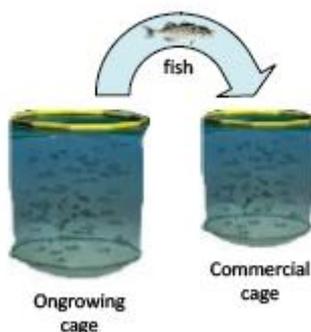


Ilustración 10: Movimiento de un subconjunto de peces a la jaula comercial.

Esta fase comienza con un traspaso de peces desde la jaula de engorde hasta la jaula comercial como se detalla en la *Ilustración 10*, atendiendo a pedidos que requiera la empresa cubrir. Los peces tienen ya un tamaño y peso que cumple con los estándares que se requiere y se los mantiene en esta jaula aislados del resto de los peces de la jaula de engorde. Los dos días anteriores a su pesca no se los alimentará, de manera que los intestinos estén limpios y aumente el tiempo de caducidad.

El vaciado de esta jaula no se realiza de una vez, estando sujeto a los pedidos y previsiones de venta que tenga la empresa y, al igual que sucedía con las jaulas de engorde, no serán rellenadas hasta que estén completamente vacías.

### 3.3 Pesca.

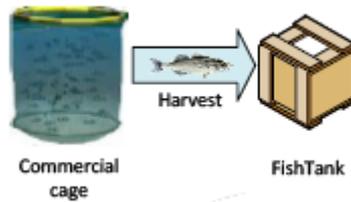


Ilustración 11: Pesca y empaquetado en cajas.

Empleando la *Ilustración 11*, en este punto se cubre la captura y empaquetado en cajas de los peces. Esta fase tiene lugar durante la noche y siempre de manera que no genere estrés en los peces o dañe su anatomía. Una vez son capturados, se confinan en tanques con agua del mar y abundante hielo para realizar su transporte a la planta de procesado. La cantidad y tipo de pez que se captura viene dado por los pedidos que tenga que atender la compañía, nunca capturando mayor cantidad y llevando un control riguroso de que se cumplan las previsiones de venta.

En la *Ilustración 12*, se puede ver como los distintos tanques de almacenamiento se encuentran llenos de hielo y organizados sobre la cubierta del barco, para su traslado a puerto.



Ilustración 12: Peces dentro de un tanque tras la pesca.

### 3.4 Envío.

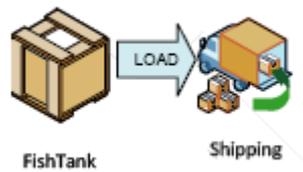


Ilustración 13: Transporte de los tanques.

Tras los pasos anteriores que suceden en el barco y como la *Ilustración 13* ayuda a indicar, se cubre la carga y envío de los tanques. Los tanques, previamente rellenos durante la pesca, son cargados en los camiones refrigerados que los transportarán a la planta de procesado. No son empaquetados siguiendo órdenes comerciales o de lote.

### 3.5 Recepción.



Ilustración 14: Recepción de los tanques.

Tras el transporte por carretera desde el muelle, los camiones llegan al área de descarga de los tanques, una zona en la que se depositarán los tanques apilados siguiendo la disposición que aparece en la *Ilustración 15*. Posteriormente son volcados en una máquina que dará comienzo a la etapa de procesado.



Ilustración 15: Tanques apilados tras la descarga del camión.

## Diseño del sistema de trazabilidad.

Tras la explicación del proceso de producción de la granja en sus líneas generales en el capítulo anterior, hay que resaltar cada uno de los requerimientos y datos que se necesitan recoger en cada uno de los pasos para llevar a cabo el control. En este caso, gracias al sistema que se está exponiendo, se consigue la trazabilidad de los productos, desde que son empaquetados en la planta de procesado, hasta su origen. La información recolectada incluye datos de tiempos de tránsito, pesca, etc., que estén asociados a ellos [2].

El nuevo sistema tiene, no sólo un valor para controlar y poder optimizar los procesos dentro de la propia empresa, sino también pretende servir para actualizar en los métodos de trabajo que actualmente emplean papel y lápiz, así como lograr una mayor seguridad en la recogida.

En los siguientes puntos se precisan los datos que son actualmente tomados por la empresa y como se integran dentro del sistema EPC, que en este caso coincidirán casi siempre con las fases estándar de su proceso.

### 4.1 Engorde.

DATOS REQUERIDOS
Fecha y hora de carga
Lote de origen
Tipo de pez
Fecha y hora de recepción
ID de la jaula de engorde

Tabla 5: Datos requeridos para el evento generado durante la fase Engorde.

La tabla que encabeza este texto recoge los datos que se necesitan en este punto. Los tres primeros datos son tomados de una hoja provista por el barco que traslada a los alevines y que después se complementa, en una nueva hoja, con la información de recepción e ID de la jaula de engorde.

Atendiendo a la descripción del proceso en el punto 3.1, como un traspaso por lotes, desde la cámara del barco que los transporta hacia una jaula de engorde, se hará uso de un *Object event* que recoja todos los datos requeridos, empleando los siguientes valores:

- **EPCbatchID** : Lote identificativo de los peces y que será de tipo SGTIN (*Serialized Global Trade Item Number*) por su carácter de producto, componiéndose de la siguiente manera:  
20000005/product/00000P/XXXXYYYY
  - 20000005: Identificador asignado a la empresa (en este caso Culmarex).
  - Product: Detalla que es un objeto de un sólo uso.
  - 00000P: Da información sobre el producto, siendo 2 si es dorada o 3 si es lubina.
  - XXXX: Número de lote en origen.
  - YYYY: Número de lote en granja, junto con XXXX conformará el número de serie concreto de este producto.
- **Action**: Toma el valor *ADD* pues se vincula por primera vez a la jaula de engorde.
- **BizStep**: Con el valor de *commissioning* se informa de que se está poniendo dentro de una caja, en este caso jaula.
- **Disposition**: Se sitúa en *active* pues está disponible para su uso.
- **ReadPoint**: El ID del lector que recoge los datos que está en el barco, cuyo identificador es *seaCagesReader*.
- **BizLocation**: La ID de la jaula de engorde donde se deposite el lote, *growingCage#n*.

La *Ilustración 16* recoge un resumen de la información a rellenar para este paso.

ObjectEvent	
eventTime:	
recordTime:	
eventTimeZoneOffset:	
epcList:	<b>EPCbatchID</b>
action:	<b>ADD</b>
bizStep:	<b>commissioning</b>
disposition:	<b>active</b>
readPoint:	<b>GrowingFarm.Culmarex.seaCagesReader</b>
bizLocation:	<b>GrowingFarm.Culmarex.growingCage#1</b>
bizTransactionList	
bizTransactionTypeID:	
bizTransactionID:	

*Ilustración 16: Object event para el paso Engorde.*

## 4.2 Comercial.

En este paso no se tiene un requerimiento concreto de datos, pues se lleva a cabo la cumplimentación de una hoja de control más adelante, desde el punto de vista de la empresa.

Por el contrario, en este nuevo sistema se sigue recabando más información en el instante mismo en el que se produce el trasvase a la jaula comercial, para mantener actualizada la trazabilidad de los peces.

Siguiendo la línea del paso anterior, pues se trata de un traspaso de un lote más pequeño dentro del lote que hay en la jaula de engorde, se emplea un *Object event* con los siguientes campos:

- **CommercialID** : Subconjunto del lote almacenado en la jaula de engorde y que tiene la forma de SGTIN (*Serialized Global Trade Item Number*) por su carácter de producto, componiéndose de la siguiente manera:  
20000005/product/00001P/XXXXYYYYZZ
  - XXXXYYYY: EPCBatchID al que pertenecen.
  - ZZ: Número de traspaso que se produce desde dicha jaula de engorde (1-99).
  - P: 2 si es dorada o 3 si es lubina. Nótese que se ha substituido un 0 por un 1 en el identificador de la clase, con lo que se denota un avance en la línea de producción (de la jaula de engorde a la jaula comercial).
- **Action**: Toma el valor *ADD* pues se vincula por primera vez a la jaula comercial.
- **BizStep**: Con el valor de *commissioning* se informa de que se está poniendo dentro de una caja, en este caso jaula.
- **Disposition**: Se sitúa en *active* pues está disponible para su uso.
- **ReadPoint**: Es el ID del lector que recoge los datos. En este caso vuelve a ser el lector manual que está en el barco, cuyo identificador es *seaCagesReader*.
- **BizLocation**: La ID de la jaula comercial donde se deposite el lote, *commercialCage#n*.

