

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE TELECOMUNICACIÓN  
UNIVERSIDAD POLITÉCNICA DE CARTAGENA



**Proyecto Fin de Carrera**

# **“Validación de protocolos de acceso al medio probabilísticos en sistemas RFID con dispositivos pasivos”**



AUTORA: María José Martínez Belmonte  
DIRECTORA: María Victoria Bueno Delgado

Septiembre / 2008





<b>Autora</b>	María José Martínez Belmonte
<b>E-mail del Autor</b>	majo421@hotmail.com
<b>Director(es)</b>	Mª Victoria Bueno Delgado
<b>E-mail del Director</b>	Mvictoria.bueno@upct.es
<b>Codirector(es)</b>	
<b>Título del PFC</b>	“Validación de protocolos de acceso al medio probabilísticos en sistemas RFID con dispositivos pasivos”
<b>Descriptores</b>	
<p><b>Resumen</b></p> <p>El objetivo del presente proyecto es validar y simular el protocolo anti-colisión EPCGlobal Class-1 Gen-2, estándar actual para sistemas de Identificación por RadioFrecuencia (RFID). Para ello, se han utilizado parámetros hardware y software de un sistema RFID real y se han validado diversas configuraciones.</p> <p>Los resultados de validación y simulación que se han obtenido han permitido recopilar un conjunto de datos que han servido, por un lado, para comprobar que el estándar y las configuraciones que se proponen de él tienen el comportamiento esperado y por otro, que dichos resultados coinciden con los resultados de simulación que previamente se obtuvieron como parte investigadora en el Área de Ingeniería Telemática de la UPCT.</p>	
<b>Titulación</b>	Ingeniería Técnica de Telecomunicaciones, especialidad Telemática
<b>Intensificación</b>	
<b>Departamento</b>	Tecnologías de la Información y las Comunicaciones
<b>Fecha de Presentación</b>	



# AGRADECIMIENTOS

---

Quisiera expresar mi agradecimiento a todas aquellas personas que me han acompañado a lo largo de estos años y que han contribuido de una forma u otra en la elaboración de este proyecto, en especial:

A mis padres y a mi hermana por su apoyo incondicional, ya que sin su ayuda no habría podido llegar hasta aquí.

A mis compañeros y amigos por hacer más fácil este camino llenándolo de momentos especiales.

Y a Victoria, por su ayuda y atención en el desarrollo de este trabajo.



# INDICE DE CONTENIDOS

---

CAPÍTULO 1 .....	- 15 -
INTRODUCCIÓN .....	- 15 -
1.1 Antecedentes .....	- 15 -
1.2 Objetivos del proyecto.....	- 15 -
1.3 Estructura del contenido .....	- 16 -
CAPÍTULO 2 .....	- 17 -
LA TECNOLOGÍA RFID .....	- 17 -
2.1 Introducción a RFID .....	- 17 -
2.1.1 Origen y evolución .....	- 17 -
2.1.2 Elementos de un sistema RFID .....	- 17 -
2.1.2.1 Tag RFID.....	- 18 -
2.1.2.2 Lector de RFID o Reader .....	- 19 -
2.1.3 Funcionamiento básico de un sistema RFID .....	- 19 -
2.1.4 Rangos de frecuencia.....	- 19 -
2.1.5 RFID vs otras tecnologías.....	- 20 -
2.1.5 Aplicaciones de RFID .....	- 22 -
2.1.6 Estándares actuales .....	- 23 -
2.2 Protocolos de identificación / anticolisión en sistemas RFID .....	- 24 -
2.2.1 Protocolos deterministas .....	- 25 -
2.2.2 Protocolos probabilísticos basados en FSA .....	- 25 -
2.2.3 EPCGlobal Class-1 Gen-2 .....	- 26 -
CAPÍTULO 3 .....	- 29 -
Estándar .....	- 29 -
ECPglobal Class-1 Gen-2 .....	- 29 -
3.1 Introducción.....	- 29 -
3.2 Requisitos Hardware .....	- 29 -
3.2.1 Reader .....	- 29 -
3.2.2 Antenas.....	- 30 -
3.2.3 Tags.....	- 30 -
3.3 Implementación software .....	- 31 -
3.2.1 Formato de paquetes EPC .....	- 31 -
3.2.2 Control de errores.....	- 31 -
3.3.3 Seguridad en EPC.....	- 31 -
3.4 Protocolo de comunicación Reader-Tags .....	- 32 -

3.4.1 Monitorización del entorno.....	- 32 -
3.4.2 Protocolo de identificación.....	- 32 -
3.4.3 Volcado de datos.....	- 33 -
3.4.4 Otras funciones.....	- 33 -
3.5 Protocolo anticolidión con trama adaptativa.....	- 33 -
3.5.1 Trama adaptativa ciclo a ciclo.....	- 34 -
3.5.2 Trama adaptativa slot a slot.....	- 34 -
CAPÍTULO 4 .....	- 37 -
VALIDACIÓN DEL ESTÁNDAR EPCGLOBAL CLASS-1 GEN-2 .....	- 37 -
4.1 Introducción a la validación .....	- 37 -
4.2 Herramientas de validación. La elección de SDL.....	- 37 -
4.3 El lenguaje SDL.....	- 38 -
4.4 Comunicación del sistema RFID .....	- 39 -
4.5 Diseño del Reader.....	- 41 -
4.5.1 Reader en escenario con trama estática .....	- 41 -
4.5.1.1 Se recibe señal Identificación.....	- 42 -
4.5.1.2 Se recibe señal T .....	- 43 -
4.5.1.2.1 No hay identificaciones correctas.....	- 43 -
4.5.1.2.2 Hay una identificación correcta .....	- 43 -
4.5.1.2.3 Hay más de una identificación correcta .....	- 45 -
4.5.1.2.3.1 Se recibe señal Identificación.....	- 47 -
4.5.1.2.3.2 Se recibe señal T .....	- 47 -
4.5.1.2.3.2.1 No hay identificaciones correctas y no se ha alcanzado el número máximo de slots en el ciclo .....	- 47 -
4.5.1.2.3.2.2 Hay una identificación correcta y no se ha alcanzado el número máximo de slots en el ciclo.....	- 47 -
4.5.1.2.3.2.3 Hay una o ninguna identificación y el número de slots es mayor que el posible en este ciclo. ....	- 50 -
4.5.1.2.3.2.4 Hay varias identificaciones y no se ha alcanzado el número máximo de slots posibles en este ciclo .....	- 50 -
4.5.1.2.3.2.5 Hay varias identificaciones y el número de slots es mayor que el posible en este ciclo .....	- 51 -
4.5.2 Reader en escenario con trama adaptativa ciclo a ciclo.....	- 52 -
4.5.2.2 Se recibe una señal T .....	- 54 -
4.5.2.2.1 No hay identificaciones correctas.....	- 54 -
4.5.2.2.2 Hay una identificación correcta .....	- 54 -
4.5.2.2.3 Hay más de una identificación correcta .....	- 55 -
4.5.2.2.3.1 Identificación .....	- 56 -
4.5.2.2.3.2 Temporizador .....	- 56 -



4.5.2.2.3.2.1 No hay identificaciones correctas y no se ha alcanzado el número máximo de slots en el ciclo .....	- 56 -
4.5.2.2.3.2.2 Hay una identificación correcta y no se ha alcanzado el número máximo de slots en el ciclo.....	- 56 -
4.5.2.2.3.2.3 Hay una o ninguna identificación y el número de slots es mayor que el posible en este ciclo .....	- 56 -
4.5.2.2.3.2.4 Hay varias identificaciones y no se ha alcanzado el número máximo de slots posibles en este ciclo .....	- 59 -
4.5.2.2.3.2.5 Hay varias identificaciones y el número de slots es mayor que el posible en este ciclo .....	- 59 -
4.5.3 Reader en escenario con trama adaptativa slot a slot .....	- 60 -
4.5.3.1 Se recibe señal Identificación.....	- 61 -
4.5.3.2 Se recibe una señal T .....	- 62 -
4.5.3.2.1 No hay identificaciones correctas.....	- 62 -
4.5.3.2.2 Hay una identificación correcta .....	- 62 -
4.5.3.2.3 Hay más de una identificación correcta .....	- 63 -
4.5.3.2.3.1 Identificación .....	- 64 -
4.5.3.2.3.2 Temporizador .....	- 64 -
4.5.3.2.3.2.1 No hay identificaciones correctas y no se ha alcanzado el número máximo de slots en el ciclo .....	- 64 -
4.5.3.2.3.2.2 Hay una identificación correcta y no se ha alcanzado el número máximo de slots en el ciclo.....	- 64 -
4.5.3.2.3.2.3 Hay una o ninguna identificación y el número de slots es mayor que el posible en este ciclo .....	- 64 -
4.5.3.2.3.2.4 Hay varias identificaciones y no se ha alcanzado el número máximo de slots posibles en este ciclo .....	- 66 -
4.5.3.2.3.2.5 Hay varias identificaciones y el número de slots es mayor que el posible en este ciclo .....	- 66 -
4.6 Diseño del Tag .....	- 67 -
4.6.1 Se recibe ACK .....	- 68 -
4.6.1.1 Se recibe NACK .....	- 69 -
4.6.1.2 Se recibe QueryRep.....	- 69 -
4.6.2 Se recibe Query .....	- 70 -
4.6.2.1 QueryRep .....	- 71 -
4.6.2.2 ACK.....	- 71 -
4.6.2.2.1 Confirmación de datos. QueryRep .....	- 71 -
4.6.2.2.1 Confirmación de datos. NACK .....	- 71 -
4.6.2.3 QueryAdjust .....	- 72 -
4.6.2.4 NACK .....	- 72 -
4.7 Puesta en marcha del Validador / Simulador .....	- 74 -
Capítulo 5 .....	- 79 -

Resultados Obtenidos .....	- 79 -
5.1 Parámetros establecidos.....	- 79 -
5.2 Resultados obtenidos de trama estática.....	- 79 -
5.3 Resultados obtenidos de trama adaptativa ciclo a ciclo.....	- 82 -
5.4 Resultados obtenidos de trama adaptativa slot a slot.....	- 84 -
Capítulo 6 .....	- 87 -
Conclusiones y líneas futuras .....	- 87 -
6.1 Conclusiones.....	- 87 -
6.2 Líneas futuras.....	- 89 -
ANEXO I. ESTÁNDARES ISO .....	- 91 -

# INDICE DE FIGURAS

---

## Capítulo 2

Figura 2.1: Tag RFID .....	18 -
Figura 2.2: Esquema de funcionamiento de RFID.....	19 -
Figura 2.3: Espectro electromagnético.....	20 -

## Capítulo 3

Figura 3.2: Formato de paquete EPC.....	31 -
Figura 3.3: Algoritmo de ajuste de la Q.....	34 -

## Capítulo 4

Figura 4.1: Esquema organizador del sistema RFID.....	39 -
Figura 4.2: Comunicación entre reader y tags.....	40 -
Figura 4.3: Inicio del proceso (estático).....	41 -
Figura 4.4: Procedimiento Emitir_Broadcast_estatico (estático).....	42 -
Figura 4.5: Procedimiento Recibe_ID_estatico (estático).....	43 -
Figura 4.6: Una o ninguna identificación correcta (estático).....	44 -
Figura 4.7: Procedimiento RecibeDatosID_estático (estático).....	45 -
Figura 4.8: Varias identificaciones correctas (estático).....	46 -
Figura 4.9: Función Elevar_estático (estático).....	47 -
Figura 4.10: Esquema para una o ninguna identificación correcta y con número de slots dentro del rango permitido en el ciclo (estático).....	48 -
Figura 4.11a: Procedimiento RecibeDatosCI_estático (estático).....	49 -
Figura 4.11b: Procedimiento RecibeDatosCI_estático. Continuación (estático).....	50 -
Figura 4.12: Nuevo Ciclo (estático).....	51 -
Figura 4.13: Esquema para varias identificaciones correctas y con número de slots dentro del rango permitido en el ciclo (estático).....	52 -
Figura 4.14: Inicio del proceso Reader_ciclo (ciclo a ciclo).....	53 -
Figura 4.15: Una o ninguna identificación correcta (ciclo a ciclo).....	54 -
Figura 4.16: Varias identificaciones correctas (ciclo a ciclo).....	55 -
Figura 4.17: Esquema para una o ninguna identificación correcta y con número de slots dentro del rango permitido en el ciclo (ciclo a ciclo).....	57 -
Figura 4.18: Algoritmo de Ajuste de la Q (ciclo a ciclo).....	58 -
Figura 4.19: Nuevo ciclo (ciclo a ciclo).....	59 -
Figura 4.20: Esquema para varias identificaciones correctas y con número de slots dentro del rango permitido en el ciclo (ciclo a ciclo).....	60 -
Figura 4.21: Inicio del proceso Reader (slot a slot).....	61 -
Figura 4.22: Una o ninguna identificación correcta (slot a slot).....	62 -
Figura 4.23: Varias identificaciones correctas (slot a slot).....	63 -

Figura 4.24: Esquema para una o ninguna identificación correcta y con número de slots dentro del rango permitido en el ciclo (slot a slot) ...	- 65 -
Figura 4.25: Nuevo Ciclo (slot a slot).....	- 66 -
Figura 4.26: Esquema para varias identificaciones correctas y con número de slots dentro del rango permitido en el ciclo (slot a slot).....	- 67 -
Figura 4.27: Inicio proceso Tag.....	- 68 -
Figura 4.28: Confirmación de Identificación.....	- 69 -
Figura 4.29: Confirmación de datos.....	- 70 -
Figura 4.30: Procedimiento Comprobar_r.....	- 71 -
Figura 4.31a: Espera confirmación de identificación dentro del Ciclo de Identificación.....	- 72 -
Figura 4.31b: Espera confirmación de identificación dentro del Ciclo de Identificación.....	- 73 -
Figura 4.32: Espera confirmación de datos dentro del Ciclo de Identificación.....	- 74 -
Figura 4.33: Ejemplo de trazas con MSC.....	- 75 -
Figura 4.34: Introducción manual de sinónimos.....	- 76 -
Figura 4.35: Introducción de sinónimos a través de ficheros.....	- 76 -

## Capítulo 5

Figura 5.1: Distribución del Numero Total de Slots en trama estática.....	- 81 -
Figura 5.2: Distribución del Número de Ciclos en trama estática.....	- 81 -
Figura 5.3: Distribución del Numero Total de Slots en trama adaptativa cada ciclo.....	- 83 -
Figura 5.4: Distribución del Número de Ciclos en trama adaptativa ciclo a ciclo.....	- 84 -
Figura 5.5: Distribución del Numero Total de Slots en trama adaptativa slot a slot.....	- 85 -
Figura 5.6: Distribución del Número de Ciclos en trama adaptativa slot a slot.....	- 86 -

## Capítulo 6

Figura 6.1: Número Total de Slots para una probabilidad PEP=0.....	- 88 -
Figura 6.2: Número Total de Slots para una probabilidad PEP=10e-6.....	- 88 -
Figura 6.3: Número Total de Slots para una probabilidad PEP=10e-3.....	- 88 -

# INDICE DE TABLAS

---

## Capítulo 2

Tabla 2.1: Comparativa de tecnologías de identificación de objeto.....	- 21 -
Tabla 2.2: Comparativa entre tecnologías.....	- 22 -
Tabla 2.3: Desarrollo de estándares.....	- 23 -
Tabla 2.4: Clasificación de tags según EPCGlobal.....	- 24 -

## Capítulo 4

Tabla 4.1 : Simbología en SDL.....	- 38 -
Tabla 4.2 : Señales entre los procesos Reader y Tag.....	- 40 -

## Capítulo 5

Tabla 5.1: Resultados de Q estática con PEP=0.....	- 79 -
Tabla 5.2 : Resultados Q estática con PEP=10e-6.....	- 80 -
Tabla 5.3 : Resultados Q estática con PEP=10e-3.....	- 80 -
Tabla 5.4: Resultados Q adaptativa ciclo a ciclo con PEP=0.....	- 82 -
Tabla 5.5: Resultados Q adaptativa ciclo a ciclo con PEP=10e-6.....	- 82 -
Tabla 5.6: Resultados Q adaptativa ciclo a ciclo con PEP=10e-3.....	- 82 -
Tabla 5.7: Resultados Q adaptativa slot a slot con PEP=0.....	- 84 -
Tabla 5.8: Resultados Q adaptativa slot a slot con PEP=10e-6.....	- 84 -
Tabla 5.9: Resultados Q adaptativa slot a slot con PEP=10e-3.....	- 85 -



# CAPÍTULO 1

## INTRODUCCIÓN

---

### 1.1 Antecedentes

Existen múltiples tecnologías basadas en la detección e identificación de objetos de forma remota. Sin embargo, la más empleada y más conocida es el código de barras. El código de barras se inventó hace más de 25 años y, durante este tiempo, ha sido la tecnología más utilizada por los comercios y la industria para identificar sus productos e ítems, aún presentando numerosas limitaciones: corto alcance, baja seguridad datos, requisito de visión directa entre lector y código de barras, etc.

La tecnología de identificación por radiofrecuencia, RFID (*Radio Frequency Identification*) es otra de las tecnologías basadas en detección e identificación de ítems. No es una tecnología nueva, de hecho, su origen se remonta a la Segunda Guerra Mundial. Sin embargo, en aquellos tiempos, la inmadurez de la tecnología y el alto coste de los dispositivos que hacía posible la identificación de los ítems hicieron que RFID dejara de ser una tecnología atractiva para su difusión, estandarización y utilización por parte de empresas y particulares. Actualmente, los costes de los dispositivos RFID se han reducido hasta tal punto que RFID se enfoca hacia una futura identificación unitaria, siendo por tanto el futuro sustituto de la actual tecnología de código de barras.

Para asegurar la adopción global y multisectorial de RFID, en el año 2003 se creó la institución EPCglobal. Esta institución tiene como fin administrar todos los estándares industriales para EPC (*Electronic Product Code*) y realizar la comercialización y difusión de ellos. De los estándares manejados por EPCglobal, EPCglobal Class-1 Gen-2 es el estándar mayoritario, ampliamente adoptado por fabricantes y distribuidores. Este estándar está diseñado para cubrir las necesidades de los sistemas RFID bajo las frecuencias UHF (860 MHz-930MHz). Incluye las especificaciones hardware de las etiquetas y los lectores y las especificaciones software, donde se incluye el protocolo anti-colisión, implementado en el lector y en las etiquetas.

Aunque los fabricantes de sistemas RFID pueden ver el estándar EPCglobal Class-1 Gen-2 como una guía para la fabricación y puesta en marcha de los dispositivos RFID, existe en dicho estándar cierta libertad para que el fabricante tome sus propias decisiones de diseño en el hardware y software, siendo, este último el quebradero de cabeza de la comunidad científica.