La *Ilustración 17* recoge un resumen de la información que debe ser cumplimentada durante el paso Comercial.

ObjectEvent	
eventTime:	
recordTime:	
eventTimeZoneOffset:	
epcList:	<b>CommercialID</b>
action:	<b>ADD</b>
bizStep:	<b>commissioning</b>
disposition:	<b>active</b>
readPoint:	<b>GrowingFarm.Culmarex.seaCagesReader</b>
bizLocation:	<b>GrowingFarm.Culmarex.commercialCage#1</b>
bizTransactionList	
bizTransactionTypeID:	
bizTransactionID:	

*Ilustración 17: Object event para el paso Comercial.*

### 4.3 Pesca.

En este punto se inicializan las pescas que se llevarán a cabo en esa jornada y que posteriormente serán completadas con más datos en los siguientes puntos. Con este método se pretende facilitar la tarea a los operarios, ya que podrán dejar inicializadas las órdenes de pesca del día.

Durante el proceso de pesca se rellenan varios tanques con agua y hielo, y se cargan de peces. Por cada proceso de pesca se genera un Object Event con la siguiente información:

- **HarvestOrderID** : Identificador de la pesca que se va a llevar a cabo y que tiene la forma de SGTIN (*Serialized Global Trade Item Number*) por su carácter de producto, componiéndose de la siguiente manera:  
20000005/product/00011P/XXXXYYYYZZH
  - XXXXYYYYZZ: CommercialID al que pertenecen.
  - H: Número de pesca realizada hasta el momento sobre una jaula comercial dada (1-9).
  - P: 2 si es dorada o 3 si es lubina. Nótese que se ha substituido un 0 por un 1 adicional, con lo que se denota un avance en la línea de producción.
- **Action**: Toma el valor *ADD* pues se vincula por primera vez a la jaula comercial.
- **BizStep**: Con el valor de *picking* se está produciendo un proceso de selección con objeto de completar una orden.
- **Disposition**: Se sitúa en *in\_progress* pues está en curso la operación.
- **ReadPoint**: Es el ID del lector que recoge los datos. En este caso vuelve a ser el lector manual que está en el barco, cuyo identificador es *seaCagesReader*.
- **BizLocation**: Es el ID del barco que procede a la pesca, *boat*.

La *Ilustración 18* recoge un resumen de la información que debe ser cumplimentada durante el paso Pesca.

ObjectEvent	
eventTime: recordTime: eventTimeZoneOffset:	
epcList:	<b>HarvestOrderID</b>
action:	<b>ADD</b>
bizStep:	<b>picking</b>
disposition:	<b>in_progress</b>
readPoint:	<b>GrowingFarm.Culmarex.seaCagesReader</b>
bizLocation:	<b>GrowingFarm.Culmarex.boat</b>
bizTransactionList bizTransactionTypeID: bizTransactionID:	

Ilustración 18: *Object event* para el paso de Pesca.

## 4.4 Empaquetado.

Completa, junto al punto anterior 4.3, lo que se describía en el punto 3.2 como *Pesca*, rellenándose unas hojas de control con los siguientes datos:

Datos requeridos
Fecha de pesca
Hora de comienzo
Hora de finalización
Número de pesca
ID del tanque
Lote de granja
ID jaula de engorde
Datos requeridos
ID jaula comercial
Fecha de la última comida
Hora de la última comida

Tabla 6: Datos requeridos para la generación de los eventos diseñados para Pesca y Empaquetado respectivamente.

De cara a nuestro sistema diseñado se requieren ir completando los datos ya tomados en pasos anteriores y complementarlos con propios del sistema. En este caso se hace uso de un *Aggregation event* que nos identificará que contenedores (tanques) estarán unidos a una determinada orden de pesca, pudiendo obtener cualquier otra información necesaria.

A continuación los campos a completar:

- **parentID:** El identificador EPC de la orden de pesca a la que queremos unir los tanques, *HarvestOrderID*.
- **childEPCs:** Los EPCs de los tanques a unir, que son de tipo *GRAI (Global Returnable Asset Identifier)* debido a que son recursos reutilizables. Son de la siguiente manera:  
20000005/asset/00002/SN
  - 00002: Indica el tipo recurso reutilizable que se está empleando (contenedor).
  - SN: Número de serie propio de cada contenedor.
- **Action:** Toma el valor *ADD* pues los hijos, en este caso tanques, se están uniendo al padre, representado en esta ocasión por la orden de pesca.
- **bizStep:** Con su valor en *staging\_outbound*, se informa de que los contenedores están cerrados y listos para ser transportados a la siguiente fase.
- **Disposition:** Tiene como valor *container\_closed* como consecuencia o precisión del *business step* anterior.
- **ReadPoint:** Es el ID del lector que recoge los datos. En este caso vuelve a ser el lector manual que está en el barco, cuyo identificador es *seaCagesReader*.
- **BizLocation:** Es el ID del barco que procede a la pesca, *boat*.

AggregationEvent	
eventTime:	
recordTime:	
eventTimeZoneOffset:	
parentID:	HarvestOrderID
childEPCs:	FishTankIDs
action:	ADD
bizStep:	staging_outbound
disposition:	container_closed
readPoint:	GrowingFarm.Culmarex.seaCagesReader
bizLocation:	GrowingFarm.Culmarex.boat
bizTransactionList:	
bizTransactionTypeID:	
bizTransactionID:	

Ilustración 19: *Aggregation event* para el paso *Empaquetado*.

## 4.5 Envío.

En el procedimiento actual, los contenedores son enviados junto con las *hojas de control de pesca*, de manera que exista un seguimiento de todo el proceso. En el sistema diseñado, por el contrario, simplemente se deben añadir estos contenedores a los respectivos camiones que los transportan, no haciéndolo con las *órdenes de pesca* por la problemática de que dos camiones distintos transportasen cajas de una misma orden, con el consiguiente problema para depurar la trazabilidad. En esta fase se generará un *Aggregation Event* que unirá los tanques al camión que los llevará hasta la factoría.

La información a tomar es la siguiente:

- **parentID:** El identificador EPC del camión que transporta las cajas, es de tipo *GRAI* (*Global Returnable Asset Identifier*) debido a que es un recurso reutilizable. Tiene la siguiente forma:  
20000005/asset/00001/SN
  - 00001: Indica el tipo de recurso reutilizable que se está utilizando (camión).
  - SN: Número de serie, en este caso los cuatro dígitos de su matrícula.
- **childEPCs:** Los EPCs de los tanques a unir.
- **Action:** Toma el valor *ADD* pues los hijos, en este caso tanques, se están uniendo al padre, representado en esta ocasión por el camión.
- **bizStep:** Informa con el valor a *departing* que se dispone a salir desde su localización actual (puerto) hasta su destino (planta de procesado).
- **Disposition:** Tiene como valor *in\_transit* precisando el *business step* anterior.
- **ReadPoint:** Recoge los datos el lector que está en el muelle, de ahí *DockReader*.
- **BizLocation:** Se sitúa en el muelle donde llega el barco, *Dock*.

En la *Ilustración 20*, se plasman todos los datos necesarios para la fase de Envío.

AggregationEvent	
eventTime: recordTime: eventTimeZoneOffset:	
parentID: childEPCs action: bizStep: disposition: readPoint: bizLocation:	<b>TruckID</b> <b>FishTankIDs</b> <b>ADD</b> <b>departing</b> <b>in_transit</b> <b>GrowingFarm.Culmarex.DockReader</b> <b>GrowingFarm.Culmarex.Dock</b>
bizTransactionList bizTransactionTypeID: bizTransactionID:	

*Ilustración 20: Aggregation event para el paso Envío.*

## 4.6 Recepción.

En el paso siguiente, los tanques, que han viajado con las *hojas de control de pesca*, son identificados y se añaden a unas nuevas hojas de recepción que existen en la dársena de la empresa donde descarga todo.

Dentro del diseño del sistema de trazabilidad, este trabajo se realiza en dos fases, siendo este punto la primera de ellas y se observa la llegada del camión a la dársena de la empresa, haciendo uso de un *Object Event*. Los datos son los siguientes:

- **truckID:** El identificador EPC del camión que llega a la planta.
- **Action:** Toma el valor *OBSERVE* para resaltar que ha sido detectada su llegada.
- **bizStep:** Ha llegado a la dársena de ahí que sea *arriving*.
- **Disposition:** Como todavía no ha terminado su cometido, se sitúa en *in\_progress*.
- **ReadPoint:** La lectura se toma en el muelle de carga de la planta, de ahí *PackingProcessingPlant.DockReader*.
- **BizLocation:** La localización es la recepción de la planta, *ReceivingRoom*.

En la *Ilustración 21* tenemos un resumen de los datos necesarios para la Recepción.

ObjectEvent	
eventTime:	
recordTime:	
eventTimeZoneOffset:	
epcList:	TruckID
action:	OBSERVE
bizStep:	arriving
disposition:	in_progress
readPoint:	PackingProcessingPlant.DockReader
bizLocation:	PackingProcessingPlant.ReceivingRoom
bizTransactionList	
bizTransactionTypeID:	
bizTransactionID:	

Ilustración 21: Object event para el paso Recepción.

## 4.7 Desempaquetado.

Este punto, como se comentaba en el anterior apartado, completa la rutina de toma de información de la empresa y además libera los recursos reutilizables de la empresa de cara a la trazabilidad. Consta de dos eventos distintos, el primero de ellos separa la unión entre cajas y camión resaltando que ya están descargadas y en el segundo se observa como éste (camión) pasa a estar inactivo para posibles usos futuros.

El evento de descarga de cajas es un *Aggregation Event* y contiene los siguientes campos:

- **parentID:** El identificador EPC del camión que transporta las cajas.
- **childEPCs:** Los EPCs de los tanques que porta en su interior.
- **Action:** Toma el valor *DELETE* pues los hijos han sido descargados y no se da más esa unión.
- **bizStep:** Informa con el valor a *receiving* que ya se está recepcionando la carga del camión.
- **Disposition:** Tiene como valor *in\_progress* pues no está concluido.
- **ReadPoint:** La lectura se toma en el muelle de carga de la planta, de ahí *PackingProcessingPlant.DockReader*.
- **BizLocation:** La localización es la recepción de la planta, *ReceivingRoom*.

En la *Ilustración 22* se recogen todos los datos necesarios para este punto de Desempaquetado.

AggregationEvent	
eventTime:	
recordTime:	
eventTimeZoneOffset:	
parentID:	<b>TruckID</b>
childEPCs:	<b>FishTankIDs</b>
action:	<b>DELETE</b>
bizStep:	<b>receiving</b>
disposition:	<b>in_progress</b>
readPoint:	<b>PackingProcessingPlant.DockReader</b>
bizLocation:	<b>PackingProcessingPlant.ReceivingRoom</b>
bizTransactionList:	
bizTransactionTypeID:	
bizTransactionID:	

*Ilustración 22: Primer evento dentro del Desempaquetado.*

En segundo lugar, cambiamos el estado del camión para dejarlo disponible para futuros usos. Esto se realiza con un *Object Event*:

- **truckID:** El identificador EPC del camión que llega a la planta.
- **Action:** Toma el valor *OBSERVE* pues se encuentra allí.
- **bizStep:** Sigue como en el evento anterior en *receiving*.
- **Disposition:** Cambia su estado hasta próximos transportes a *unknown*.
- **ReadPoint:** La lectura se toma en el muelle de carga de la planta, *PackingProcessingPlant.DockReader*.
- **BizLocation:** La localización es la recepción de la planta, *ReceivingRoom*.

La *Ilustración 23* se corresponde con los datos que conforman este segundo evento en el paso de Desempaquetado.

ObjectEvent	
eventTime:	
recordTime:	
eventTimeZoneOffset:	
epcList:	<b>TruckID</b>
action:	<b>OBSERVE</b>
bizStep:	<b>receiving</b>
disposition:	<b>unknown</b>
readPoint:	<b>PackingProcessingPlant.DockReader</b>
bizLocation:	<b>PackingProcessingPlant.ReceivingRoom</b>
bizTransactionList:	
bizTransactionTypeID:	
bizTransactionID:	

*Ilustración 23: Segundo evento dentro del Desempaquetado.*

Al contrario que sucede con los camiones, los tanques no son separados de la orden de pesca (*HarvestOrderID*) a la que estaban unidos hasta que no son vaciados en etapas posteriores (se da durante la fase de clasificación ya en la factoría).

Para finalizar, en la *Ilustración 24* se recoge de manera gráfica todo el proceso y los diversos eventos que tienen lugar.

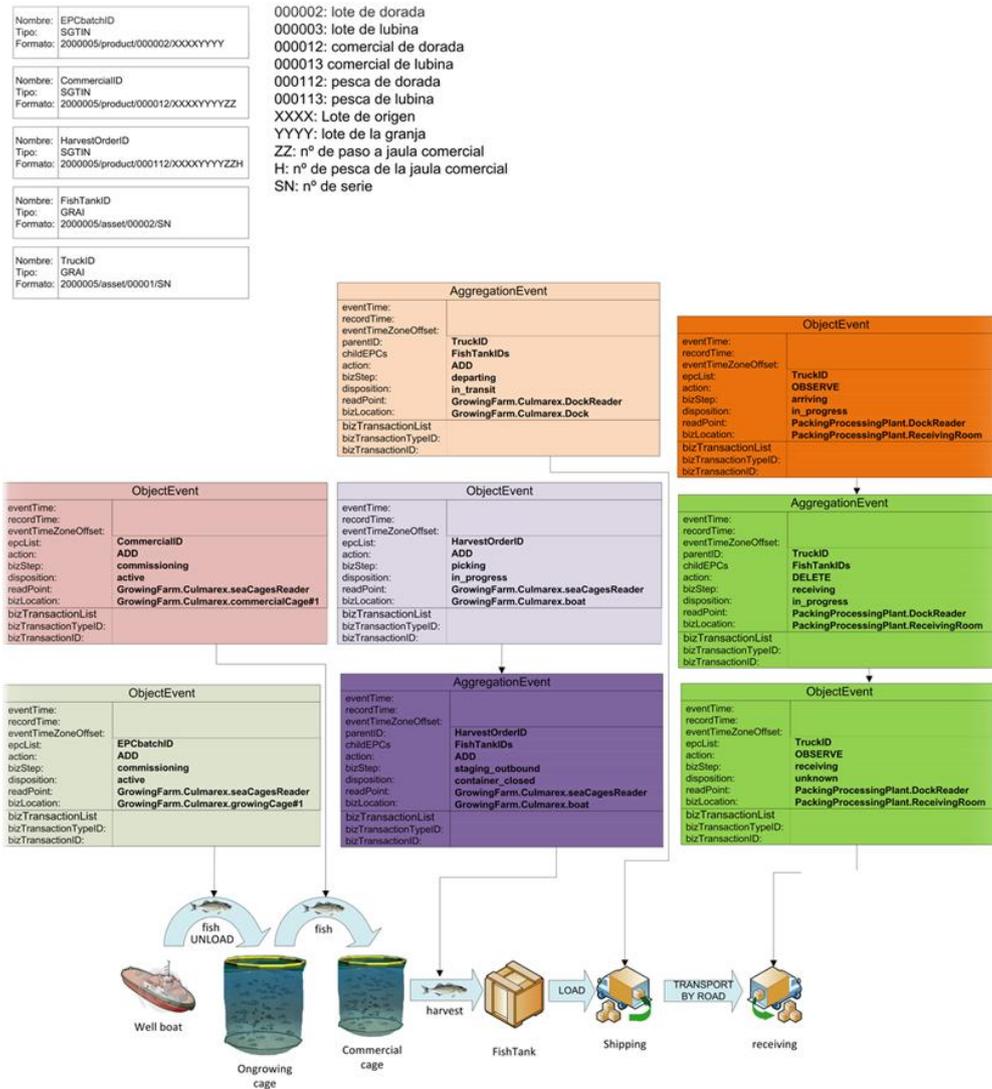


Ilustración 24: Resumen del diseño de eventos para el sistema de trazabilidad.

## Descripción de los equipos empleados y software desarrollado.

En este punto se pretende dar información de los equipos empleados para conseguir hacer funcionar el sistema diseñado, tanto a nivel físico de los mismos como el software que implementa la idea [1].

### 5.1 Hardware.

En este segmento del proyecto se requieren dos cosas fundamentales (sin tener en cuenta la infraestructura que recibe los datos, los procesa y almacena) que son el *handheld* o lector y los *tags* que irán variando según en el entorno en el que estén. Gracias a ellos se podrá llevar el control en cada una de las distintas zonas explicadas en los puntos anteriores, de manera que se sustituya el método de lápiz y papel que emplea la empresa en la que se realizan las pruebas por el sistema electrónico propuesto.