La libertad de decisión en el protocolo anti-colisión propuesto por el estándar solo produce dudas y desconocimiento ya que, dependiendo de los parámetros y las asunciones que haga cada fabricante, el protocolo anti-colisión, y por tanto, el sistema RFID, puede presentar un mejor o peor rendimiento en cuanto al tiempo de identificación, número máximo de etiquetas que se pueden identificar por unidad de tiempo, etc.

### 1.2 Objetivos del proyecto

El objetivo del presente proyecto es la validación o comprobación del protocolo de acceso al medio EPCGlobal Class-1 Gen-2 para distintos parámetros y asunciones del protocolo anti-colisión que implementa con el fin de caracterizar el rendimiento del

estándar ante diversos parámetros de configuración. Este estudio forma parte del trabajo realizado por el grupo de investigación del área de ingeniería Telemática de la UPCT.

Para llevar a cabo este trabajo se va a implementar dicho protocolo según el estándar Gen-2 en el lenguaje de especificación SDL. Se establecerán 3 versiones del protocolo distintas, que corresponden a las diversas asunciones que un fabricante puede realizar y establecer en su sistema RFID: protocolo trama estática, protocolo trama dinámica ciclo a ciclo y protocolo trama dinámica slot a slot. Para cada uno de los protocolos propuestos se realizarán distintas pruebas de validación y simulación. Para finalizar, se obtendrá un conjunto de resultados que servirán para comprobar el funcionamiento del estándar ante las diversas posibilidades de configuración y adaptación del protocolo propuesto por el estándar. Además, dichos resultados de validación servirán para comprobar la coherencia de los resultados obtenidos con otros resultados de simulación obtenidos por el grupo de investigación en un trabajo previo. También se comprobará la efectividad de cada uno de los escenarios validados y simulados, que pueden servir como guía a los fabricantes y los instaladores de sistemas RFID para obtener un mejor rendimiento y unas mayores prestaciones de los sistemas RFID. Por último se estudiarán las posibles alternativas para la obtención de futuros resultados que puedan ser de interés.

## 1.3 Estructura del contenido

El proyecto se compone de seis capítulos, siendo el primero una breve introducción al esquema seguido para su desarrollo. El resto, está organizado en los siguientes capítulos.

El capítulo 2 ofrece una visión de la tecnología RFID, conociendo su evolución, componentes y estándares que la regulan. Además, describe los protocolos de identificación/anticolisión en sistemas RFID, incluyendo una breve descripción del protocolo EPCGlobal Class-1 Gen-2.

En el capítulo 3, se estudia con detenimiento el estándar EPCGlobal Class-1 Gen-2, asimilando conceptos relacionados tanto con los requisitos hardware de cada uno de los componentes del sistema RFID como la implementación software. Se analizará el protocolo de comunicación entre el reader y los tags y las distintas implementaciones del protocolo anticolisión que se pueden adoptar en un lector: trama estática, y trama adaptativa en sus dos versiones (ciclo a ciclo y slot a slot).

El capítulo 4 trata sobre la validación del estándar, objeto de estudio del proyecto. Se define en que consiste validar un protocolo, así como las herramientas necesarias y la justificación de la elección de SDL para llevarlo a cabo. A continuación se da a conocer la implementación software tanto del reader, como de los tags y de la comunicación entre ambos.

En el capítulo 5 se muestran los resultados de validación y simulación realizados para cada uno de los escenarios configurados, estableciendo diversos parámetros relacionados con el entorno simulado, el protocolo anticolisión establecido, o la población de tags a identificar.

Para finalizar, se reflexionan las conclusiones obtenidas y argumentamos las posibles líneas de trabajo futuras.



# CAPÍTULO 2

## LA TECNOLOGÍA RFID

---

### 2.1 Introducción a RFID

RFID (*Radio Frequency IDentification*, o Identificación por Radiofrecuencia) es una tecnología que permite transmitir la identidad de un objeto (similar a un número de serie único, ID) mediante ondas de radio con el fin de identificar dicho objeto. Un sistema RFID es un sistema de almacenamiento y recuperación de datos remoto compuesto por una antena lectora, llamada *Reader* y una serie de dispositivos denominados etiquetas, transpondedores o tags RFID.

#### 2.1.1 Origen y evolución

El origen de la tecnología RFID se remonta a la II Guerra Mundial, época en la que el uso del radar permitía la detección de aviones a kilómetros de distancia, pero no su identificación. Los sistemas de radar y de comunicaciones por radiofrecuencia avanzaron en las décadas de los 50 y 60 en las que los científicos de los países más avanzados trabajaban para explicar cómo identificar objetos remotamente. Las primeras patentes para dispositivos RFID fueron solicitadas en EEUU, concretamente en Enero de 1973 cuando Mario W. Cardullo se presentó con un tag RFID activo que contenía una memoria reescribible. El mismo año, Charles Walton recibió la patente para un sistema RFID pasivo que abría las puertas sin necesidad de llaves, una tarjeta con un tag comunicaba una señal al lector de la puerta, que cuando validaba la tarjeta, desbloqueaba la cerradura. El principal responsable del desarrollo e implantación de esta tecnología es Auto-ID Center, una sociedad constituida en 1999 por un centenar de empresas punteras, universidades y centros de investigación de todo el mundo. Hace unos años desarrollaron lo que hoy conocemos como la Red EPC™ y sus componentes. El AutoID Center, ahora conocido como AUTOID Labs., está formado por 6 laboratorios localizados en universidades de prestigio como el MIT (*Massachussets Institute of Technology*) de EEUU, Universidad de Cambridge en el Reino Unido, Universidad de Adelaide en Australia, Keio University en Japón, Universidad Fudan en China y Universidad de St. Gallen en Suiza.

A lo largo de su evolución, han sido numerosas las mejoras en la capacidad de emisión y recepción, así como en la distancia, lo cual ha llevado a extender su uso al ámbito tanto doméstico como de seguridad nacional. Empleando esta tecnología se puede identificar personas, animales y todo tipo de productos de consumo (similar al tradicional código de barras) y rastrear a una determinada distancia para proporcionar cierta información. Varios estudios han revelado la tendencia al fuerte crecimiento de esta tecnología gracias a las ventajas que presenta frente a factores desincentivadores, como el coste, el desconocimiento de la tecnología y la necesidad de seguridad en los niveles de acceso que aseguren los derechos de privacidad de los consumidores.

#### 2.1.2 Elementos de un sistema RFID

Un sistema RFID está compuesto por cientos o miles de tags, los cuales se identifican enviando su información a uno o varios dispositivos lectores, llamados readers. Los readers están conectados a su vez, con un subsistema de tratamiento de

datos (LAN, WAN, etc....) de forma inalámbrica o mediante cableado. Cada uno de éstos dispositivos se explican con más detalle a continuación:

### 2.1.2.1 Tag RFID

Este dispositivo contiene una antena cuyo propósito es permitir al chip la transmisión de información de identificación del tag que contiene en su memoria interna (capacidad que depende del modelo y varía de una decena a millares de bytes). Podemos clasificar estos dispositivos atendiendo a dos criterios diferentes, por un lado, según el tipo de memoria y por otro, dependiendo de la fuente de alimentación. Según el tipo de memoria, los tags se pueden clasificar en:

- Tags solo lectura: El código de identificación que contiene es único y no puede reescribirse. Normalmente se establece durante la fabricación del tag.
- Tags de múltiple lectura y una única escritura: La información de identificación puede ser modificada por el lector una sola vez. Estos tags vienen sin información a la salida de fábrica y es el usuario del sistema RFID quien escribe la información de interés en el tag mediante un lector.
- Tags de múltiple lectura, múltiple escritura. La memoria de los tags puede ser leída y escrita múltiples veces.

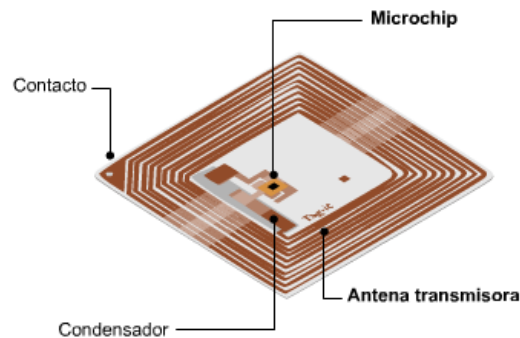


Figura 2.1: Tag RFID

Por otro lado, si nos basamos en la fuente de alimentación utilizada, los tags se pueden clasificar principalmente en:

- Pasivos: Este tipo de tags no tienen fuente de alimentación integrada, utilizan la energía emitida por el lector para autoalimentarse y transmitir su información almacenada al reader. La comunicación la inicia siempre el lector, con lo que la presencia de este es imprescindible para que el tag transmita sus datos. Como ventaja, destacar que son simples y baratos de fabricar, además, al no tener partes móviles, tienen una mayor vida. Soportan condiciones extremas como corrosivos o altas temperaturas. Sin embargo, el rango de alcance es inferior al del resto [10mm – 6m].
- Semipasivos: Incluyen una pequeña batería que permite que el circuito integrado esté constantemente alimentado. Emplean una batería para activar sus circuitos y, a partir de ese momento, emplean la energía procedente del lector para funcionar. Responden más rápido y el radio de lectura es más grande que el de los tags pasivos. Tienen una fiabilidad comparable a la de los tags activos a la vez que pueden mantener el rango operativo de un tag pasivo. Suelen durar más que los tags activos.

- **Activos:** Contienen una fuente de alimentación incorporada (una batería o panel solar). El tag activo utiliza la energía de su batería para enviar la señal al reader, con lo que no necesita que éste envíe la onda continua para energizar la antena. El tag es siempre el primero en comunicarse, seguido de la respuesta/consulta del reader. Es importante destacar su capacidad para almacenar información y una duración de batería de hasta varios años. Además, cuenta con factores como exactitud, funcionamiento en ambiente cercano al agua o metal y un alto nivel de fiabilidad, con rangos de aproximadamente 10 m.

### 2.1.2.2 Lector de RFID o Reader

El reader se compone de una circuitería más compleja que la de un tag. Está compuesto por una o varias antenas que emiten señales electromagnéticas creando un área de cobertura para capturar los datos de los tags que entran en esa área. Cuando el lector captura los datos de los tags, estos los envía a un subsistema de procesamiento de datos para tratarlos según las necesidades de la aplicación final.

### 2.1.3 Funcionamiento básico de un sistema RFID

Un sistema RFID se comporta de la siguiente forma: un reader envía una señal de interrogación a un conjunto de tags y estos responden enviando cada uno su número único de identificación. Una vez el reader ha recibido el código único del producto, lo transmite a una base de datos, donde se han almacenado previamente las características del artículo en cuestión: fecha de caducidad, material, peso, dimensiones, localización, etc. De este modo se hace posible consultar la identidad de algo o alguien en cualquier momento, ya sea el caso de una aplicación a un producto o a una persona.

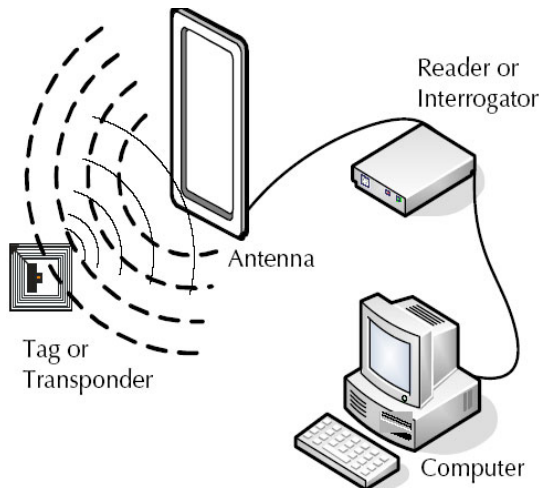


Figura 2.2: Esquema de funcionamiento de RFID

### 2.1.4 Rangos de frecuencia.

Los sistemas RFID pueden trabajar a distintos rangos de frecuencia. Dichos rangos influyen directamente en el alcance de los dispositivos y en los materiales con los que un tag puede trabajar.

- **Sistemas de frecuencia baja (LF):** Comprende las frecuencias emitidas en un rango de 125 KHz y 134 KHz, los tags LF se caracterizan por la baja velocidad y las cortas distancias de lectura y por la identificación unitaria (no existe multilectura).

- Sistemas de alta frecuencia (HF): Comunicación a 13.56 MHz, empleando tags con lectura a velocidad media y distancia inferior a 2m.
- Sistemas de Ultra alta frecuencia (UHF): Es la banda comprendida entre 860 y 960 MHz. Se utiliza en aplicaciones donde se deben leer masivamente varios tags a alta velocidad, además permiten lecturas a grandes distancias. Se utilizan tags de tamaño mediano y multilectura.
- Sistemas de microondas (~2.45GHz): Ofrecen grandes alcances (por encima de los 30 metros).

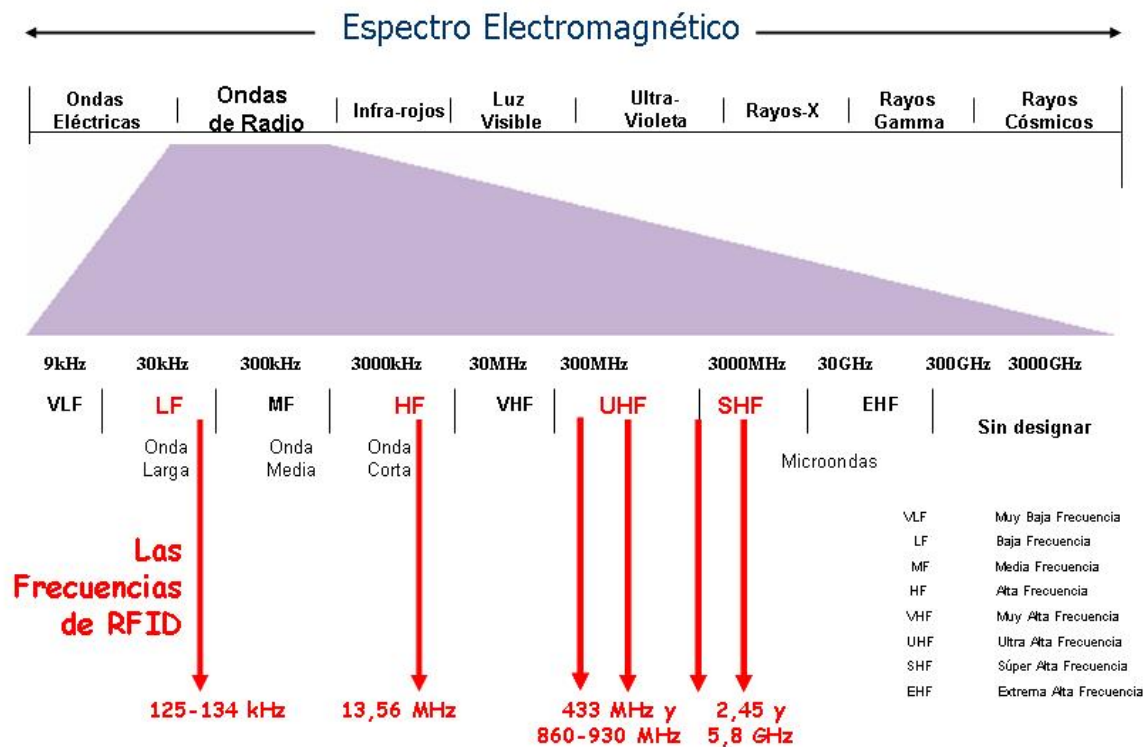


Figura 2.3: Espectro electromagnético

## 2.1.5 RFID vs otras tecnologías

La tecnología RFID es muy versátil y puede ser aplicada en diferentes sectores. El uso de soluciones basadas en RFID permite la captura de datos en tiempo real de manera ágil, lo que es sinónimo de enormes ventajas donde destacan:

- Procesos de captura de información automatizados, reduciendo los costes derivados de errores en la introducción manual de datos.
- Información disponible en tiempo real, acelerando la toma de decisiones y facilitando la reacción ante los distintos escenarios de negocio.
- Escalabilidad del sistema.

Es posible estudiar RFID en comparación con otras tecnologías desde dos puntos de vista muy distintos. En primer lugar (Tabla 1) se trata desde el punto de vista de las distintas alternativas de identificación de objetos: Códigos de barras, memorias de contacto y los dos tipos principales de sistemas de identificación por radiofrecuencia (activos y pasivos).

Actualmente, los costes de los dispositivos RFID se han reducido hasta tal punto que RFID se enfoca hacia una futura identificación unitaria, siendo por tanto el futuro sustituto de la actual tecnología de código de barras. RFID destaca frente al código de barras por muchas e interesantes características:

- A diferencia del código de barras, las etiquetas de tecnología RFID identifican los objetos y no necesitan estar en visión directa con el lector que las identifica. Además, dependiendo de la frecuencia utilizada, las lecturas pueden producirse a distancias de hasta 100 metros.
- El código de barras identifica un tipo de producto de un fabricante (p. ej. Leche Pascual de 1 litro), mientras que los tags identifican cada producto de forma individual (p. ej. Ítem 1423 de Leche Pascual, partida 427, peso 1 litro, caducidad 20-Septiembre-2008, proveedor: Pascual S.A, etc.).
- La tecnología RFID permite leer múltiples tags de forma simultánea mediante protocolos anti-colisión eficientes implementados en los lectores. Por el contrario, los códigos de barras se deben leer individualmente, de forma secuencial. Esta característica del sistema de identificación por radiofrecuencia ofrece numerosas ventajas, entre las que destaca el ahorro en tiempo de identificación.
- Los tags pueden almacenar mucha más información sobre un producto que el código de barras, ya que poseen una memoria donde se pueden almacenar hasta decenas de kbytes.
- El código de barras es un elemento de solo lectura mientras que los tags permiten múltiple lectura y escritura de datos en su memoria.
- La tecnología RFID evita falsificaciones ya que la información se almacena encriptada en la memoria de los tags. Sin embargo, un código de barras es vulnerable a todo tipo de falsificaciones, ya que con una simple fotocopiadora es posible reproducir un código de barras.
- Un código de barras se estropea o se rompe fácilmente, mientras que un tag es más resistente porque, normalmente, forma parte del producto o se coloca bajo una superficie protectora y soporta mejor la humedad y las altas y bajas temperaturas.

	Código de Barras	Memorias de Contacto	RFID Pasivo	RFID Activo
Modificación de datos	No modificable	Modificable	Modificable	Modificable
Seguridad de datos	Seguridad mínima	Altamente seguro	Rango de baja a alta seguridad	Alta seguridad
Almacenamiento	Lineal: 8-30 caracteres 2D: 7200 dígitos	De 8Mb en adelante	Alrededor de 64Kbytes	Alrededor de 8Mbytes
Coste	Bajo	Alto	Medio	Muy alto
Estándares	Estable e implantado	Propietario, sin estándar	Estándares en fase de implantación	Propietario y estándares abiertos
Tiempo de vida	Bajo por deterioro	Largo	Indefinido	3-5 años de vida de batería
Distancia de lectura	Pocos centímetros	Contacto necesario	Del orden de 1m	Del orden de 100m
Interfaz	Lectura óptica directa	Contacto	Sin barreras aunque con interferencias	Sin barreras aunque con interferencias

Tabla 2.1: Comparativa de tecnologías de identificación de objeto

La Tabla 2 presenta otro enfoque. Se puede tratar RFID como una de las tecnologías de comunicación entre sistemas computacionales y las alternativas ya maduras con las que podría competir serían Bluetooth y WiFi. Sin embargo, en muchas aplicaciones, la convivencia entre RFID y otras tecnologías de comunicación inalámbrica es posible y deseable.

Tecnología	Bluetooth	WIFI	RFID
Transmisión	Voz y datos	Voz y datos	Datos (código EPC)
Equipos	Hasta 8 equipos (Piconets)	Equipos con configuración compatible	Miles de etiquetas RFID con 1 o varios lectores
Comunicación	Síncronos, Bidireccional y asíncronos	Varios estándares IEEE 802.11	Síncrono y Bidireccional Varios estándares
Velocidad	Síncronos: 432 Kbps y Bidireccional: 721 kbps en un sentido y 57.6 en el otro	Según el estándar	
Encriptación		WEP, WPA	Código EPC
Utilización	Señales de radiofrecuencia	Señales de radiofrecuencia	Señales de radiofrecuencia
Distancia	10 m a 100 m	30 m	Pasivas: 10 mm a 6m Activas: varios km
Red	Inalámbrica	Inalámbrica conocida como 802.11	Transmisión por campos electromagnéticos e identificación

Tabla 2.2: Comparativa entre tecnologías

## 2.1.5 Aplicaciones de RFID

Las aplicaciones de esta tecnología son muy diversas en función de la frecuencia utilizada, ya que ésta afecta directamente al alcance y al coste de la misma. Las etiquetas de baja frecuencia suelen emplearse por ejemplo, para la identificación de animales, como llave para automóviles, para el control de acceso a edificios, seguimiento de camiones, en el cobro de peajes, en la elaboración de la cuenta de la compra... Además, se aplica para registrar la información de trazabilidad de algunos productos, ya que permite registrar información que un código de barras convencional no puede contener. Algunas de las aplicaciones más importantes de la tecnología RFID son las siguientes:

- **Inventariado:** Es la principal aplicación de RFID en la actualidad. Consiste en asignar un tag a los artículos, reemplazando el actual código de barras. Tener información adicional de los objetos ofrece numerosos beneficios, tanto para el control de stock como en la gestión de distribución y abastecimiento.
- **Facturación:** Los tags proporcionan un código de identificación única para cada maleta y se utiliza para dirigir el equipaje de manera más rápida y eliminando gran parte de riesgo de pérdidas. Además, la posibilidad de actualización automática de la información de las etiquetas permite una mejor gestión ante cambios de última hora en los vuelos o de las conexiones.
- **Seguridad:** La tecnología RFID puede complementar los sistemas actuales de seguridad, tanto es así, que ya se están comercializando sistemas de seguridad de acceso basados en tarjetas RFID que además, incluyen un código secreto para evitar que la tarjeta sea usada por una persona incorrecta.
- **Información estaciones autobuses:** Cada plataforma dispone de una antena y cada autobús de una tarjeta RFID. Se informa a los pasajeros en tiempo real y

de forma totalmente automática de las llegadas y salidas mediante displays. Se consigue mayor versatilidad y capacidad de reacción a cambios sin que los usuarios se vean afectados.

- Identificación de animales: Se emplea un microchip en una cápsula de cristal que se introduce bajo la piel del animal. Sirve para identificar unívocamente al animal y almacenar información relevante (por ejemplo, quién es el dueño).
- Implantes humanos: Los chips RFID implantables presentan una solución a la usurpación de la identidad, el acceso seguro a un edificio, a un ordenador, al almacenamiento de expedientes médicos, a iniciativas antisequestro y una gran variedad de aplicaciones.

## 2.1.6 Estándares actuales

RFID es una tecnología heterogénea con un significativo número de estándares asociados. Los estándares más relevantes describen las capas física y de enlace de datos (interfaz aéreo, anticollisión, protocolos de comunicación y funciones de seguridad). Adicionalmente los estándares RFID describen métodos de prueba y estándares de aplicación de datos (formato de identificación única, protocolo de datos e interfaces de programación de aplicación). La estandarización de RFID se debe a dos organizaciones, por un lado, a la Organización Internacional para la Estandarización, ISO, y por otro, a la responsable del desarrollo del Electronic Product Code (EPC), la organización Auto-ID Center que desarrolló el estándar EPCGlobal. A continuación se muestra una relación entre las bandas de frecuencias y los distintos estándares, detallando la región de operación:

Banda de frecuencias	Nombre del estándar	Detalles
Menos de 135 KHz	ISO 18000-2	
6,765 ~ 6,795 MHz		
7,4 ~ 8,8 MHz		
13,55 ~ 13,57 MHz	ISO 18000-3	Uso común
26,96 ~ 27,78 MHz		Aplicaciones especiales
433MHz	ISO 18000-7	Tags activos en Asia
868 ~ 870 MHz	ISO 18000-6 A/B	Europa
902 ~ 928 MHz	AutoID clase 0/1	Norteamérica
860 ~ 960 MHz	EPCGlobal Gen2	Mundial
2400 ~ 2483 MHz	ISO 18000-4	
8725 ~ 5785 MHz	ISO 18000-5	Usado raramente en RFID

Tabla 2.3: Desarrollo de estándares

### Estándar ISO

ISO ha desarrollado estándares de RFID para la identificación automática y la gestión de objetos. La serie de estándares de ISO relacionada con las RFID y las frecuencias que se emplean en dichos sistemas es la serie 18000. En el Anexo I se incluye la lista completa de estándares ISO.



### Estándar EPCGlobal

EPC es administrado a nivel mundial por el EPCglobal. El proceso de desarrollo de estándares EPCglobal es un proceso en evolución y continúa siendo actualizado de acuerdo a las necesidades de los usuarios. Los estándares desarrollados por EPCglobal son elementos fundamentales para la continuidad de la Red EPCglobal que combina tecnología RFID, la actual infraestructura de red y el EPC para proveer información precisa y óptima de los artículos. Está compuesto por:

- Código Electrónico del Producto: número de identificación cuyo esquema se detalla en el apartado 3.2.1 del capítulo 3.
- Sistema de identificación: En la Tabla 4 se muestra la clasificación de los diferentes tipos de tags por EPCGlobal.

Clase	Descripción
Class 0	Pasivo, solo lectura
Class 0+	Pasivo, escritura única usando el protocolo Class 0
Class I	Pasivo, solo lectura con memoria programable no volátil
Class II	Pasivo, lectura / escritura de hasta 65 KB
Class III	Reescribible, semi-pasivo (funciona con batería, comunicación impulsada por el lector), sensores integrados.
Class IV	Reescribible, activo, "bidireccionales", pueden comunicarse con otros tags, impulsando sus propias comunicaciones.
Class V	Activo, puede comunicarse con tags de la Class IV y otros dispositivos

Tabla 2.4: Clasificación de tags según EPCGlobal

- Middleware: Para una administración de acontecimientos en tiempo real [Auto-ID Center 2002c].
- Servicio de Nombramiento de Objeto (ONS): servicio de conexión de red similar al Servicio de Nombre de Dominio (DNS) [Auto-ID Center 2002b].
- El lenguaje de marcado físico (PML): es basado en XML y provee de un Standard de representación de información desde la red EPC [Brock 2001, Auto-ID Center 2003c].
- Servicios de información EPC.
- Arquitectura de referencia de la red EPCGlobal.

## 2.2 Protocolos de identificación / anticolisión en sistemas RFID

Tanto en los sistemas con tags pasivos, como en los sistemas de tags activos surge el problema de las colisiones en el acceso al medio. Dicho conflicto aparece cuando varios tags intentan identificarse en el mismo instante temporal y por tanto, se produce una colisión. Surge la necesidad de un mecanismo que asegure la identificación correcta de todos los tags en cobertura. Para los tags activos, y debido a la confusión existente entorno a esta tecnología, los protocolos anticolisión se estudian desde otro enfoque. El estándar ISO 18000-7 propone usar FSA (*Frame Slotted Aloha*) como protocolo anticolisión, sugiriendo un mecanismo de adaptación de la longitud de la trama sin especificar uno en particular, por otro lado, algunos estudios proponen el



empleo del protocolo CSMA. Para los tags que nos competen en este trabajo, los tags pasivos, se dispone de los protocolos anticollisión pueden agruparse en dos grandes categorías, algoritmos deterministas o algoritmos probabilísticos.

Por último, la organización EPCGlobal ha fijado el estándar EPC “Gen 2”, como referencia. Es un protocolo probabilístico y se usa independientemente del tipo de dispositivo, activo o pasivo. En las siguientes secciones resumimos estos protocolos.

## 2.2.1 Protocolos deterministas

En los protocolos deterministas, el reader posee una lista con todos los números de serie únicos de cada uno de los tags que pueden identificarse. Cuando los tags están en cobertura, el reader interroga uno a uno hasta identificarlos a todos. Estos algoritmos también se llaman algoritmos de “búsqueda de árbol”. El mecanismo consiste en la descomposición en pequeños grupos del conjunto de tags a identificar mediante técnicas de segmentación o *Splitting*, hasta que el número de tags por subgrupo sea de uno. Para llevar a cabo este algoritmo, el reader transmite a todos los tags el primer bit de un número de serie aleatoriamente seleccionado. Cada tag comparará si el bit recibido coincide con el primer bit de su número de serie, de forma que, si coincide, transmitirá su identificador provocando una colisión entre todos los tags cuyo primer bit sea el mismo. Ahora tendremos 2 subgrupos de tags. En el siguiente ciclo, el reader transmitirá 2 bits, y así sucesivamente hasta que solo quede un tag cuyo número de serie coincida con los bits que envía el reader y se identifique correctamente. Finalmente se habrá creado un árbol de identificación. Este procedimiento es ineficiente cuando el número de tags a identificar es elevado.

## 2.2.2 Protocolos probabilísticos basados en FSA

Un algoritmo probabilístico, es aquel que basa su resultado en la toma de algunas decisiones al azar, de tal forma que, en promedio, obtiene una buena solución al problema planteado para cualquier distribución de los datos de entrada. Esta gran familia de protocolos se basa en el protocolo Aloha. Aloha fue la primera técnica de control de acceso para medios compartidos que fue mejorando sucesivamente hasta evolucionar al método de acceso CSMA/CD (*Carrier Sense Multiple Access with Collision Detection*) empleado actualmente en gran parte de redes. La mayoría de los sistemas RFID trabajan en entornos donde el reader no conoce previamente el número de tags que van a competir para identificarse. Por ello, se hace necesario implementar protocolos basados en Aloha.

### **Aloha**

La idea básica consiste en que, en el momento en el que el reader excita a los tags, éstos envían su identificación al canal independientemente de si ya había otro tag transmitiendo. Evidentemente, la transmisión de varias identificaciones se puede solapar en el tiempo y se produce entonces una colisión, impidiendo, en este caso, realizar una recepción correcta de ninguna de las identificaciones implicadas. En el Aloha original, la información que no es transmitida con éxito simplemente se pierde.

### **Aloha ranurado o SA (*Slotted Aloha*)**

Es una modificación del algoritmo Aloha original. En este nuevo método se emite periódicamente sobre el canal una referencia temporal que marca el inicio de una ranura o slot temporal. El periodo de esta señal es igual al tiempo requerido para transmitir una identificación sobre el canal. Ahora, si un tag quiere transmitir su identificación, se espera al inicio de la siguiente ranura temporal, de forma que solo podrá colisionar la información transmitida en el mismo slot, reduciendo así la ventana de colisión. En caso de colisión o de slot vacío, los tags esperarán un tiempo aleatorio para volver a intentarlo, empleando algún algoritmo para elegir dicho tiempo de

espera. Al contrario, si en un slot se identificó un único tag, el reader le indicará que se ha identificado correctamente y dejará de competir, evitando así que vuelva a participar en el proceso de identificación. Este método tan simple, no es eficiente para un elevado número de tags, ya que el número medio de slots que necesitan para identificarse, aumenta exponencialmente, por lo tanto, no será un algoritmo escalable y necesitaremos un mecanismo de adaptación si el número de tags aumenta considerablemente.

### **Aloha ranurado por trama FSA (*Frame Slotted Aloha*)**

Es una variante del Aloha ranurado, donde los slots están agrupados en tramas consecutivas. Cada trama equivale a un ciclo y los tags transmiten solo una vez en cada ciclo. Existen mecanismos de adaptación, con tamaño de trama dinámico (D-FSA), donde al finalizar un ciclo y en función de ciertos criterios, incrementa o decrementa el número de slots que compondrán el siguiente ciclo.

### **CSMA/CD (*Carrier Sense Multiple Access /Collision Detect*)**

CSMA/CD es la evolución del protocolo Aloha en el intento de resolver las bajas prestaciones de éste. En este algoritmo, los tags que quieren transmitir sus datos primero deben determinar si el medio está ocupado o no. En función de cómo actúen, se pueden clasificar en:

- CSMA no-persistente: si el canal está ocupado espera un tiempo aleatorio y vuelve a escuchar. Si detecta el canal libre, emite inmediatamente.
- CSMA 1-persistente: con el canal ocupado, el tag pasa a escuchar constantemente sin esperar tiempo alguno. En cuanto lo detecta libre, emite. Puede ocurrir que, si durante un retardo de propagación posterior a la emisión de la identificación emite otro tag, se produzca una colisión.
- CSMA p-persistente: después de encontrar el canal ocupado, y quedarse escuchando hasta encontrarlo libre, el tag decide si emite. Para ello ejecuta un algoritmo que dará orden de transmitir con una probabilidad  $p$ , o de permanecer a la espera (probabilidad  $(1-p)$ ). Si no transmitiera, en la siguiente ranura temporal volvería a ejecutar el mismo algoritmo. Así hasta transmitir. De esta forma se reduce el número de colisiones.

Una vez se ha comenzado a emitir, continuará hasta transmitir la trama completa. Si se produjera una colisión la transmisión será fallida. CSMA/CD supone una mejora, puesto que un tag escucha a la vez que emite, de forma que si detecta que se produce una colisión, para inmediatamente la transmisión y por lo tanto se produce una ganancia en tiempo. Además, el número de ciclos aumenta de forma prácticamente lineal con el número de tags, luego este algoritmo sí será escalable.

## **2.2.3 EPCGlobal Class-1 Gen-2**

EPCGlobal Class-1 Gen-2, define las especificaciones físicas y lógicas para sistemas RFID activos y pasivos que trabajan a una frecuencia de 860 MHz a 960 MHz, está basado en el protocolo Aloha Ranurado y se caracteriza por:

- Es simple y robusto, debido a que la complejidad del algoritmo de identificación recae sobre el reader.
- Puesto que los tags que se comercializan hoy en día cumplen los requisitos para su implementación, no supone un coste hardware añadido.
- Es ideal para sistemas RFID donde el reader no conoce a priori los tags que hay en su zona de cobertura.

- Mediante un sencillo algoritmo, el reader es capaz de adaptar el tamaño de cada ciclo, obteniendo mejores resultados en tiempo medio de identificación, utilización del canal y probabilidad de colisión.

En el capítulo 3 se detallará el funcionamiento de este estándar, así como otros aspectos relevantes para el desarrollo de este proyecto.



# CAPÍTULO 3

## Estándar

### EPCglobal Class-1 Gen-2

---

#### 3.1 Introducción

Como ya se comentó en el capítulo anterior, el estándar EPCglobal, cuya pieza fundamental es el código EPC, se encarga de impulsar la adopción e implementación mundial de la red EPCglobal a lo largo de todos los sectores.

El estándar EPCglobal Class-1 Gen-2 detalla los parámetros que deben tener los dispositivos que envían y reciben datos desde tags UHF. Especifica también cómo debe ser el uso de los canales y frecuencias, banda ancha y otros aspectos técnicos. Este estándar permite que todos los participantes en una misma cadena de suministro codifiquen sus tags de la misma manera para poder compartir la infraestructura tecnológica e interoperar en ella.

#### 3.2 Requisitos Hardware

En los siguientes puntos se detallan cada uno de los componentes principales de un sistema RFID, reader y tag con sus respectivas antenas.

##### 3.2.1 Reader

El reader es un dispositivo que puede leer y escribir datos hacia tags RFID compatibles. Es el elemento principal de un sistema RFID y tiene los siguientes componentes:

- **Transmisor:** Emite potencia y envía el ciclo de reloj a través de su antena hacia los tags que se encuentran dentro de su rango de lectura.
- **Receptor:** Recibe las señales analógicas provenientes del tag a través de la antena y envía estos datos al microprocesador, donde esta información es convertida en su equivalente digital.
- **Antena:** Va conectada directamente al transmisor y al receptor. Existen lectores con múltiples puertos para antenas, lo que les permite tener múltiples antenas y extender su cobertura.
- **Microprocesador:** Es necesario para implementar el protocolo de lectura para la comunicación con tags compatibles. Decodifica y realiza verificación de errores a las señales recibidas.
- **Memoria:** Utilizada para el almacenamiento de información como los parámetros de configuración del lector y una lista de las últimas lecturas.
- **Canales de Entrada/Salida:** Es un componente opcional que permite al lector interoperar con sensores y actuadores externos.
- **Controlador:** Permite a una entidad externa comunicarse y controlar las acciones del lector.

- Interfaz de comunicación: Provee las instrucciones de comunicación que permiten la interacción con entidades externas para transferir datos y recibir comandos.
- Fuente de alimentación.

Conforme a la especificación EPCglobal, el reader conocerá los requisitos, implementará los comandos obligatorios definidos en esta implementación, modulará/transmitirá y recibirá/demodulará el conjunto de señales eléctricas ajustándose a la normativa de señalización y será conforme a la regulación local de radiofrecuencia. Además, un reader puede implementar subconjuntos de comandos opcionales y propios conformes con esta especificación, no pudiendo implementar cualquier comando que suponga un conflicto.