#### 5.1.1 Handheld.

El entorno en el que se producen las lecturas está caracterizado por la necesidad de movilidad, pues se desarrolla en un barco que se desplaza a distintas localizaciones según las tareas del día, organiza tanques con las capturas del día en su cubierta, etc., con lo que se requiere un dispositivo móvil de un tamaño manejable por el operario, que resista las condiciones ambientales del entorno (humedad, golpes) y que sea capaz de desempeñar la interrogación de los tags situados en las diferentes jaulas de engorde, comerciales y en los tanques de transporte.

Para este cometido se cuenta con el lector *Marktrace UHF Handheld Reader MR6081*, un equipo como el mostrado en la *Ilustración 25*, que cuenta con las siguientes características:

- Capacidad de lectura y escritura en elementos RFID, y almacenamiento en el propio aparato.
- Tecnologías de comunicación Bluetooth, WIFI, GPS, GPRS y lectura de códigos de barras de 1D y 2D.
- Equipo ergonómico, con buen agarre y cinta de seguridad para sujetarlo.
- Certificación IP65, con lo que es resistente a la entrada de polvo y agua.



Ilustración 25: Lector Marktrace UHF Handheld Reader MR6081A.

Sus parámetros técnicos son [1]:

Modelo	MR6081A
Protocolo	ISO18000-6B, ISO18000-6C (EPC GEN2)
Rango de frecuencias	Estándar ISM 902-928MHz o ISM 865-868MHz
Modo de operación	FHSS
Potencia RF	0-30dBm, ajustable vía software
Distancia de identificación	Lectura 2m (tipo tarjeta), escritura 1,5m (tipo tarjeta)
Sistema operativo	Microsoft WinCE 5.0
CPU	C2440 Samsung 400MHz
Almacenamiento	64MB ROM Flash, 64MB ROM
Interfaces de datos	Bluetooth, WIFI, códigos de barras de 1D y 2D, GPS, GPRS
Baterías	3,7V y cargador de red
Autonomía	Alrededor de 9 horas
Certificación de protección	IP65
Peso neto	0,5 Kg
Dimensiones	210 mm x 90 mm x 53 mm
Temperatura de operación	-20°C ~ +80°C

Tabla 7: Datos técnicos del lector Marktrace UHF Handheld Reader MR6081A.

### 5.1.2 Tag.

En la elección del modelo de tag, que ha de ir situado en las jaulas de engorde y comercial, se debe tener en cuenta que estarán a la intemperie y sujetos a una estructura de metal. En consecuencia, el material del que este compuesto el encapsulado debe de ser capaz de resistir esos factores y operar correctamente.

Se han escogido para esta tarea los tags modelo *Halo* de la marca *Confidex*, que como resaltan en la descripción del producto en su documentación [6]:

*“Combina un peso bajo, tamaño pequeño y un encapsulado duradero sin sacrificar rendimiento. El tag ha sido diseñado para ser fijado fácilmente mediante el uso de cables a través de sus orificios, pegamento o un soporte magnético especialmente diseñado para el traslado del tag de un lado a otro si se necesitase.”*

En la *Ilustración 26* se observa su forma y además la forma sencilla para ser fijado en una estructura similar a la de las jaulas de Culmarex.



Ilustración 26: Tag Confidex Halo.

Tipo	Transpondedor pasivo de UHF RFID clase 1 GEN2
Protocolo	EPC Global clase 1 GEN2 ISO 18000-6C
Frecuencia de operación	885-869 MHz (EU), 902-928 (EEUU), 952-955 MHz (JPN)
Opciones IC	NXP UCODE G2XM
Memoria EPC	Hasta 240 bits
Memoria extendida	512 bits
Rango de lectura	Hasta 5-7 m
Posibilidad de fijar	Superficies metálicas
Material del encapsulado	PC/ABS
Color	Gris oscuro
Peso	7 g
Dimensiones	60 x 12 x 14 mm
Temperatura de operación	-35°C ~ +85°C
Resistencia	IP67, resistencia a rayos ultravioleta y agua del mar

Tabla 8: Datos técnicos del tag Confidex Halo.

## 5.2 Software.

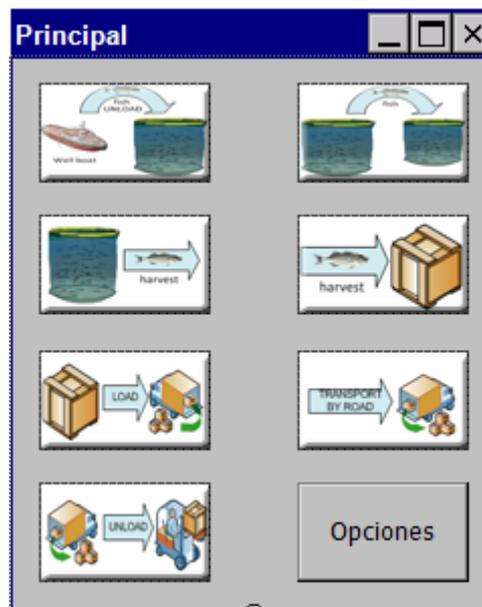
Al comienzo del proyecto había que definir el lenguaje más apropiado para la implementación del software, teniendo dos opciones: Java o C#. A pesar de que el primero de ellos destacaba por ser el lenguaje en el que estaba escrito parte del software de recogida de los datos para el servidor y de la cantidad de ayuda que se proveía, la imposibilidad de encontrar para el aparato una máquina virtual de probada solvencia hizo que sólo quedase en liza la opción de C#, que tenía sus propias ventajas:

- Visual Studio 2005 como entorno de programación ya adaptado a Windows CE.
- No necesidad de pasar por una máquina virtual (eficiencia).

Como ya se tiene el diseño del sistema de trazabilidad, sólo queda llevarlo al terreno de la programación, siempre teniendo en cuenta que debe de ser sencillo, fácil de utilizar, con una interfaz preparada para el entorno de trabajo (pocas selecciones, tamaños grandes) y que, en definitiva, no suponga un problema para el operario que tiene que centrarse en el desempeño de su puesto.

Siguiendo esas líneas, el programa se estructura según el sistema de trazabilidad diseñado en la sección 4, con lo que no se pierde la orientación al proceso de producción de la empresa. El programa comienza con una pantalla de presentación en la que se referencian con imágenes cada una de las distintas fases (de izquierda a derecha y de arriba hacia abajo la pantalla se muestra en la *Ilustración 27*):

- Engorde.
- Comercial.
- Pesca.
- Empaquetado.
- Envío.
- Recepción.
- Desempaquetado.
- Opciones.



*Ilustración 27: Captura de la aplicación, pantalla Principal.*

Antes de que se comience a operar con el programa (únicamente el primer encendido del programa o si hay cualquier cambio en la IP del servidor al que se envíen los datos), habrá que entrar en *Opciones* e introducir el valor de la IP, que ya quedará registrada hasta que se requiera cambiar ese dato y persiste más allá del apagado del equipo o programa, pues va almacenada en un archivo aparte. El menú de opciones se muestra en la *Ilustración 28*.

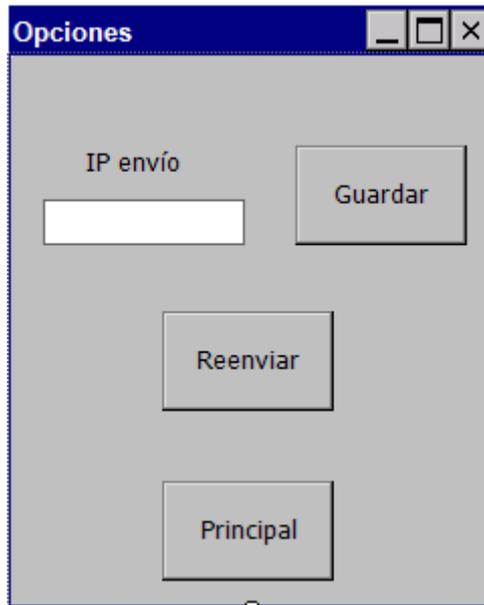


Ilustración 28: Captura de la aplicación, pantalla *Opciones*.

El botón *Reenviar* dará la posibilidad de volver a enviar eventos que por falta de cobertura no hayan podido ser enviados, avisando de la cantidad de éstos que lo hagan correctamente.

Una vez configurado el equipo, ya se puede proceder a operar en las distintas pestañas del programa según sea necesario y teniendo en cuenta el siguiente diagrama de flujo:

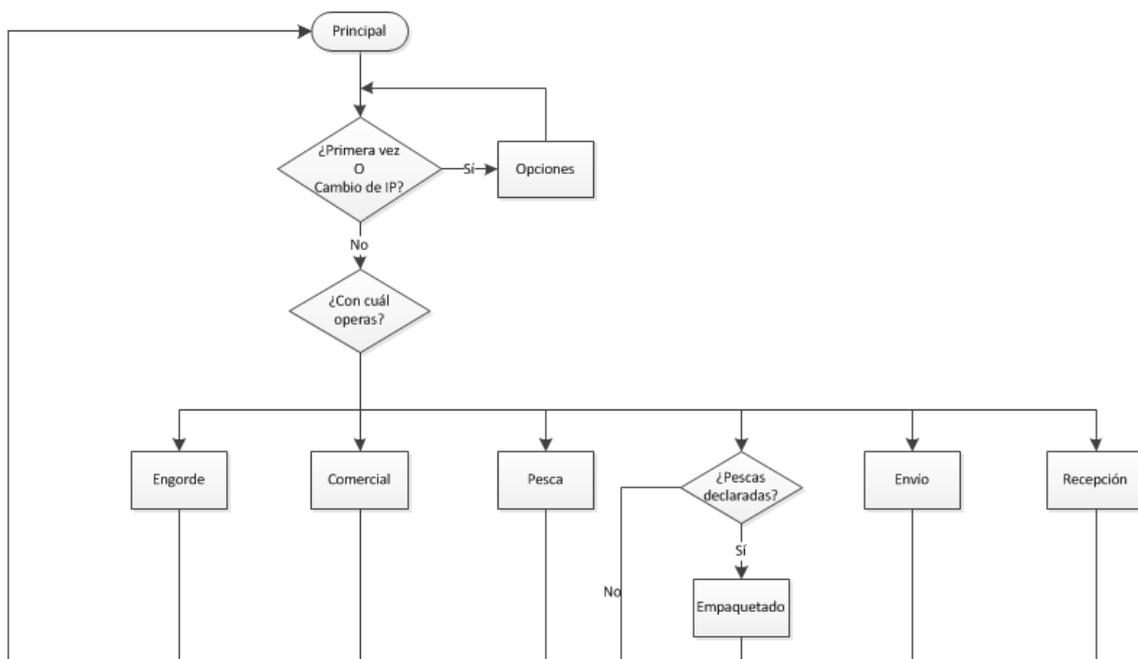


Ilustración 29: Diagrama de flujo de la aplicación.

Como recoge la *Ilustración 29*, el programa comienza en una pantalla principal, si es la primera vez que se inicia el programa o se ha cambiado la IP del servidor donde se envían los datos, se deberá, pulsar *Opciones* y configurar los parámetros. Ya en la pantalla principal, se podrá escoger cualquiera de los botones que nos llevan a cada una de las pestañas según en la fase en la que se esté operando. Una vez concluida cada una de ellas, se retornará a la pantalla principal.

### 5.2.1 Engorde.

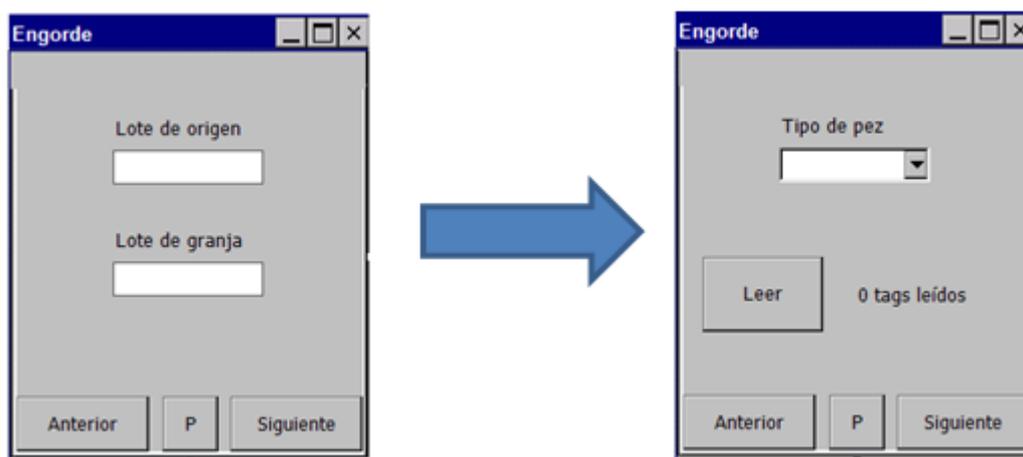


Ilustración 30: Capturas de pantalla de *Engorde*.

Como se observa en la *ilustración 30*, hay que recabar cuatro datos:

- **Lote de origen:** Cuatro dígitos que vienen recogidos del barco que traslada los alevines.
- **Lote de granja:** Cuatro dígitos que identifican la granja de origen que serán puestos a la llegada del barco que traslada los alevines.
- **Tipo de pez:** Selección entre lubina o dorada.
- **Leer:** Identifica el tag sujeto a la jaula de engorde donde se vuelcan los alevines que llegan.

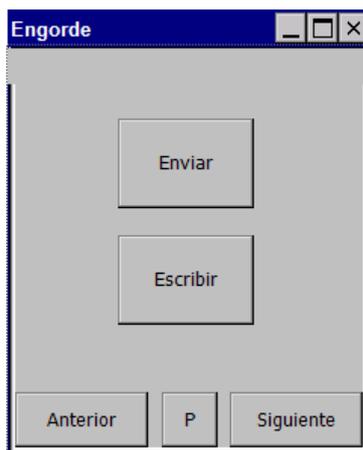


Ilustración 31: Última ventana dentro de *Engorde*.

Ya sólo queda enviarlo utilizando el botón de *Enviar*, tal y como se observa en la *Ilustración 31*, y, una vez avise el programa de que se envió la información, escribir en la memoria del tag mediante el uso del botón *Escribir* [5] de modo que se pueda seguir la trazabilidad del producto. Estos datos a escribir, que serán necesarios en los siguientes pasos (tal y como se ha descrito en anteriores secciones de este proyecto), son los siguientes:

- Tipo de pez.
- *BatchID*.
- Número de traspaso inicializado a 0 (no ha existido traspaso alguno a jaula comercial).

### 5.2.2 Comercial.

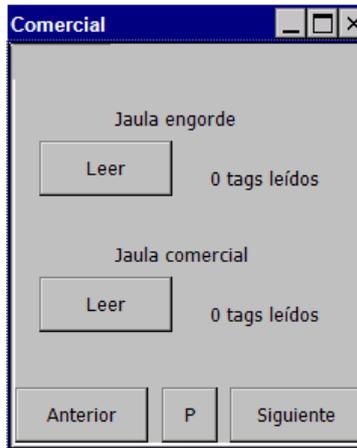
Esta pestaña, por motivos espaciales evidentes pues las jaulas de engorde se sitúan a una cierta distancia de las comerciales, la forma de proceder es ligeramente distinta para integrarlo dentro de los procesos que actualmente están ejecutando en la empresa.



Ilustración 32: Diagrama de flujo de la pestaña *Comercial*.

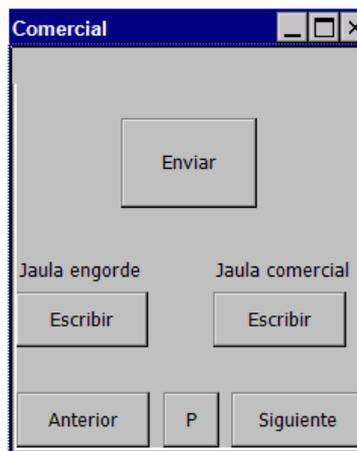
Como se observa en la *Ilustración 32*, en primer lugar se toma lectura del tag de la jaula de engorde (en este caso de la memoria de usuario) para poder, posteriormente, conformar el EPC de este traspaso.

A continuación actualizamos el valor de traspaso (va existir un movimiento de peces desde la jaula de engorde a la comercial) de la memoria de dicho tag, con lo que no habría que volver a la jaula de engorde al finalizar el proceso si no existe otro movimiento de peces. En la *Ilustración 33* se tiene la interfaz para esta pestaña de Comercial.