## 3.2.2 Antenas

El reader se comunica con los tags a través de la antena. Ésta es la que crea el campo electromagnético que induce la corriente a la antena del tag. Por tanto, para conseguir que se lea un tag deberá estar próximo a la antena del lector. Es importante hallar el rango de acción de la antena, es decir, donde mejor reparte la señal y es más fácil leer el tag, prestando especial atención a las zonas muertas. Para mejorar la robustez y el alcance de la antena es importante tener en cuenta su polarización (dirección en la que son emitidas las ondas), pudiendo distinguir entre polarización lineal y circular. La polarización lineal sólo se emplea cuando la orientación del tag está predefinida, mientras que, la polarización circular se puede emplear sin esta condición ya que no se ve afectada por la orientación del tag, además, presenta un mayor alcance. La potencia de la antena está limitada a los estándares adoptados en cada país.

## 3.2.3 Tags

Como se comentó en el capítulo anterior, un tag se puede clasificar dependiendo de si fuente de alimentación. Un tag pasivo está compuesto de antena y microprocesador, mientras que los tags activos y semipasivos están formados por antena, microprocesador, fuente de alimentación y electrónica integrada (sensores, controladores...). Las principales características físicas que hoy en día cumplen gran parte de los tags pasivos que se comercializan son:

- Frecuencia de trabajo: UHF, 868 MHz-928 MHz
- Rango de comunicaciones: 10cm-3m
- Disponibilidad de memoria:96-256 bits
- Modulación: ASK
- Codificación de datos: Banda base.
- Velocidad de transmisión/recepción: 40 Kbps
- Puertas lógicas y generador de números aleatorios.

Conforme a la especificación EPCglobal, el tag conocerá los requisitos, implementará los comandos obligatorios definidos en esta implementación, modulará la señal backscatter (modo de comunicación detallado en la sección 3.4) solo después de recibir los comandos requeridos procedentes del reader y será conforme a la regulación local de radiofrecuencia. Además, un tag puede implementar subconjuntos de comandos opcionales y propios conformes con esta especificación, no pudiendo implementar cualquier comando que suponga un conflicto.

### 3.3 Implementación software

La información que envía el tag al reader se puede desglosar en: 96 bits para el número de identificación único (dato EPC), 16 bits de código de control de errores (CRC) y 24 bits de código de habilitación / deshabilitación (kill code).

#### 3.2.1 Formato de paquetes EPC

En la figura 3.2 se muestra el contenido del paquete de datos EPC así como el tamaño y el significado de cada uno de sus componentes.

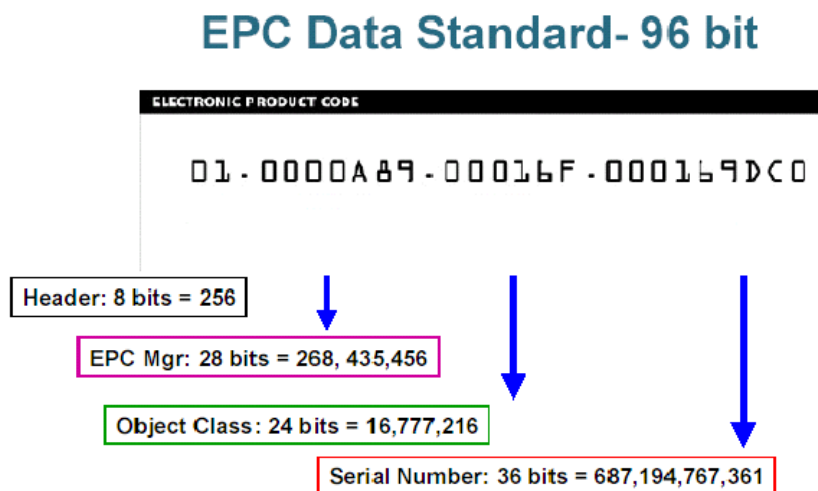


Figura 3.2: Formato de paquete EPC

Header [0 a 7 bits]: Número de versión

EPC Mgr [8 a 35 bits]: Número de administración de dominio o fabricante.

Object Class [36 a 59 bits]: Identificador del objeto o producto.

Serial Number [60 a 95 bits]: Número de seriado

#### 3.2.2 Control de errores

El CRC o control de redundancia cíclica le permite al reader comprobar que recibió correctamente el paquete, ya que el CRC es calculado por el tag que envía el paquete agregándolo al final de éste, y es recalculado por el reader. Los resultados obtenidos, tienen que ser los mismos para considerar una recepción correcta.

#### 3.3.3 Seguridad en EPC

En EPC, al igual que para básicamente todas las tecnologías inalámbricas, la seguridad en la información es uno de los objetivos fundamentales. Se busca asegurar aspectos como la confidencialidad, integridad, disponibilidad, autenticidad y autorización. Un tag, por simple que sea, envía su identificador único a cualquier lector de forma que fácilmente se puede obtener información acerca de él, por ejemplo, se puede obtener información personal acerca del pasaporte de una persona. Un esquema de seguridad podría ser deshabilitar los tags una vez usados y antes de ponerlos al alcance de un posible usurpador.

## 3.4 Protocolo de comunicación Reader-Tags

Dependiendo del tipo de tag, la comunicación entre éste y el reader puede ser de distintos tipos:

- Modulación backscatter. Es empleada tanto para los tags activos como para los pasivos. El lector envía una señal de radiofrecuencia continua que contiene corriente alterna y el reloj de la señal a la frecuencia en la que trabaja el tag. El tag obtiene la energía transformándola a corriente continua y alimentando los sistemas.
- Tipo transmisor. Comunicación usada solo para los tags activos. El tag emite su mensaje al entorno en intervalos regulares independientemente de la existencia o no de un lector.
- Tipo transpondedor. El tag permanece en un estado de letargo mientras no recibe peticiones del lector. En este estado, el tag envía periódicamente un mensaje para consultar la existencia del lector. El lector puede decidir cuando responder con un “wake up” con lo que el tag comienza a mandar su mensaje.

### 3.4.1 Monitorización del entorno

Los lectores pueden realizar una exploración de los tags de su entorno para realizar inventarios o, si los tags se asocian a sensores y pueden obtener información acerca de las condiciones del entorno. Si un reader intenta trabajar con un conjunto de tags debe conocer los dispositivos que se encuentran en su área de cobertura para después recorrerlos uno a uno o bien hacer uso de protocolos anticollisión.

Es habitual necesitar tags de varios tipos en una aplicación distribuida. Los tags pasivos no pueden realizar labores de monitorización continua sino que realizan tareas bajo demanda cuando los readers se las solicitan. Son útiles para realizar actividades regulares y bien definidas con necesidades de almacenamiento y seguridad acotadas. Si hay accesos frecuentes, continuos o impredecibles, o bien existen requerimientos de tiempo real o procesamiento de datos (como búsqueda en bases de datos internas) suele ser conveniente el empleo de tags activos.

### 3.4.2 Protocolo de identificación

Antes de comenzar un ciclo de identificación, el reader envía un paquete Broadcast a la población de tags en cobertura, indicando si los tags que han recibido el paquete deben identificarse o no. Los  $N$  tags que reciben el paquete y deben identificarse responden enviando su ID, produciéndose una colisión múltiple que el reader detectará. En este instante, el reader comenzará un ciclo de identificación transmitiendo un paquete Query con un campo de cuatro bits en el que se almacena el valor en hexadecimal  $Q$  donde  $Q \in [0,15]$ .

Los tags, al recibir el paquete generan un número aleatorio  $r$ , cuyo valor estará comprendido en el rango  $[0, 2^{(Q-1)}]$ . A continuación, se inicializará un contador con el valor  $r$  obtenido, el cual, irá decrementando al comienzo de cada slot. En el momento en el que un tag obtiene un valor de dicho contador a 0, transmitirá su identificador en ese instante. Una vez iniciado el ciclo de identificación, el comportamiento del sistema será el siguiente. Si se detecta un slot con un único identificador, el reader enviará un paquete ACK a todos los tags en su zona de cobertura, de modo que el tag que había enviado su ID contestará enviando ahora sus datos almacenados. El reader comprobará los datos, si son correctos, enviará una señal QueryRep a todos los tags, indicando que comienza un nuevo slot (incrementará su contador de slots). Por el contrario, si los datos recibidos son incorrectos, enviará un paquete NACK a todos los



tags, seguido de un QueryRep. Si el tag que se había identificado recibe ahora un paquete QueryRep, su proceso de identificación habrá finalizado.

Los tags que reciben QueryRep y no habían intentado identificarse en este slot, decrementan su contador de slot ( $r$ ). Sin embargo, si el tag que se había identificado y había enviado sus datos recibe un paquete NACK seguido de un QueryRep, actualizará su contador a  $2^{(Q-1)}$ , evitando así que el tag que envió sin éxito sus datos siga compitiendo en el ciclo actual. Los tags que reciben NACK seguido de QueryRep y no habían intentado identificarse en esta ranura, decrementan su contador de slots. Por otro lado, si tras expirar el temporizador de slot, no se ha recibido ninguna respuesta procedente de algún tag, el reader enviará un paquete QueryRep a todos los tags indicando el comienzo de un nuevo slot. Cada tag, al recibir el paquete QueryRep, decrementará su contador de slots.

Si en un slot se ha producido una colisión, el reader enviará un paquete QueryRep a los tags para informarles que comienza un nuevo slot. Los tags que reciben dicha señal y no habían enviado su ID, simplemente decrementarán su contador de slot. Aquellos tags que habían enviado su ID actualizarán su contador a  $2^{(Q-1)}$  de forma que ya no competirán en este ciclo de identificación.

Cuando el contador de slots del reader alcance el valor máximo de  $2^{(Q-1)}$ , finalizará un ciclo. El reader enviará un nuevo paquete Query (QueryAdjust si implementa algún algoritmo de ajuste de la trama). Todos los tags que aún estén sin identificar volverán a competir eligiendo un nuevo slot. El proceso finalizará cuando todos los tags se hayan identificado correctamente.

### 3.4.3 Volcado de datos

Una vez que el lector recibe la señal que contiene la información almacenada en el tag, la transmite a una aplicación que se encarga de asociar el identificador almacenado en el tag en cuestión con la información referente al producto al que se encuentra adherido. Una vez procesados los datos, se transmiten a los sistemas de gestión que se encargan de actualizar la información correspondiente.

### 3.4.4 Otras funciones

Se pueden nombrar otras funciones relacionadas con EPC como por ejemplo, el uso de una simple autenticación de password como herramienta de seguridad que emplean algunos tags. Otra posible herramienta de seguridad, ya nombrada con anterioridad, consiste en habilitar/deshabilitar los tags mediante el comando kill.

Una función que ofrece muchas ventajas es la escritura en los tags de lectura / escritura. Al contrario que los tags de sólo lectura, ofrecen la habilidad de contar con información actualizada, por tanto, son necesarias en aplicaciones con información variable. Una aplicación de este tipo es, por ejemplo, en las cadenas de montaje de automóviles, donde la etiqueta cuenta con un código que se va actualizando cada vez que se realiza alguna operación para identificar el estado actual del vehículo en la cadena.

## 3.5 Protocolo anticolidión con trama adaptativa

Puesto que en la realidad el reader no conoce previamente el número de tags que van a competir en cada ciclo de identificación, es necesario calcular el valor óptimo de la trama para minimizar el tiempo medio de identificación y que por tanto se presente una mejor respuesta en términos de probabilidad de pérdida de tag. A continuación se detalla el algoritmo adaptativo de  $Q$  en cada ciclo y una modificación

de dicho algoritmo propuesta por el grupo de investigación del área de Telemática de la UPCT, la trama adaptativa de Q en cada slot.

### 3.5.1 Trama adaptativa ciclo a ciclo

El objetivo de este algoritmo es calcular el valor óptimo de la Q para maximizar el número de identificaciones por ciclo. Se parte de las siguientes consideraciones:

- El reader comienza el primer ciclo con un valor de  $Q=4$ .
- El reader dispone de contadores para cuantificar el número de slots con ID, vacíos y con colisión, así como el número total de slots en cada ciclo.
- Al finalizar un ciclo, se contabilizan el número de slots con ID y vacíos y se ajusta el valor de la Q.
- C es una variable de ajuste que toma valores  $0.1 < C < 0.5$ . El reader utiliza valores altos de C cuando Q es un valor bajo y viceversa.

El ajuste de la Q se realiza siguiendo el flujo del esquema de la figura 3.3

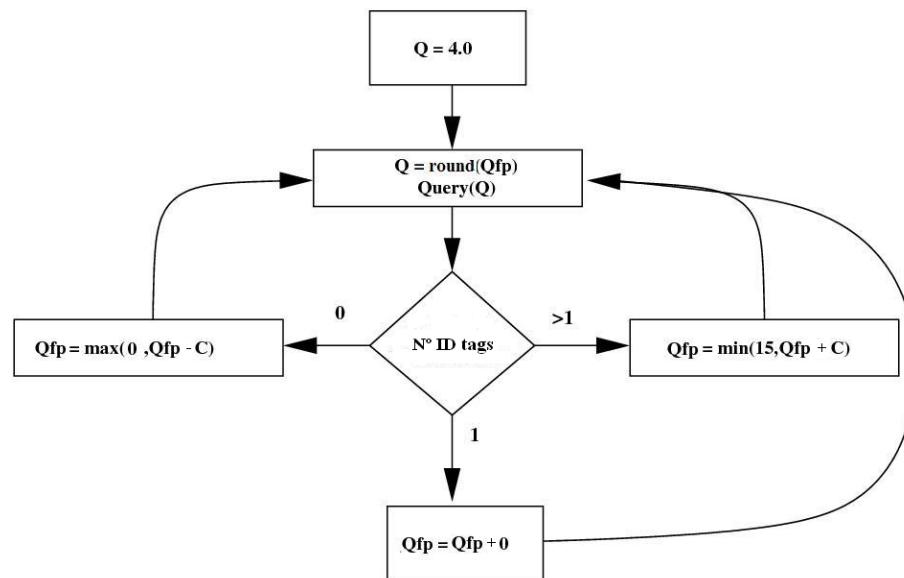


Figura 3.3: Algoritmo de ajuste de la Q

### 3.5.2 Trama adaptativa slot a slot

En esta variación del algoritmo anterior, el reader calcula un nuevo valor de Q al finalizar un slot considerando si era un slot vacío, con colisión o un slot ocupado con un identificador. Así, los tags eligen un slot aleatorio para transmitir, al finalizar cada slot, de forma que:

- Si al finalizar el slot temporal éste se encuentra vacío, comienza un nuevo ciclo donde el reader disminuye la longitud de la trama Q. De esta forma se evita que en el nuevo ciclo se sucedan una gran cantidad de slots vacíos, lo que puede causar un mayor retardo en la identificación, y por tanto una mayor probabilidad de pérdida de tag.
- Si el slot temporal es un slot con colisión, comenzará un nuevo ciclo. La longitud de la trama Q en este nuevo ciclo será mayor. Al aumentar la Q, se disminuye la probabilidad de colisión en el siguiente ciclo, y por tanto, se

evita un mayor retardo en la identificación, lo que puede llevar una mayor probabilidad de pérdida de tag.

- Si el slot es ocupado, significa que un tag ha podido identificarse satisfactoriamente. En esta situación el ciclo continúa.

Al igual que en el caso del algoritmo adaptativo ciclo a ciclo, comienza con un valor de  $Q=4$ .



# CAPÍTULO 4

## VALIDACIÓN DEL ESTÁNDAR EPCGLOBAL CLASS-1 GEN-2

---

### 4.1 Introducción a la validación

La validación es una de las principales etapas de un estudio de simulación. A través de esta es posible detallar deficiencias en la definición de un protocolo. Consiste en evaluar un sistema o parte de este durante o al final de su desarrollo para determinar si satisface los requisitos iniciales, comprobándose también datos como el rendimiento y la capacidad. Existen varias formas de validar un protocolo, entre otras, se puede validar un protocolo haciendo uso de la opinión de expertos sobre los resultados de la simulación, estudiando la exactitud de los datos obtenidos con los estudios previos y comprobando los fallos del modelo por la persona que hará uso de los resultados que arroja el experimento de simulación.

### 4.2 Herramientas de validación. La elección de SDL

Para la validación de protocolos se emplean lenguajes formales o semiformales denominados lenguajes de especificación o de descripción. Podemos nombrar algunos lenguajes como LOTOS (basado en álgebra de procesos y en tipos de datos abstractos), SDL (basado en máquinas de estado finitas extendidas y que permite el modelado gráfico o en texto) y ESTELLE (semejante a PASCAL). Estas herramientas deben permitir expresar las características y el comportamiento del sistema, enlazando las herramientas de simulación, modelado e implementación y además, deben ser intuitivas para el usuario. Para el desarrollo de este proyecto, hemos elegido SDL and TTCN Suite 4.3 por las ventajas que ofrece en la especificación, implementación y pruebas de servicios. Telelogic SDL Suite™ es una herramienta de desarrollo de software en tiempo real que proporciona capacidades de especificación de software de comunicaciones y desarrollo de software para complejos sistemas de comunicación orientados a eventos (por ejemplo, protocolos de comunicación de datos, gestión de redes, mantenimiento y tratamiento de fallos en sistemas de telecomunicaciones...) y que se describen mediante el estándar Lenguaje de Especificación y Descripción (SDL) de ITU-TZ.100. Entre sus características se incluyen la posibilidad de describir escenarios mediante un lenguaje de seguimiento visual y generar código ejecutable y documentación automáticamente. Además, permite integraciones flexibles con otros lenguajes y generar sus conjuntos de pruebas de TTCN. SDL Suite se basa en SDL, un lenguaje de diseño orientado a objetos, y en el lenguaje de seguimiento MSC (secuencia de mensajes gráficos UIT-TZ.120), siendo ambos lenguajes formales, visuales y estandarizados internacionalmente.

Junto con Telelogic TTCN Suite™, muchos desarrolladores de productos de comunicaciones han utilizado satisfactoriamente SDL Suite. El lenguaje de comandos de pruebas de TTCN Suite, es un amplio lenguaje de programación específico para realizar pruebas de conformidad y cumple con los estándares ETSI e ISO. Podemos destacar que incluye un editor gráfico moderno y depuración de conjuntos de pruebas. Entre sus características se incluye también la generación automática de casos de pruebas a partir de sus especificaciones y la integración con su entorno de pruebas.