**Ilustración 33:** Primer paso dentro de la pestaña *Comercial*.

Una vez cargado el pescado, el barco se dirige a la jaula comercial para depositarlo en ella. El operario se mantiene a la espera (sin cerrar la aplicación o pestaña) hasta la llegada del barco a la jaula comercial pertinente en donde procederá a la lectura del tag y envío de los datos. Ya sólo resta almacenar los datos necesarios en los bancos de memoria del tag para que estén disponibles en los siguientes pasos. La *Ilustración 34* muestra la pestaña Comercial en el momento de envío y actualización de datos.



**Ilustración 34:** Paso de envío y actualizaciones dentro de *Comercial*.

### 5.2.3 Pesca.

Esta parte del proceso consta de tres pasos:

- Lectura de la memoria del tag de la jaula comercial.
- Envío de los datos.
- Actualización de la información en la memoria del tag de la jaula comercial.

Toda la información necesaria para los pasos siguientes quedará registrada en unos archivos dentro del equipo de lectura, con lo que estará inmediatamente accesible y sin dependencia de tener cobertura de red con la estación base (lo que puede ser complicado dada la localización de las jaulas en alta mar). Cada vez que se requiera dar de alta una pesca nueva, se deberá pasar por la pantalla *Principal* (botón *P*).

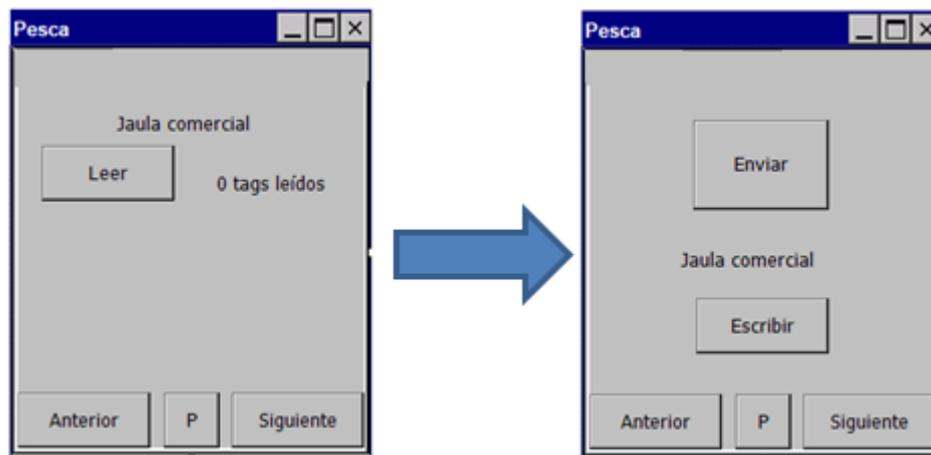


Ilustración 35: Pantallas en pestaña *Pesca*.

La *Ilustración 35* muestra la secuencia de pantallas que hay que completar para inicializar cada una de las órdenes de pesca del día, mientras que en la *Ilustración 36* se tiene un XML de ejemplo, resultado de este proceso, con el apartado *Cajas* con valor "empty" (no tiene ningún tanque unido a esta orden de pesca).

```
<?xml version="1.0" encoding="UTF-8"?>
<Ordenes>
  <Orden>
    <HarvID>123487650101</HarvID>
    <Cajas>empty</Cajas>
  </Orden>
</Ordenes>
```

Ilustración 36: Documento XML ejemplo de guardado de los EPCs de pesca.

### 5.2.4 Empaquetado.

En este caso se reduce la operación a la lectura de los identificadores de las cajas, teniendo en cuenta que se asociarán con la primera orden de pesca que declarasen, en consecuencia, se debe de mantener un orden entre la pesca que se va a enviar y los tanques a los que se va a asociar en este punto del programa.

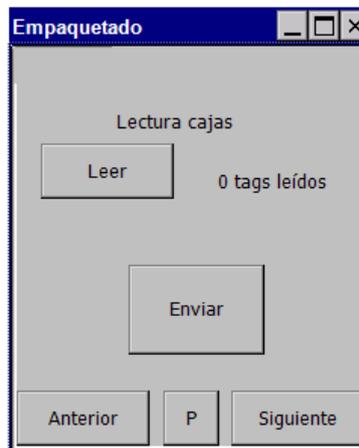


Ilustración 37: Captura de pantalla en el paso de *Empaquetado*.

En la *Ilustración 37* se observa el programa durante su uso en *Empaquetado*, y a continuación, en la *Ilustración 38*, un XML de ejemplo que ha sido actualizado su parámetro *Cajas* a un valor numérico (se corresponde con el número de tanques unidos a esta orden de pesca).

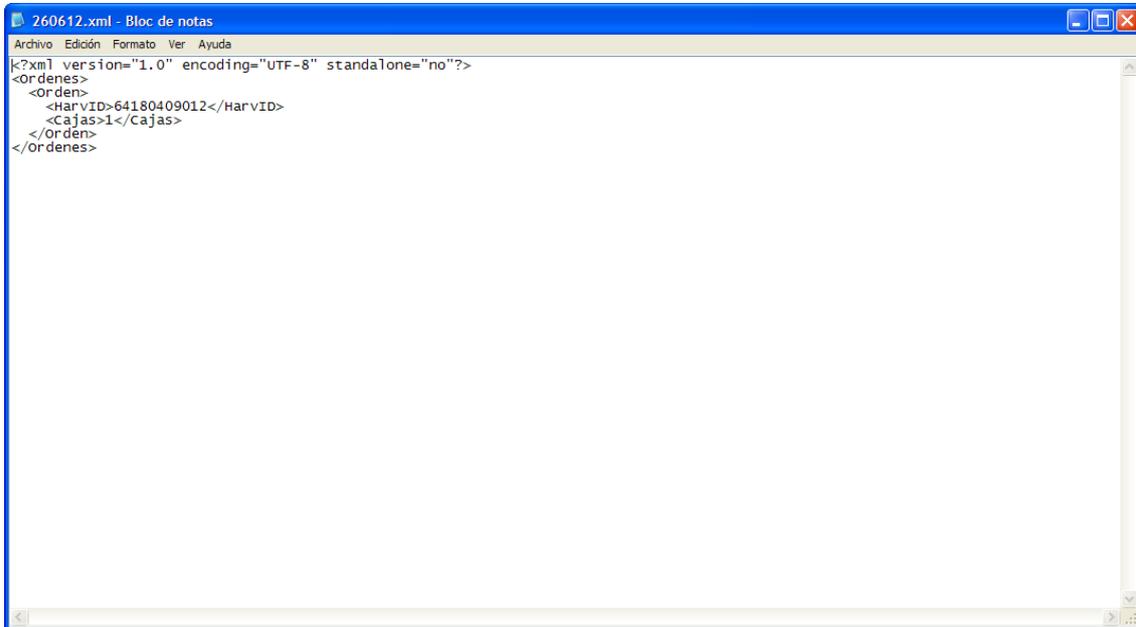


Ilustración 38: Archivo XML ejemplo de un EPC de pesca almacenado ya en uso.

### 5.2.5 Envío.

En este paso se deben de leer las identificaciones de los tanques que van a ser transportados por cada camión (exclusivamente los que traslade uno específico) e introducir los cuatro dígitos de su matrícula. Ya tomados los datos, se puede proceder al envío.



Ilustración 39: Pantallas dentro de *Envío*.

## 5.2.6 Recepción.

En este caso simplemente se necesita insertar los cuatro dígitos de la matrícula del transporte que desee constatar su llegada a la planta, tal y como se observa en la *Ilustración 40*.

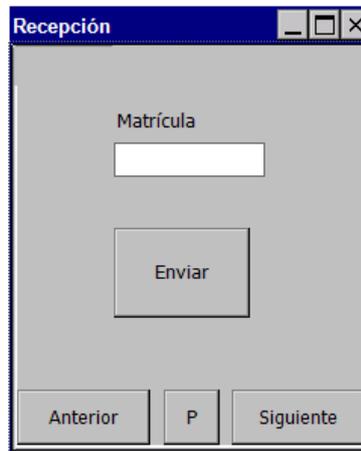


Ilustración 40: Pantalla única dentro de *Recepción*.

## 5.2.7 Desempaquetado.

Este proceso es exactamente igual que el descrito en el punto 5.2.6, deberá leer el tag de cada tanque que se descargue del camión y después introducir los cuatro dígitos de la matrícula en el espacio en blanco. Finalmente podemos enviar los datos.

La *Ilustración 41* muestra la secuencia de pantallas que hay que completar para inicializar cada una de las órdenes de pesca del día

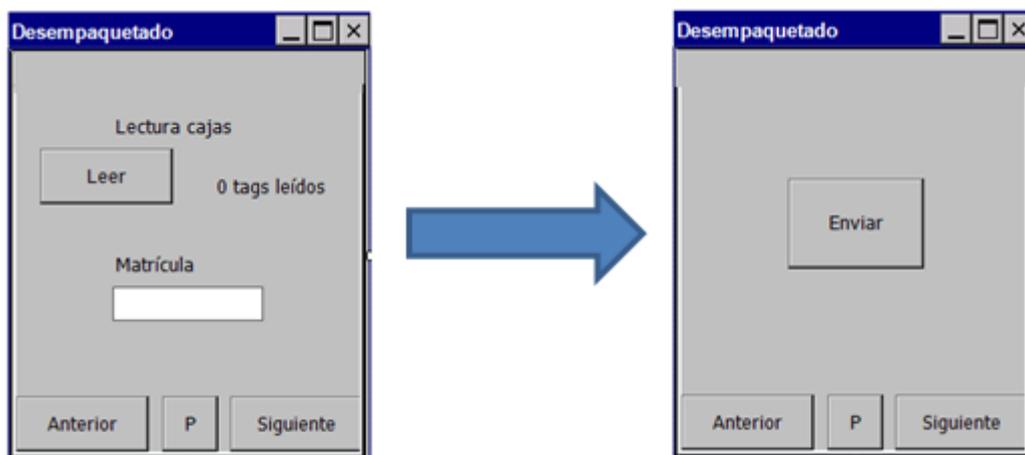


Ilustración 41: Pasos dentro de la pestaña *Desempaquetado*.

## Pruebas y resultados obtenidos.

Tras todo el proceso de conocer el estándar EPC, estudiar el proceso productivo de la empresa en la que está el proyecto centrada, diseñar el sistema de trazabilidad y describir tanto la parte hardware, como el software, se debe poner a prueba todo el sistema diseñado en dos tipos de entorno:

- En el laboratorio, dentro de un entorno controlado y en el que se asumen las mejores condiciones posibles.
- En el terreno, donde se comprueba el funcionamiento y fiabilidad en un día normal dentro de los esquemas reales de trabajo.

En el caso particular de este proyecto, se han realizado tres pruebas, siendo una de ellas de laboratorio y dos de ellas de campo (una parcial dentro de la planta y otra completa).

### 6.1 Prueba de laboratorio.

En esta prueba se pretende ver que el sistema funciona perfectamente y que se integra bien con la parte posterior que tiene lugar en la planta de procesado de la empresa (no es objeto de este trabajo). La idea es verificar que el sistema se podría probar en su entorno definitivo.

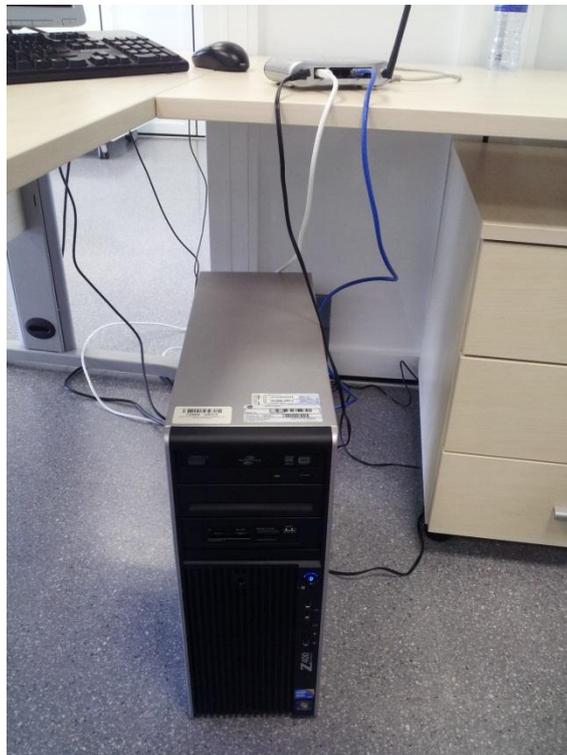


Ilustración 42: Elementos empleados para la prueba.

Para este test contaremos con el hardware que aparece en la *Ilustración 42* y que consta de:

- Dos tags de marca *Wisteq* y aptos para ambientes húmedos, que simularán ser la jaula de engorde y comercial respectivamente.
- Dos tags de tipo pegatina y revestimiento de papel, que simularán ser dos tanques.
- Un lector como el descrito en el punto 5.1.1.
- Infraestructura para la recepción de la información generada (servidor, red WIFI).

Se simulará cada uno de los pasos por los que debería pasar un lote de peces que llega a la planta de procesado, comprobando que el lector genera correctamente los ficheros y que tienen el formato adecuado para, una vez enviados mediante WIFI, la estación base (representada en la *Ilustración 43*), los almacene en su base de datos y se puedan seguir utilizando los mismos en otras partes del proceso.



**Ilustración 43:** Servidor receptor de la información conectado a un router.

Toda la prueba se realiza sin ningún problema, con lo que además de probar a enviar en el momento, se prueba el método de reenvío posterior para solventar el caso de falta de cobertura en la granja real (la *Ilustración 44* muestra la ventana de error que se obtendría en caso de fallo de la transmisión, por ejemplo, por falta de cobertura), por encontrarse en alta mar. En este caso tampoco se obtiene ningún problema.



Ilustración 44: Ejemplo de error en el envío por fallo de cobertura.

## 6.2 Primera prueba de campo.

Al tener lugar las pruebas de campo en dos localizaciones diferentes (zona de pesca y planta de procesado), se decide hacer primero una prueba de campo para testear toda la infraestructura de la red que se tendrá que instalar, además de comprobar la cobertura que tendría el propio lector de mano. La *Ilustración 45* muestra la estación base que se empleará para el proyecto.



Ilustración 45: Servidor receptor de la información.

Se tendrá un servidor instalado en la parte superior de la planta de procesado, conectado a una red WIFI, por donde le llegarán las capturas de los eventos que se vayan leyendo a lo largo de todo el proceso.

Ya en el piso de abajo, los puntos WIFI que aumentarán la cobertura se sitúan dentro de la propia planta, como se ve en la *Ilustración 46*, y serán de vital importancia para que el lector de mano consiga enviar la información correctamente.



**Ilustración 46: Punto de mejora de cobertura WIFI.**

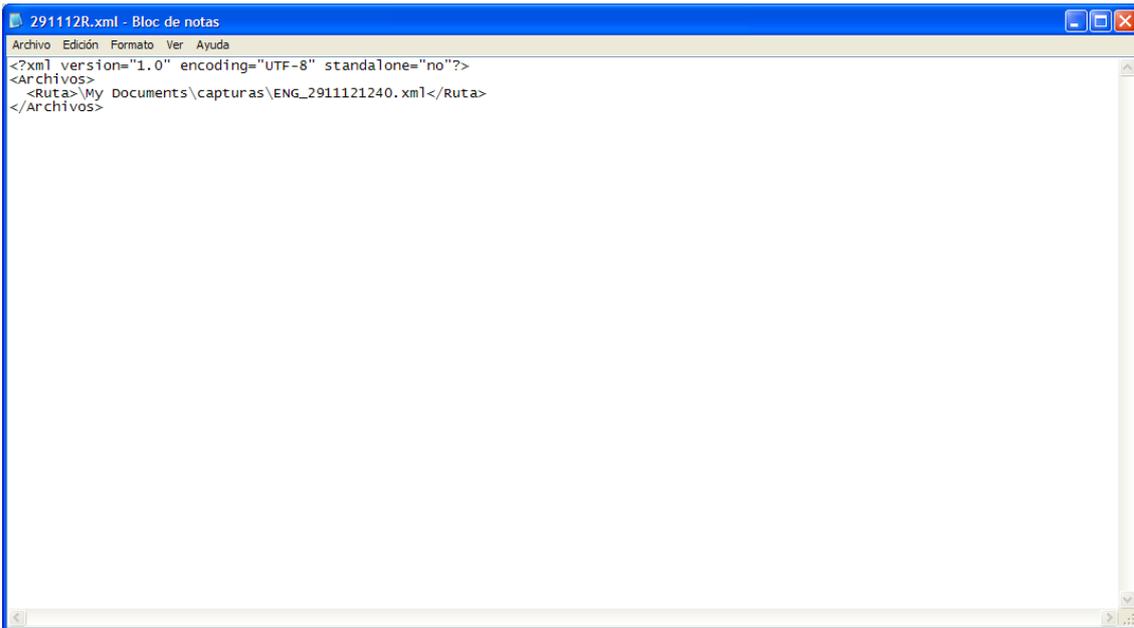
La *Ilustración 47* muestra detalladamente el equipo que se empleará para la prueba, siendo este similar al empleado en la sección 6.1:

- Dos tags de marca *Confidex* como los descritos en el punto 5.1.2 y que simularán ser las cajas de engorde y comercial.
- Dos tags de tipo pegatina y revestimiento de papel, que esta vez irán fijados en los tanques de manera que se pueda testear la lectura en un caso real.
- Un lector como el descrito en el punto 5.1.1.