## 4.3 El lenguaje SDL

La entidad principal en SDL es el sistema (SYSTEM) que se compone de bloques (BLOCK). Los bloques se conectan entre sí y con el entorno mediante señales (SIGNAL), que son transportadas por canales. Cada bloque puede contener varios procesos, los cuales están modelados como máquinas de estados finitas extendidas y por tanto, añade el uso de variables, parámetros, acciones y temporizadores. Una transición de un estado a otro se produce cuando se recibe una señal válida de otro proceso o del entorno, una vez recibida, se puede manipular los datos locales al proceso o enviar señales a otros procesos o al entorno y a continuación, se permanece a la espera en otro estado. En la Tabla 4.1 se presenta una breve descripción de los principales símbolos SDL empleados en nuestra implementación:






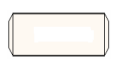
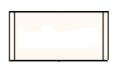







SÍMBOLO	NOMBRE	DESCRIPCIÓN
	Include	Incluye librerías en el sistema.
	Text	Declaración de variables (DCL), de constantes (SYNONYM) y de señales (SIGNAL)
	Arranque proceso	Símbolo inicial de un proceso.
	Estado	El sistema permanece a la espera de recibir una señal que producirá una transición de estado. Cada estado tendrá un identificador.
	Entrada	Permite el consumo de la señal de entrada especificada, poniendo a disposición del sistema la información que transporta. Para que no se descarte el valor de la señal, debe tener una variable asociada.
	Procedimiento	Último objeto en el que se puede dividir un proceso. Conjunto de instrucciones.
	Llamada a procedimiento	Permite la ejecución de un procedimiento declarado en un proceso.
	Salida	Permite el envío de información a través de una señal.
	Tarea	Contiene sentencias de asignación.
	Decisión	Transfiere el control al trayecto cuya condición contiene el valor dado por la interpretación de la pregunta.
	Conector	Representa la continuación de un trayecto desde otro conector con el mismo nombre y en la misma área.
	Inicio procedimiento	Símbolo de inicio de procedimiento
	Fin procedimiento	Símbolo final de procedimiento.
	Create Request	Se emplea para crear procesos hijo o instancias de proceso.

Tabla 4.1 : Simbología en SDL

De un proceso, es posible crear varias instancias con las que se va a poder trabajar haciendo uso del identificador único de proceso o PID. Este puede ser pasado como parámetro en señales, y puede utilizarse en llamadas a procedimientos y en operaciones. Obtenemos el PID usando las siguientes funciones básicas:

- Self: contiene el PID de la instancia actual.
- Sender: Contiene el PID de la instancia que le ha enviado la última señal consumida.
- Parent: Contiene el PID de la instancia padre (la que crea la instancia actual).
- Offspring: Contiene el PID de la última instancia creada. Su valor debe ser almacenado en una variable de tipo PID para su uso posterior.

Para enviar una señal a un proceso concreto usaremos la sentencia “señal TO pid\_proceso”, donde pid\_proceso contiene el PID de dicho proceso. Un proceso puede enviarle señales a quien lo creó (su padre) con la sentencia “señal TO parent” , donde parent contiene dicho PID.

## 4.4 Comunicación del sistema RFID

Para comenzar, se muestra el esquema organizador del sistema RFID que se ha implementado.

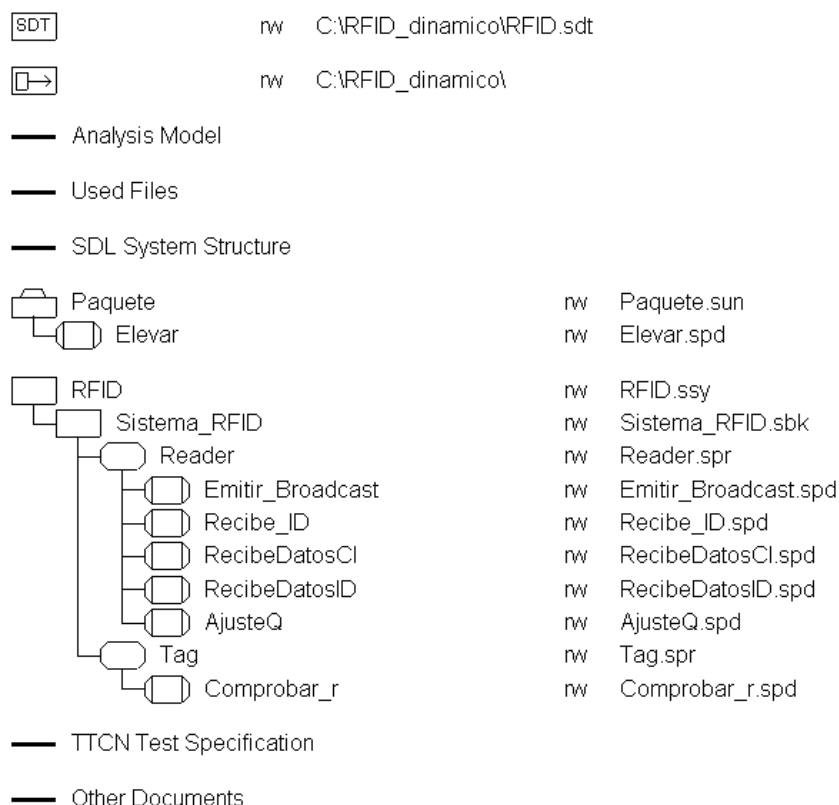


Figura 4.1: Esquema organizador del sistema RFID

Consta de un bloque denominado Sistema\_RFID que a su vez está formado por los procesos Reader y Tag, cada uno de los cuales incluyen un conjunto de

procedimientos. A continuación, accedemos al Sistema\_RFID y observamos la comunicación entre el Reader y el Tag.

block Sistema\_RFID

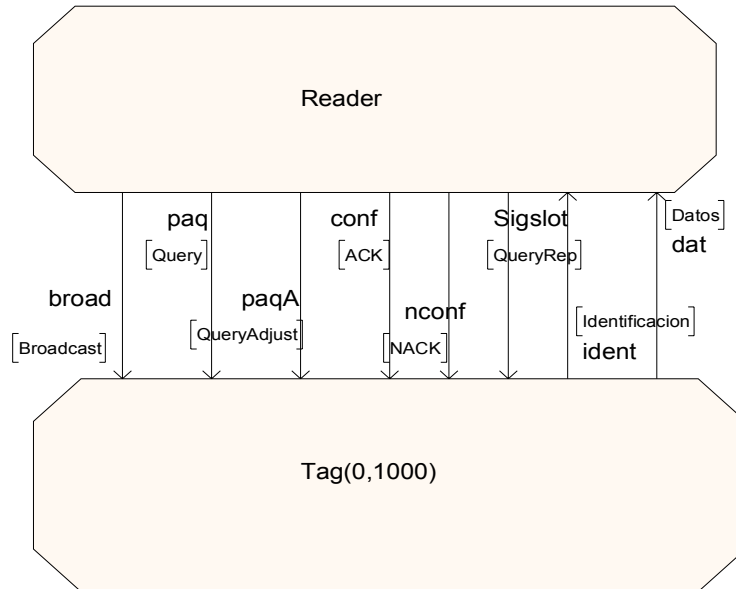


Figura 4.2: Comunicación entre reader y tags

Al comenzar existirá un solo proceso Reader. El proceso Reader creará un número de procesos hijos (entre 0 y 1000), el cual, es introducido por el usuario externamente. El par (n,m) al lado del proceso Tag representa el número mínimo y máximo de instancias de ese proceso, si se omitiera se crearía una única instancia. Para que se comuniquen es necesario establecer un canal por el que se van a direccionar cada una de las señales y su sentido (Tabla 4.2).

Canal	Señal	Origen	Destino
broad	Broadcast	Reader	Tag
paq	Query	Reader	Tag
paqA	QueryAdjust	Reader	Tag
conf	ACK	Reader	Tag
nconf	NACK	Reader	Tag
Sigslot	QueryRep	Reader	Tag
ident	Identificación	Tag	Reader
dat	Datos	Tag	Reader

Tabla 4.2 : Señales entre los procesos Reader y Tag



## 4.5 Diseño del Reader

Se ha estudiado el sistema RFID atendiendo tres escenarios distintos, el primero de ellos con trama estática, el siguiente con una trama dinámica ciclo a ciclo, y por último, con una trama dinámica slot a slot. Se han definido tres Readers distintos uno para cada escenario y se muestran a continuación.

### 4.5.1 Reader en escenario con trama estática

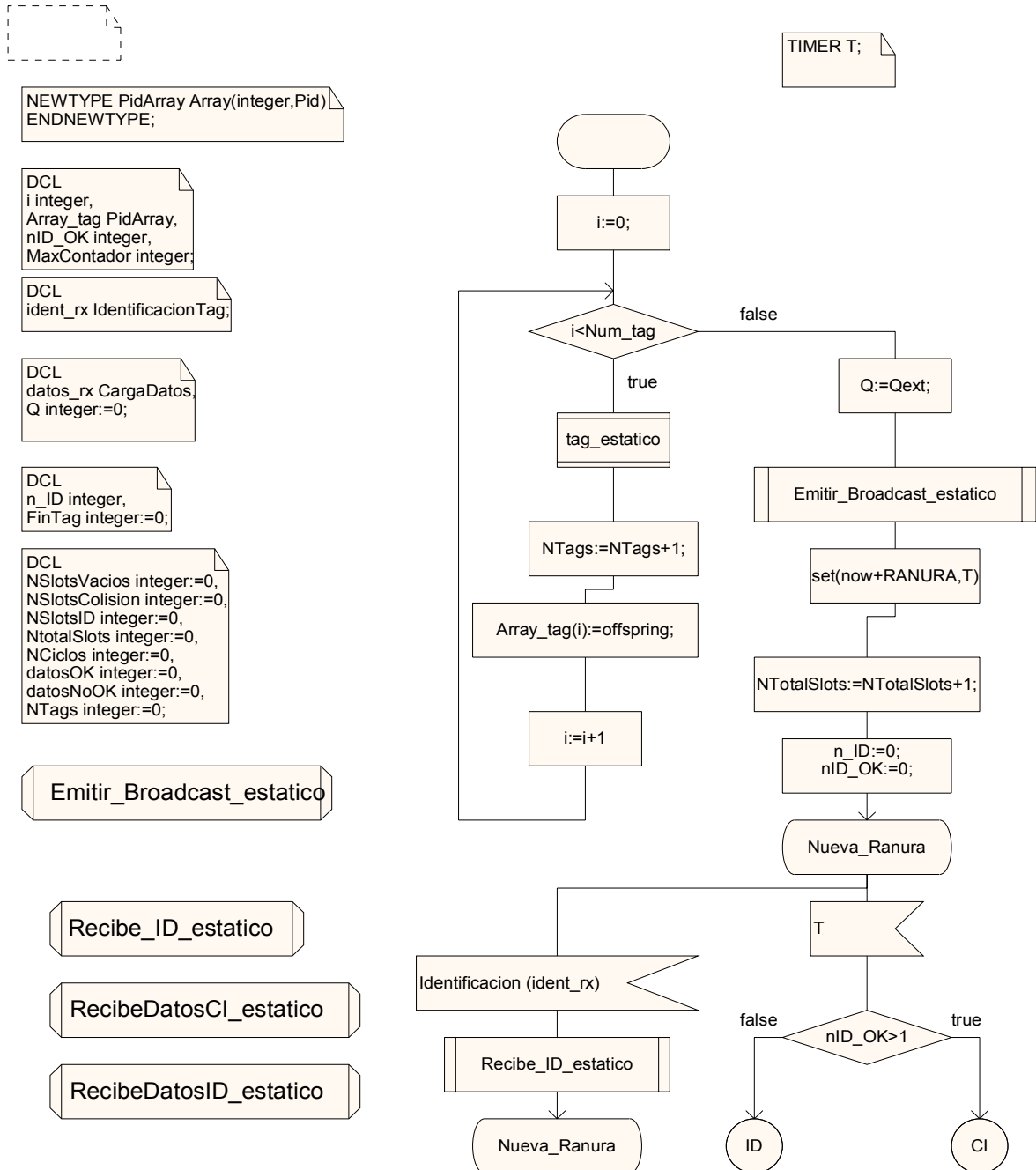


Figura 4.3: Inicio del proceso (estático)

En la figura 4.3 se muestra el inicio del proceso Reader\_estático creando Num\_tag (variable que indica el número de tags en cobertura) instancias de proceso

tag y guardando el identificador de cada tag en Array\_tag (array de PIDs). Seguidamente se ejecuta el procedimiento Emitir\_Broadcast\_estático, en el que el proceso Reader va a enviar una señal Broadcast a cada uno de los tags en cobertura, tal y como se muestra en el siguiente esquema.

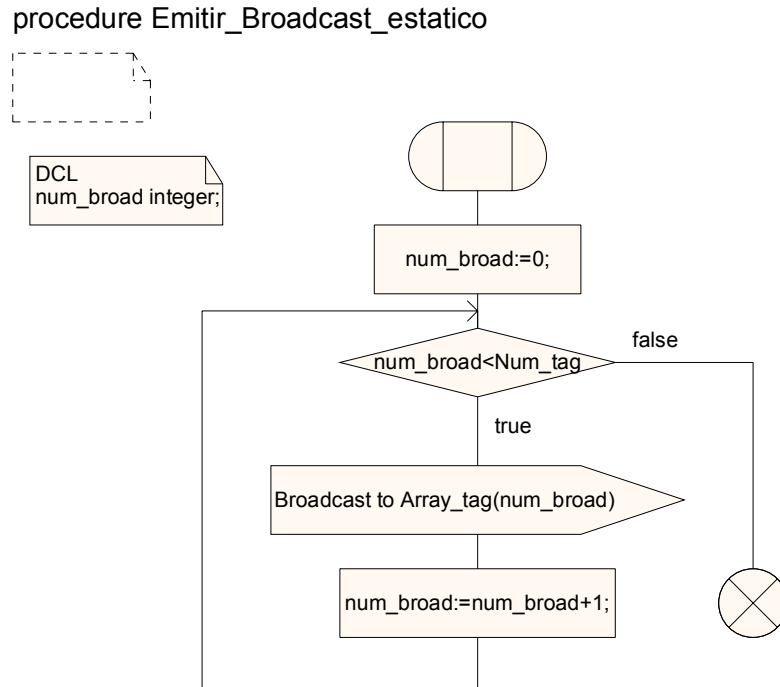


Figura 4.4: Procedimiento Emitir\_Broadcast\_estatico (estático)

Siguiendo la figura 4.3, una vez ejecutado el procedimiento Emitir\_Broadcast\_estático, se lanza el temporizador T, se incrementa NTotalSlots y se inicializan a 0 los contadores n\_ID y n\_IDOK (este método es el que se va a seguir a lo largo de toda la implementación cada vez que comience un nuevo slot). A continuación se queda a la espera de recibir una señal en el estado Nueva\_Ranura, pudiendo recibir ninguna, una o varias señales de Identificación, o una señal T indicando que ha expirado el temporizador.

#### 4.5.1.1 Se recibe señal Identificación

Como se puede observar en la figura 4.3, una vez recibida una señal identificación, se almacena el dato que transporta en ident\_rx para tratarlo en el procedimiento Recibe\_ID\_estatico. En dicho procedimiento, mostrado en la figura 4.5, se extrae del campo SerialNumber la identificación recibida y se almacena en ID\_rx, seguidamente se comprueba si la identificación del tag es correcta o no usando la función Sender, comentada en el apartado 4.3 de este capítulo. Como se puede apreciar, al recibir una identificación, se incrementa el contador n\_ID, y si la identificación es correcta, se incrementa también el contador nID\_OK.

procedure Recibe\_ID

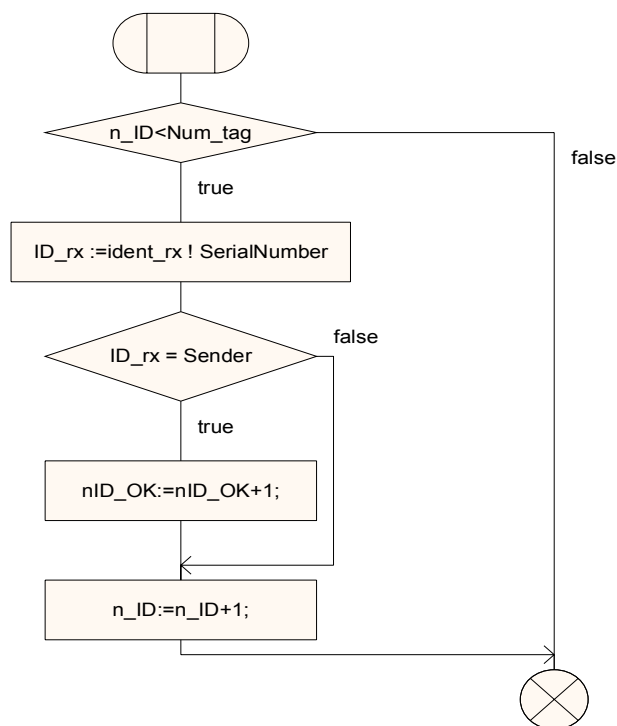
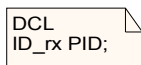
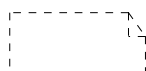


Figura 4.5: Procedimiento Recibe\_ID\_estatico (estático)

### 4.5.1.2 Se recibe señal T

Volviendo a la figura 4.3, la recepción de la señal T indica que ha expirado el temporizador. En este momento se evalúa el número de identificaciones correctas recibidas en el slot actual, existiendo varias posibilidades.

#### 4.5.1.2.1 No hay identificaciones correctas

Tal y como muestra la figura 4.6, si no hay identificaciones correctas (incrementa NSlotsVacios) comienza un nuevo slot y vuelve al estado Nueva\_Ranura, localizado en la figura 4.3.