Ilustración 47: Equipo de pruebas.

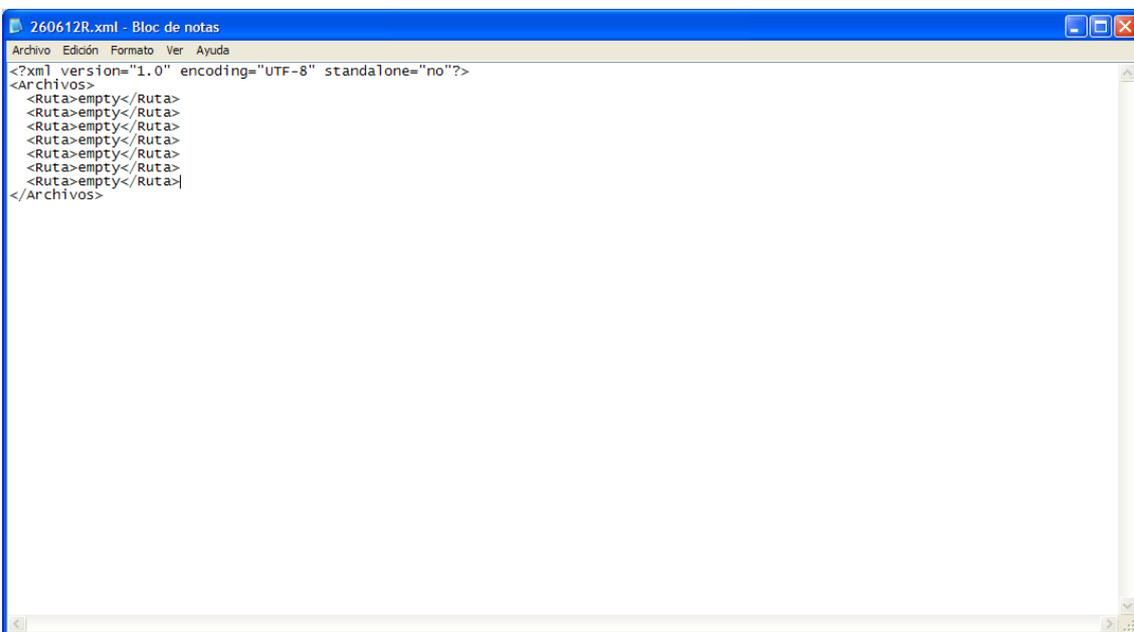
Durante el proceso de lectura, escritura y generación de eventos todo se completa satisfactoriamente a excepción de un problema en la cobertura que se tiene en el exterior. Se observó que la cobertura era variable dejando incluso el lector en un falso estado de conexión que no se advierte hasta que no se produce el error de envío. La *Ilustración 48* se corresponde con un XML generado tras un error de envío y contiene la ruta del fichero que se tendrá que reenviar.



```
291112R.xml - Bloc de notas
Archivo Edición Formato Ver Ayuda
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<Archivos>
  <Ruta>My Documents\capturas\ENG_2911121240.xml</Ruta>
</Archivos>
```

Ilustración 48: Archivo XML con las rutas de los documentos a reenviar.

Para finalizar la prueba se emplea la opción de reenvío, comprobándose, por tanto, su correcto funcionamiento (en la *Ilustración 49* el campo de *Ruta* en todas la entradas tiene el valor de “empty”) y su utilidad para estos casos.



```
260612R.xml - Bloc de notas
Archivo Edición Formato Ver Ayuda
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<Archivos>
  <Ruta>empty</Ruta>
  <Ruta>empty</Ruta>
  <Ruta>empty</Ruta>
  <Ruta>empty</Ruta>
  <Ruta>empty</Ruta>
  <Ruta>empty</Ruta>
  <Ruta>empty</Ruta>
  <Ruta>empty</Ruta>
</Archivos>
```

Ilustración 49: Archivo XML con las rutas de los documentos ya reenviados.

### 6.3 Segunda prueba de campo.

Para esta segunda prueba de campo se pretende probar el sistema completo y no sólo en la planta de procesado, con lo que se probarán tanto equipos como sistemas en situación real. Los tags estarán fijados tanto en las jaulas como en los tanques como se observa en la *Ilustración 50*, poniendo a prueba tanto la capacidad de lectura como la idoneidad del lector para manejarse durante una jornada real de trabajo.



Ilustración 50: Tags fijados a jaula y tanque respectivamente.

Al igual que en las dos pruebas anteriores, no existe ningún problema en la lectura, no así en la posibilidad de envío de los datos que no se puede producir por la ausencia de cobertura en ese punto del proceso (granja situada en alta mar). Hay que resaltar el excelente comportamiento en estas condiciones del *Handheld*, que cumple con su cometido en ergonomía y comodidad de uso. La *Ilustración 51* se corresponde con un momento durante la prueba real en el barco de pesca de Culmarex.



Ilustración 51: Toma de datos durante la prueba en el barco.

Ya en la planta de procesado, tras ir completando las sucesivas pestañas a medida que los operarios van trabajando, si se consigue que todo funcione como se espera al tener, esta vez sí, cobertura en toda la instalación.

Como resumen de esta prueba se observa que la aplicación funciona de acuerdo al plan de diseño. Queda como problema a solventar el diseño de una red wireless que dé servicio cuando se está en el mar o bien, optar por una situación de compromiso y esperar a llegar a la planta de procesado para reenviar los eventos a los servidores.

## Conclusiones.

Los objetivos de trazabilidad con los que se iniciaron este proyecto se han visto cumplidos en esta implementación software, consiguiendo integrarse dentro del sistema general y dotando de movilidad en entornos de trabajo en donde es primordial la misma.

Cabe destacar en el apartado hardware, que se requiere más atención a la hora de leer y sobretodo de escribir [5] en la memoria interna de los tags, al contrario de lo que sucede con la identificación de los mismos que es llevada a cabo con solvencia. En lo referente a la conexión con la estación base se ha empleado una conexión WIFI, que si bien se reveló perfectamente fiable en el entorno de laboratorio, en las pruebas de campo realizadas fue imposible establecer cobertura en el barco, con lo que tuvo que ser reenviado posteriormente una vez ya en la planta de procesado. Pruebas con la tecnología de datos GPRS podrían solventar este problema o la inclusión de puntos de acceso en el barco que la conectasen con el resto de la red.

En revisiones futuras de la aplicación podrían incorporarse nuevas funcionalidades que ampliasen y mejorasen la experiencia de uso y su fiabilidad frente a problemas como cobertura así como la substitución de documentos tipo XML como medio de almacenaje de datos temporales por bases de datos dentro del propio lector de mano.

## Bibliografía.

- [1] “Task 3.1 – Hardware investigation.” – Alfredo Parreño, Fernando Quesada, Pedro Vera, Alejandro Álvarez.
- [2] “Task 2.2 – Detail of Information flow from WP4-8.” – Alfredo Parreño, Fernando Quesada, Pedro Vera, Alejandro Álvarez.
- [3] “Core Business Vocabulary Standard.” – GS1 EPCglobal.
- [4] “EPC Information Services (EPCIS) Version 1.0.1 Specification.” – GS1 EPCglobal.
- [5] “Low Level Reader Protocol (LLRP).” – GS1 EPCglobal.
- [6] “Halo Tags Datasheet.” – Confidex.
- [7] “RFID a guide to radio frequency identification.” – V. Daniel Hunt, Albert Puglia, Mike Puglia.