#### 4.5.1.2.2 Hay una identificación correcta

Si hay una identificación correcta se incrementa NSlotsID, se envía una señal ACK a todos los tags y se ejecuta el procedimiento RecibeDatosID\_estatico. En este procedimiento, mostrado en la figura 4.7, se va a comprobar si los datos recibidos y almacenados en datos\_rx son correctos o no, es decir, si contienen CodigoEPC u OtroCodigo respectivamente. Si los datos son correctos, se incrementa datosOK y FinTag. Por otro lado, si los datos son incorrectos, se incrementa datosNoOK, seguidamente se envía NACK a todos los tags en cobertura y para indicar a los tags que comienza un nuevo slot por el que pueden competir se envía QueryRep.

process Reader\_estatico

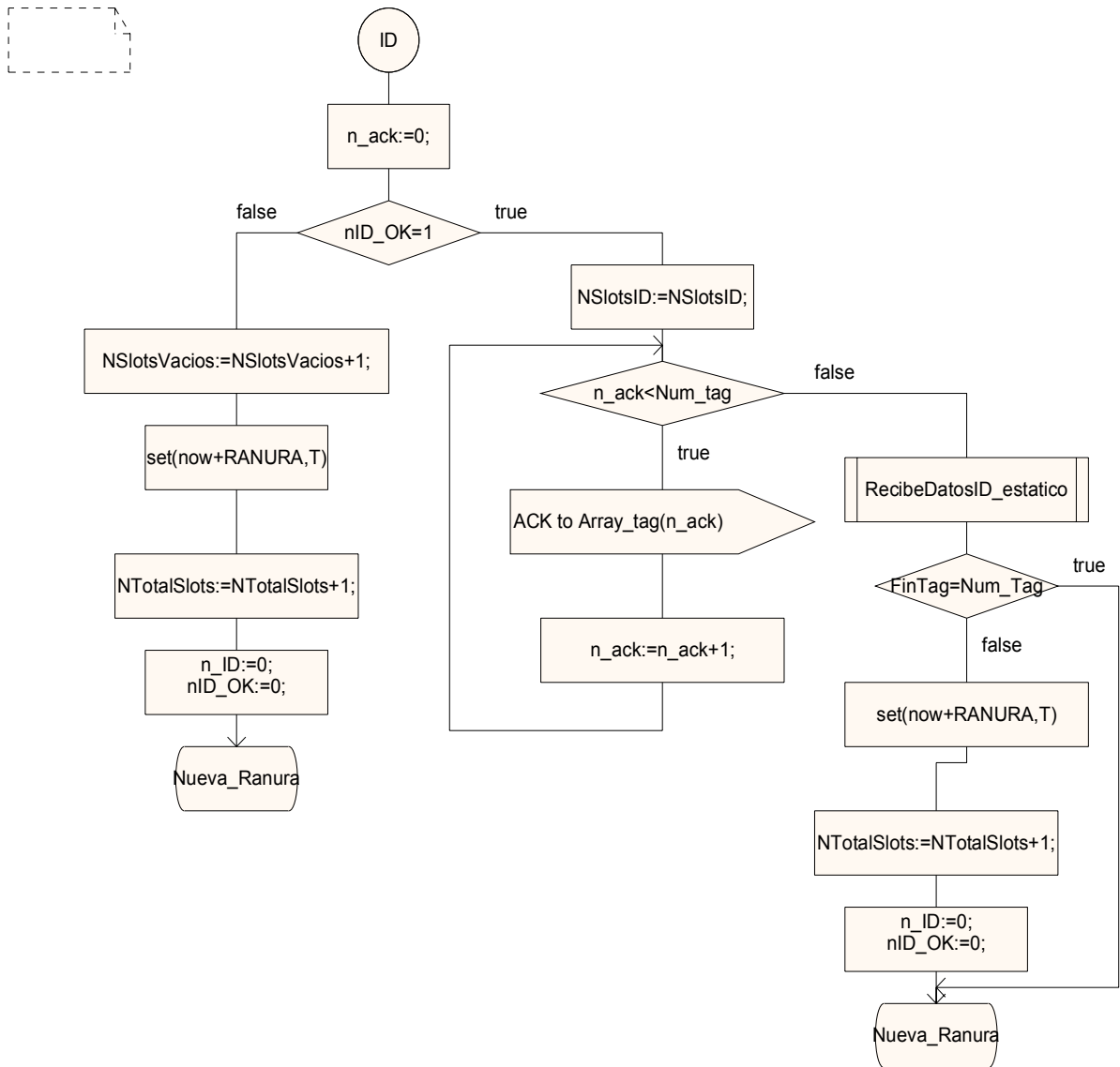


Figura 4.6: Una o ninguna identificación correcta (estático)

procedure RecibeDatosID\_estatico

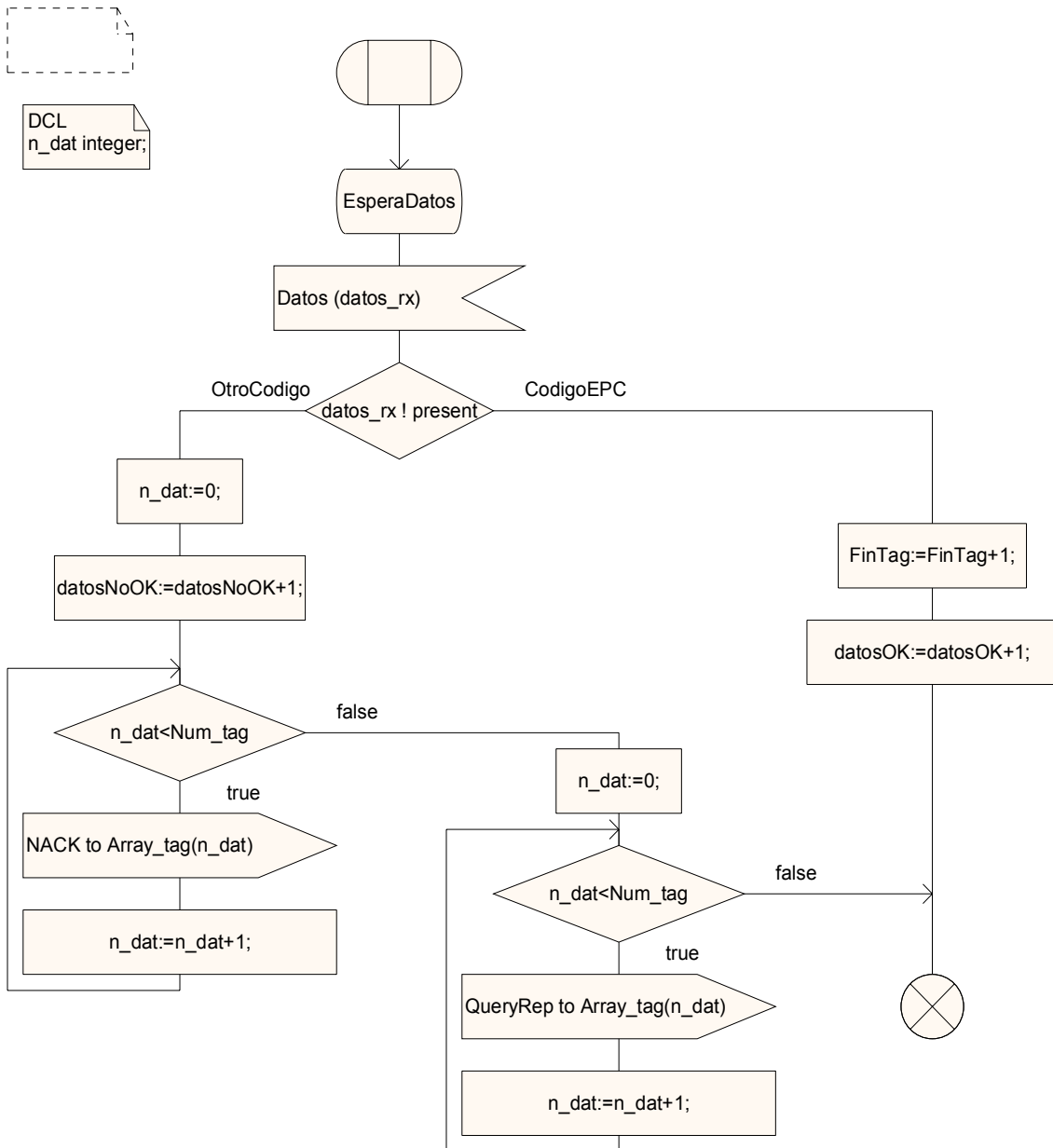


Figura 4.7: Procedimiento RecibeDatosID\_estático (estático)

Una vez finalizado el procedimiento, continuamos el flujo del diseño en la figura 4.6, donde se comprueba si FinTag es igual a NumTag; si es igual, significa que todos los tags se han identificado y han enviado sus datos correctamente, luego el proceso se para ya que los tags dejan de competir. Si no es igual, comenzará un nuevo slot y esperará en el estado Nueva\_Ranura, situado en la figura 4.3.

#### 4.5.1.2.3 Hay más de una identificación correcta

En este caso se va a producir una colisión que va a dar lugar al comienzo de un Ciclo de Identificación y se actuará, siguiendo la figura 4.8, primero incrementando el contador NSlotsColision y a continuación, notificándole la situación a todos los tags mediante una señal Query con el valor de la trama Q.

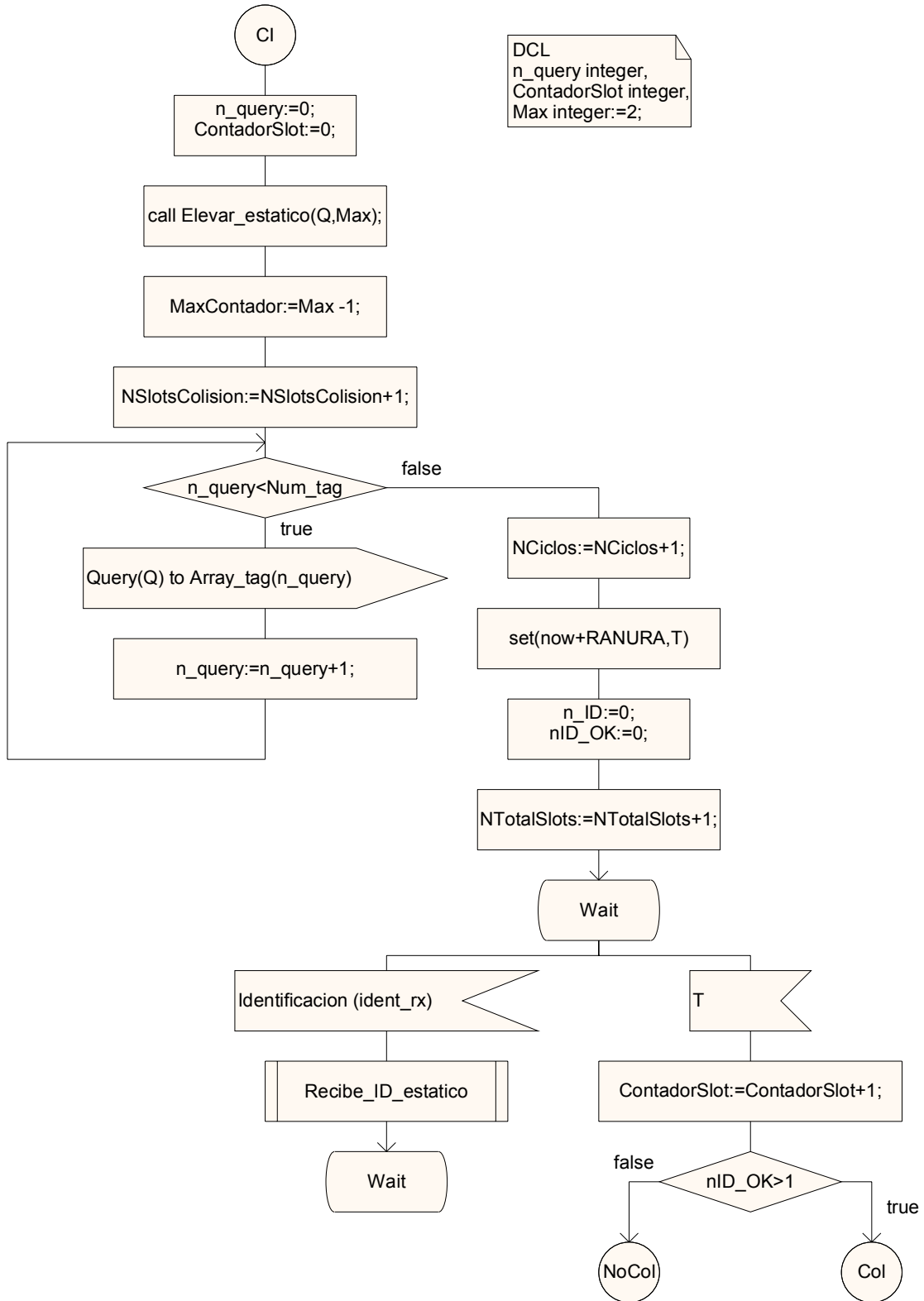


Figura 4.8: Varias identificaciones correctas (estático)

Para determinar el número de slots que tendrá este Ciclo de Identificación, haremos una llamada a la función `Elevar_estatico` pasándole como parámetro la  $Q$ . Esta función, cuya implementación se muestra en la figura 4.9, devuelve el valor  $2^Q$  que se guarda en la variable `Max`.

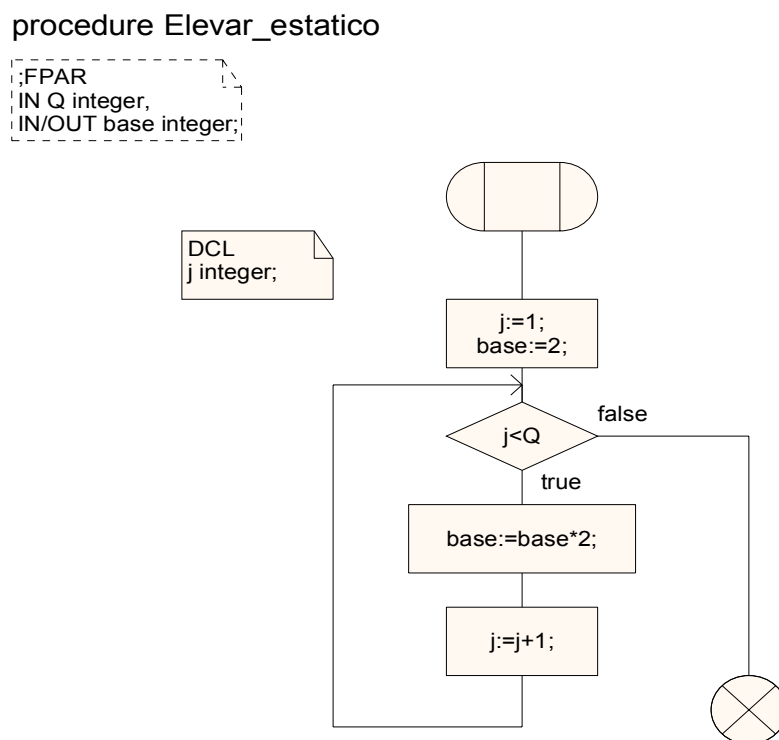


Figura 4.9: Función Elevar\_estático (estático)

Volvemos a la figura 4.8. El valor del número de slots del ciclo será  $2^Q - 1$ , y se almacena en la variable MaxContador. Posteriormente, el Reader permanecerá a la espera en el estado Wait hasta recibir una señal Identificación o una señal T.

#### 4.5.1.2.3.1 Se recibe señal Identificación

Cuando se recibe una señal de Identificación, se accede al procedimiento Recibe\_ID\_estatico, (cuyo esquema aparece en la figura 4.5 y funcionamiento ya se ha detallado) y al finalizar este, vuelve al estado Wait (figura 4.8).

#### 4.5.1.2.3.2 Se recibe señal T

Al recibir una señal T (figura 4.8) se incrementa ContadorSlot y se vuelve a evaluar el número de identificaciones recibidas en el slot actual comprobando si se ha llegado al número máximo de slots en el ciclo.

##### 4.5.1.2.3.2.1 No hay identificaciones correctas y no se ha alcanzado el número máximo de slots en el ciclo

Como se puede apreciar en la figura 4.10, primero se incrementa el contador NSlotsVacios y a continuación se envía un QueryRep a todos los tags en cobertura indicando que comienza un nuevo slot.

##### 4.5.1.2.3.2.2 Hay una identificacion correcta y no se ha alcanzado el número máximo de slots en el ciclo

Según la figura 4.10, se incrementa el contador NSlotID, se envía una señal ACK a todos los tags y a continuación se ejecuta el procedimiento RecibeDatosCI\_estatico.

Para este procedimiento, mostrado en las figuras 4.11a y 4.11b, si los datos recibidos son correctos y aún quedan tags por competir, se envía una señal QueryRep a cada uno de los tags, informando así a aquellos tags que aún pueden apostar, el comienzo de un nuevo slot. Por el contrario, si los datos recibidos son incorrectos, se enviará NACK y QueryRep.

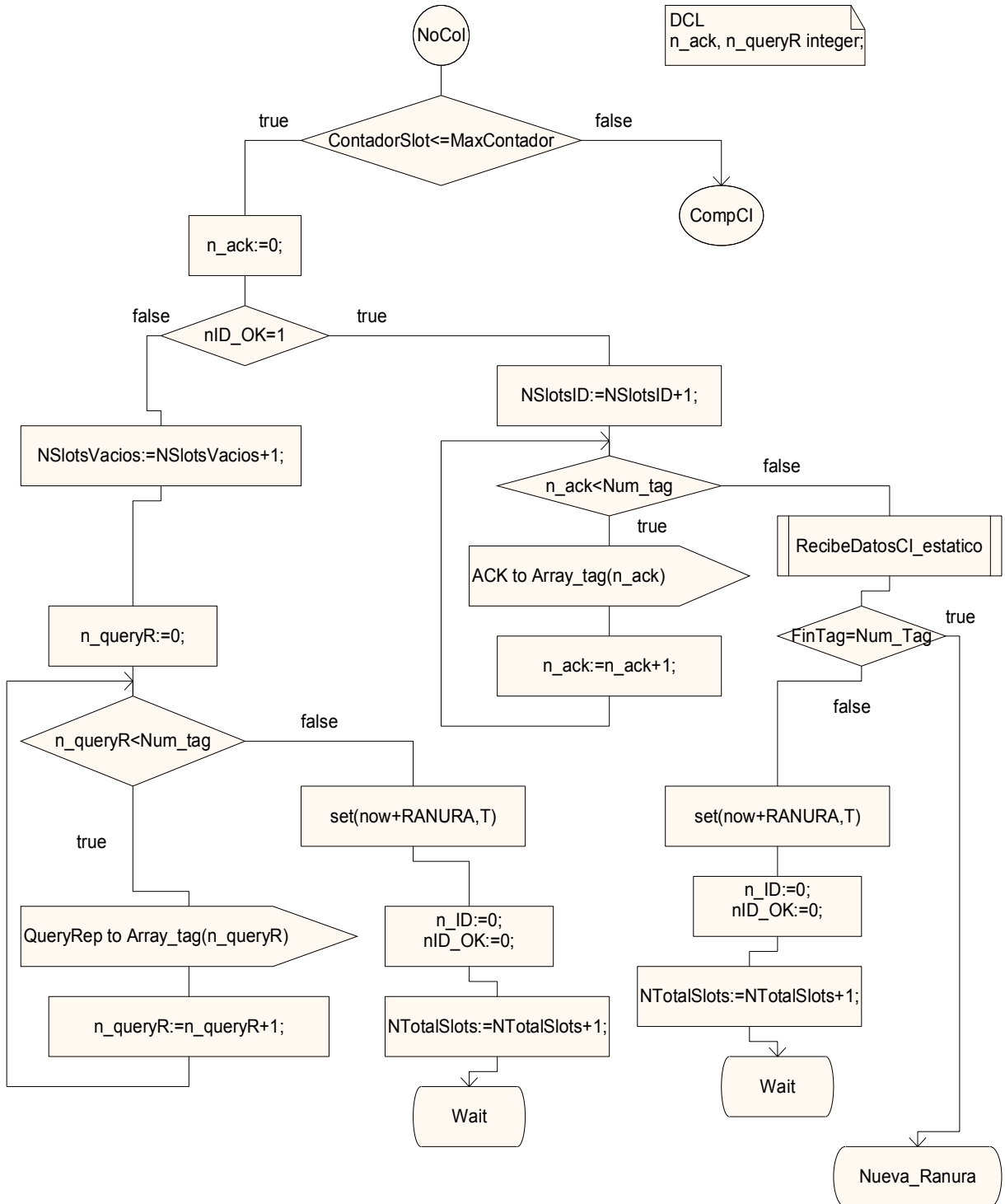


Figura 4.10: Esquema para una o ninguna identificación correcta y con número de slots dentro del rango permitido en el ciclo (estático)



procedure RecibeDatosCI\_estatico

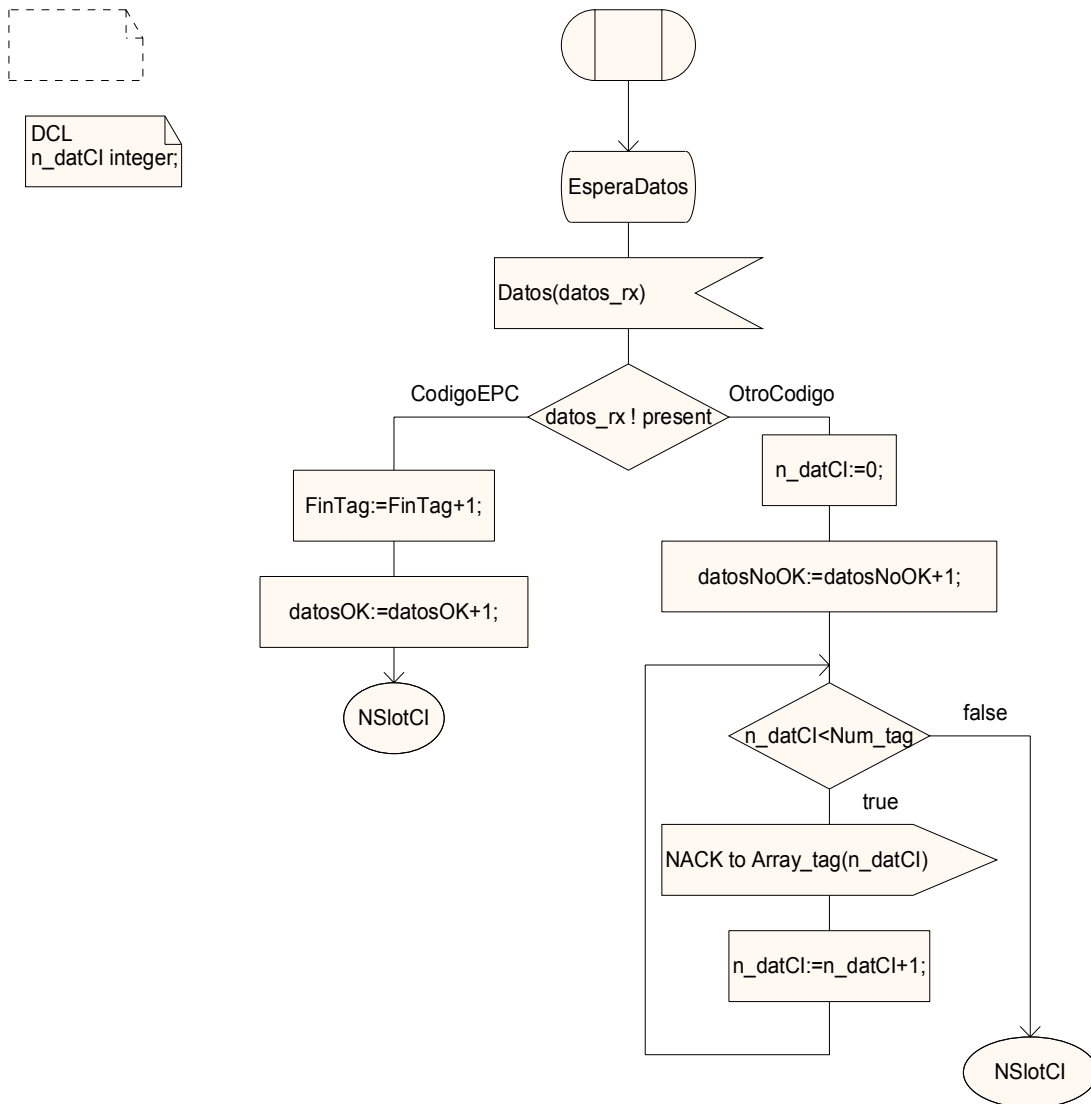


Figura 4.11a: Procedimiento RecibeDatosCI\_estático (estático)

procedure RecibeDatosCI\_estatico

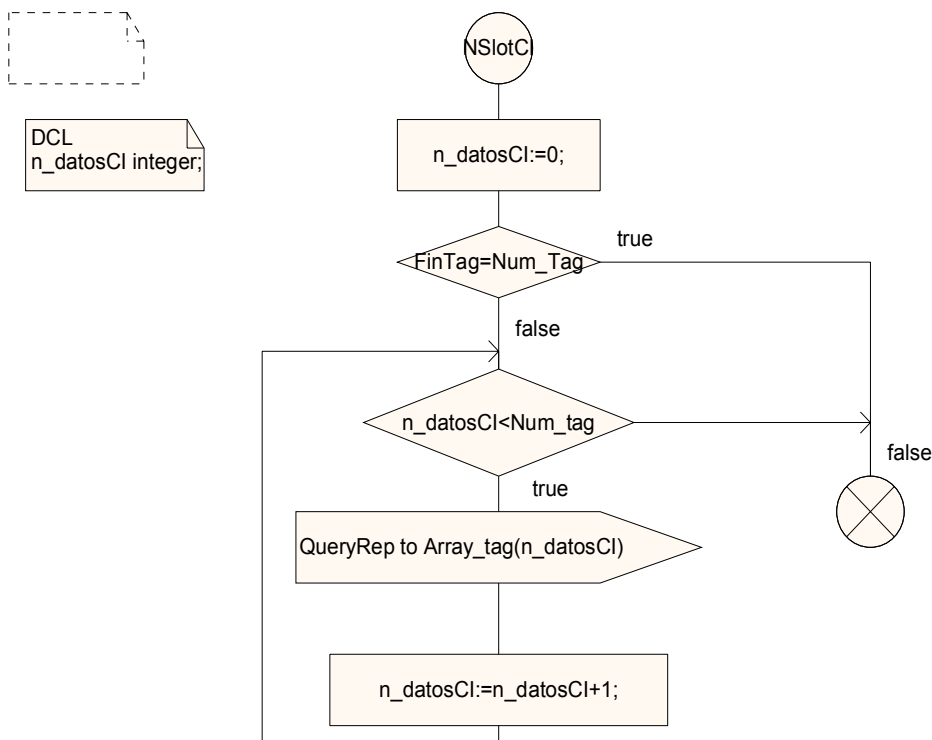


Figura 4.11b: Procedimiento RecibeDatosCI\_estático (estático)

Una vez finalizado el procedimiento volvemos a la figura 4.10. Con la variable FinTag se comprueba si todos los tags en cobertura han dejado de competir, si es así, se vuelve al estado Nueva\_Ranura (figura 4.3) a esperar que se vuelvan a detectar tags sin identificar en la zona de cobertura, y si no, da comienzo el nuevo slot y espera en el estado Wait (figura 4.8).

#### 4.5.1.2.3.2.3 Hay una o ninguna identificación y el número de slots es mayor que el posible en este ciclo.

En la figura 4.12 se puede apreciar, como se informa a los tags el comienzo de un nuevo Ciclo de Identificación enviando nuevamente la señal Query con el valor de Q. Como consecuencia, se incrementa NCiclos, se inicializa ContadorSlot a cero y además, se calcula el número máximo de slots posibles en este nuevo ciclo. Inmediatamente, el Reader permanece a la espera en el estado Wait (figura 4.8).

#### 4.5.1.2.3.2.4 Hay varias identificaciones y no se ha alcanzado el número máximo de slots posibles en este ciclo

Se producirá una nueva colisión y por tanto, se incrementa NSlotsColision, como se observa en la figura 4.13. A continuación se envía QueryRep a todos los tags y comienza un nuevo Slot.

### 4.5.1.2.3.2.5 Hay varias identificaciones y el número de slots es mayor que el posible en este ciclo

Esta situación, reflejada en la figura 4.12, dará lugar a un nuevo ciclo de igual forma que la descrita en el punto 4.5.1.2.3.2.3.

process Reader\_estatico

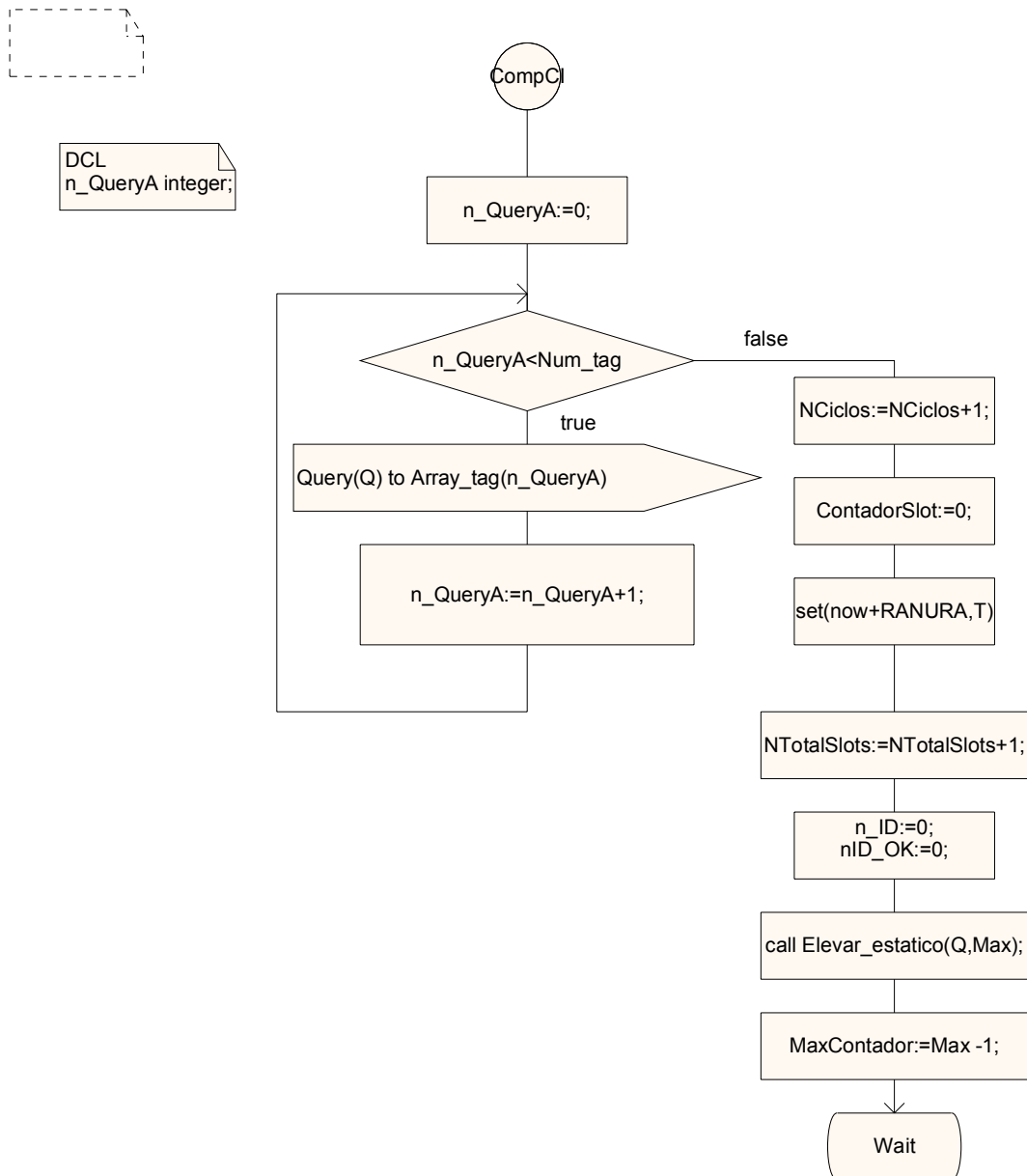


Figura 4.12: Nuevo Ciclo (estático)

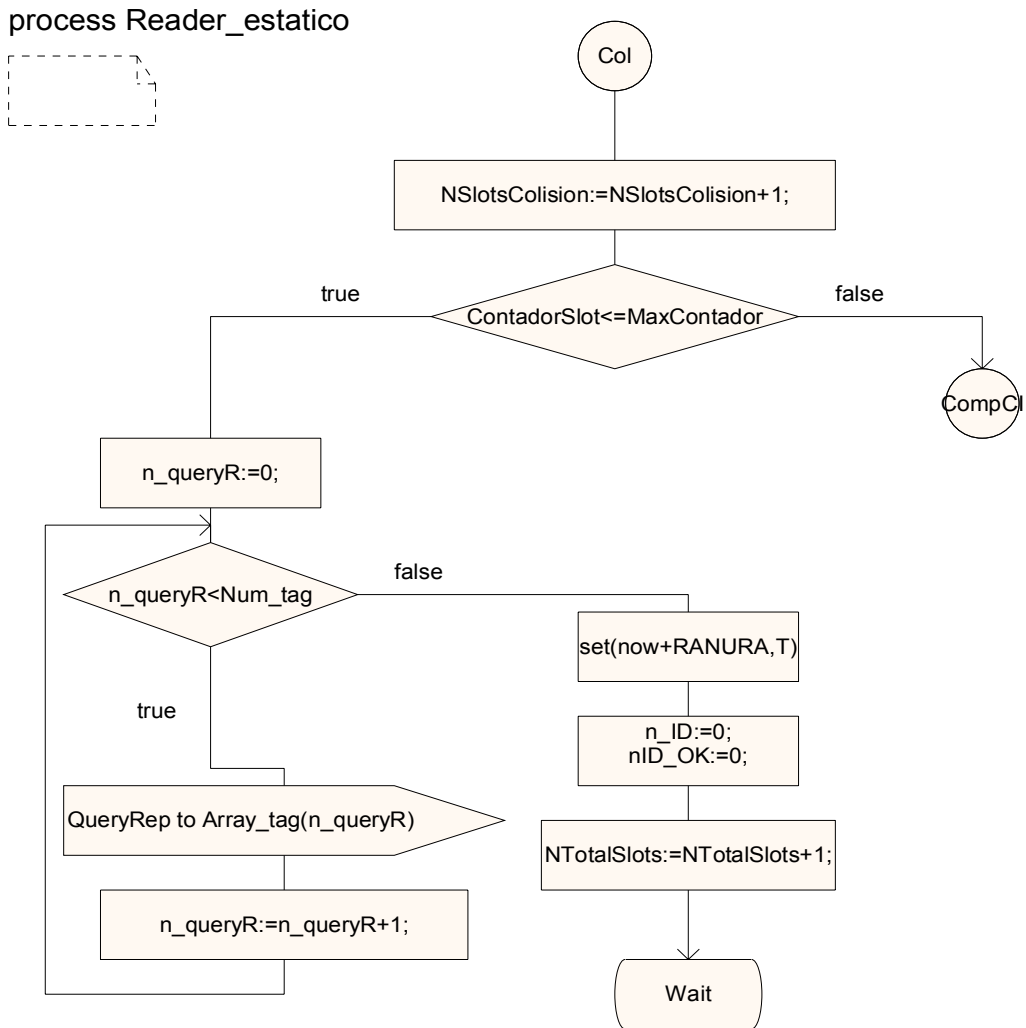


Figura 4.13: Esquema para varias identificaciones correctas y con número de slots dentro del rango permitido en el ciclo (estático)

## 4.5.2 Reader en escenario con trama adaptativa ciclo a ciclo

El proceso Reader en este escenario consta de los procedimientos Emitir\_Broadcast\_ciclo, Recibe\_ID\_ciclo, RecibeDatosID\_ciclo y RecibeDatosCI\_ciclo, similares a los procedimientos Emitir\_Broadcast\_estatico, Recibe\_ID\_ciclo\_estatico, RecibeDatosID\_estatico y RecibeDatosCI\_estatico, respectivamente. Por ello, no se van a volver a explicar para este escenario. Además contiene el procedimiento AjusteQ\_ciclo, cuyo uso, es la principal diferencia entre cada una de las implementaciones. Puesto que gran parte del desarrollo de este Reader es idéntico al del Reader descrito anteriormente, se va a detallar únicamente las diferencias con él que se han considerado oportunas para el entendimiento de la implementación. Se podrá seguir el desarrollo del flujo de este Reader con los comentarios realizados en el Reader con trama estática, y por tanto, únicamente se muestra el diseño y una simple puesta en situación para cada uno de los esquemas.

En la figura 4.14 se muestra el inicio del proceso Reader\_ciclo con la creación de las instancias de tags y la emisión de la señal Broadcast a cada uno de ellos:

process Reader\_ciclo

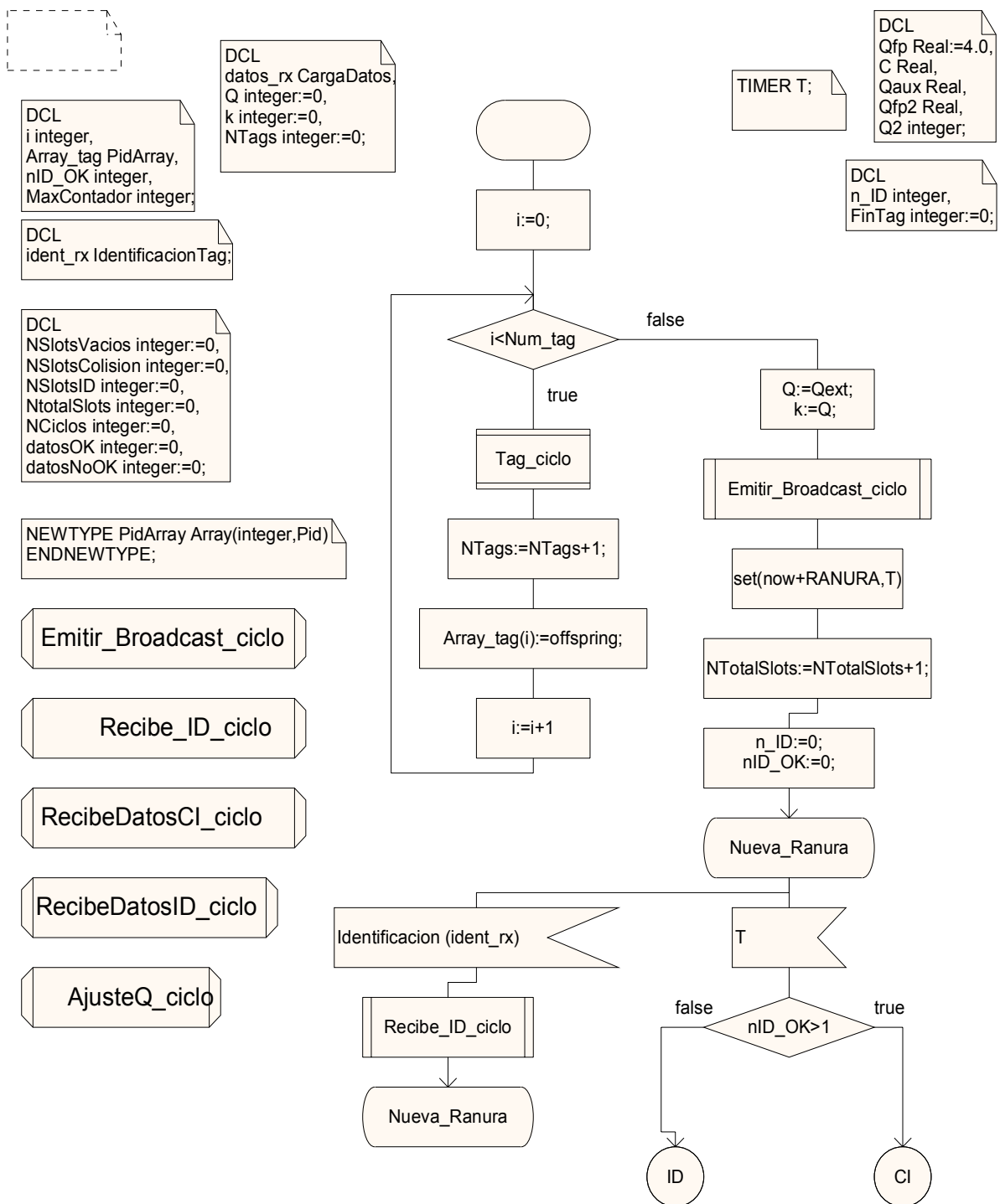


Figura 4.14: Inicio del proceso Reader\_ciclo (ciclo a ciclo)

### 4.5.2.1 Se recibe señal Identificación

Se pueden recibir una o varias señales de identificación dentro de un slot temporal (figura 4.14) y se comprobará si son correctas o no en el procedimiento Recibe\_ID\_ciclo, idéntico al ilustrado y detallado en la sección 4.5.1.1.

### 4.5.2.2 Se recibe una señal T

Se evaluarán tres posibilidades, si no se detectan identificaciones correctas, si se recibe solo una o si se reciben varias.

#### 4.5.2.2.1 No hay identificaciones correctas

En la figura 4.15 se muestra la situación en la que, al finalizar el slot, no se han recibido identificaciones, dando comienzo un nuevo slot.

process Reader\_ciclo

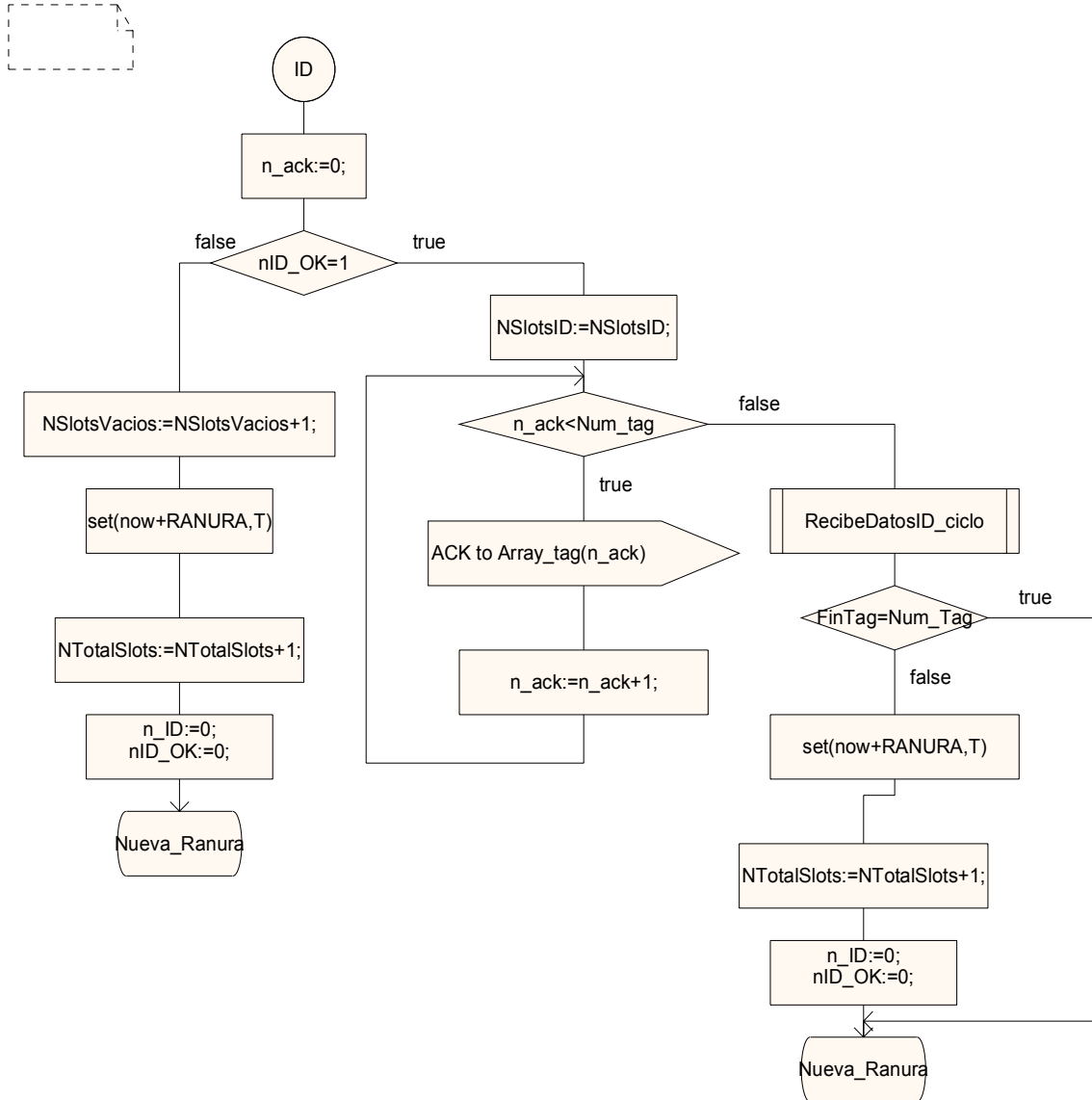


Figura 4.15: Una o ninguna identificación correcta (ciclo a ciclo)

#### 4.5.2.2.2 Hay una identificación correcta

Si hay una identificación correcta (figura 4.15) se le comunica a los tags, se espera recibir los datos del tag que había enviado su identificación y comprueba si los datos son correctos o no en el procedimiento RecibeDatosID\_ciclo (similar al detallado en el apartado 4.5.1.2.2).

### 4.5.2.2.3 Hay más de una identificación correcta

Contexto en el que se ha producido una colisión y comienza un Ciclo de Identificación, figura 4.16. Se incrementa el contador NslotsColision y a continuación, se le notifica la situación a todos los tag mediante la señal Query (Q).

process Reader\_ciclo

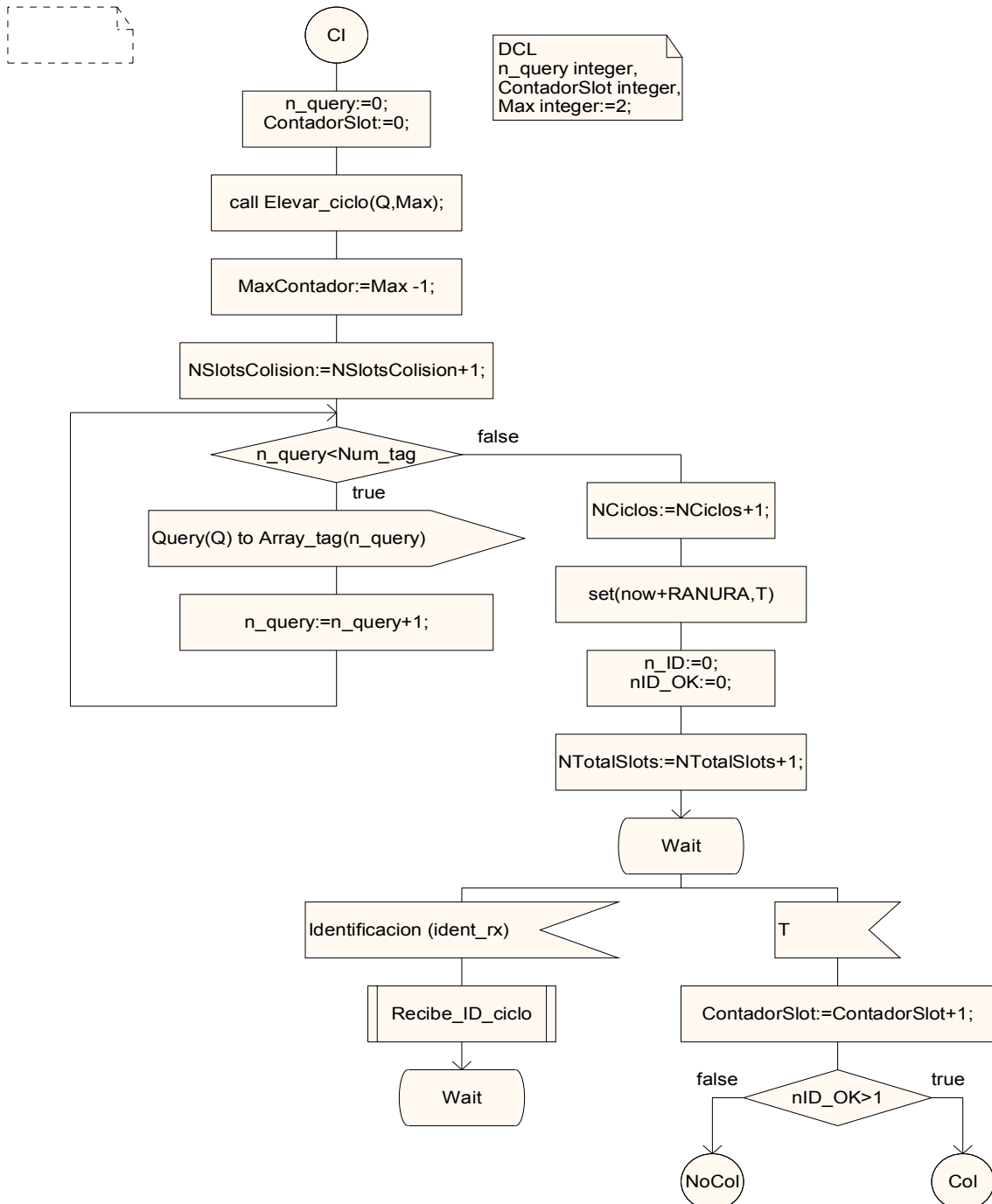


Figura 4.16: Varias identificaciones correctas (ciclo a ciclo)

El valor del número de slots del ciclo será  $2^Q - 1$ , y se almacena en la variable MaxContador. Posteriormente, el Reader permanecerá a la espera en el estado Wait hasta recibir una señal Identificación o una señal T.

#### 4.5.2.2.3.1 Identificación

Se evaluará si la identificación es correcta o no mediante el procedimiento Recibe\_ID\_ciclo y al finalizar este vuelve al estado Wait (figura 4.16)

#### 4.5.2.2.3.2 Temporizador

Al recibir una señal T (figura 4.16) se incrementa el contador de slots, se evalúa el número de identificaciones recibidas y se comprueba para cada caso si se ha llegado al número máximo de slots que contiene el ciclo.

##### 4.5.2.2.3.2.1 No hay identificaciones correctas y no se ha alcanzado el número máximo de slots en el ciclo

Simplemente se incrementa el contador de slots y se le comunica a los tags el comienzo de un nuevo slot mediante QueryRep (figura 4.17).

##### 4.5.2.2.3.2.2 Hay una identificación correcta y no se ha alcanzado el número máximo de slots en el ciclo

Según la figura 4.17, se incrementa el contador NSlotID, se notifica a los tags que el identificador es correcto (ACK), y se espera la recepción de los datos del tag identificado para posteriormente analizar si son correctos o no en el procedimiento RecibeDatosCI\_ciclo (similar al detallado en la sección 4.5.1.2.3.2.2 e ilustrado en las figuras 4.11a y 4.11b).

##### 4.5.2.2.3.2.3 Hay una o ninguna identificación y el número de slots es mayor que el posible en este ciclo

En este caso, como se puede apreciar en la figura 4.17, se ejecuta el procedimiento AjusteQ\_ciclo. Este algoritmo, representado en la figura 4.18, se emplea para minimizar el tiempo medio de identificación necesario. Se calcula un valor óptimo de la Q logrando maximizar el número de identificaciones por ciclo. El valor de Q será el valor redondeado que tenga Q<sub>fp</sub> al finalizar el procedimiento. Inicialmente Q<sub>fp</sub> valdrá 4.0, y su valor irá modificando dependiendo de si en el slot se ha registrado más de un identificador, o no se ha registrado identificador alguno.

- Si no hay identificador,  $Q_{fp} = \max(0, Q_{fp} - C)$
- Si hay más de un identificador,  $Q_{fp} = \min(0, Q_{fp} + C)$

Donde C valdrá 0.2 si Q es mayor de 5 o 0.4 si el valor de Q es menor de 5. Las variables Qaux, Qfp2 y Q2 son variables auxiliares empleadas para poder realizar los cálculos. La función fix se emplea para obtener la parte entera de una cifra.

Una vez realizado el ajuste de la Q, se envía dicho valor a los tags mediante la señal QueryAdjust, y comienza un nuevo ciclo (figura 4.19 )



process Reader\_ciclo

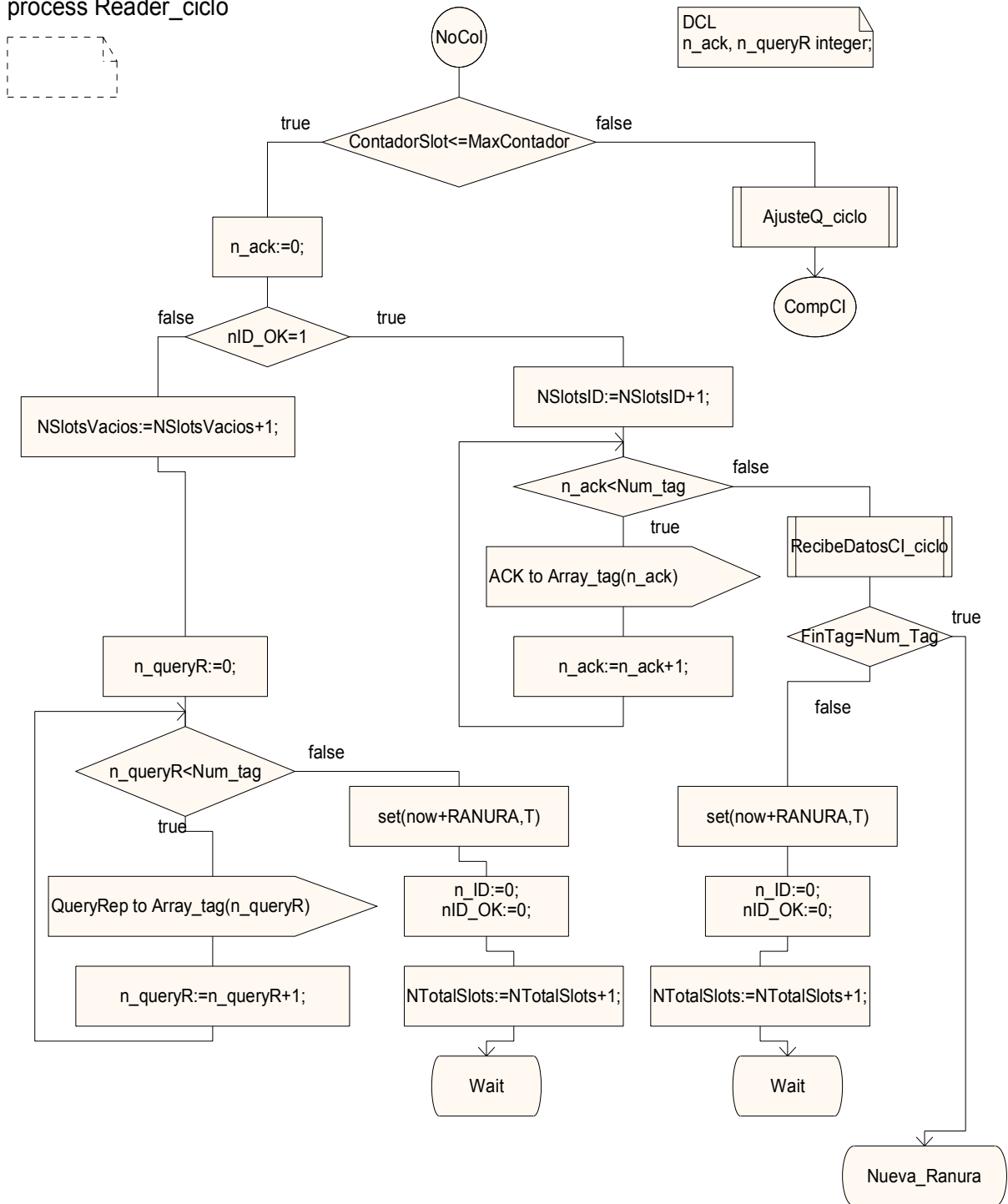


Figura 4.17: Esquema para una o ninguna identificación correcta y con número de slots dentro del rango permitido en el ciclo (ciclo a ciclo)

procedure AjusteQ\_ciclo

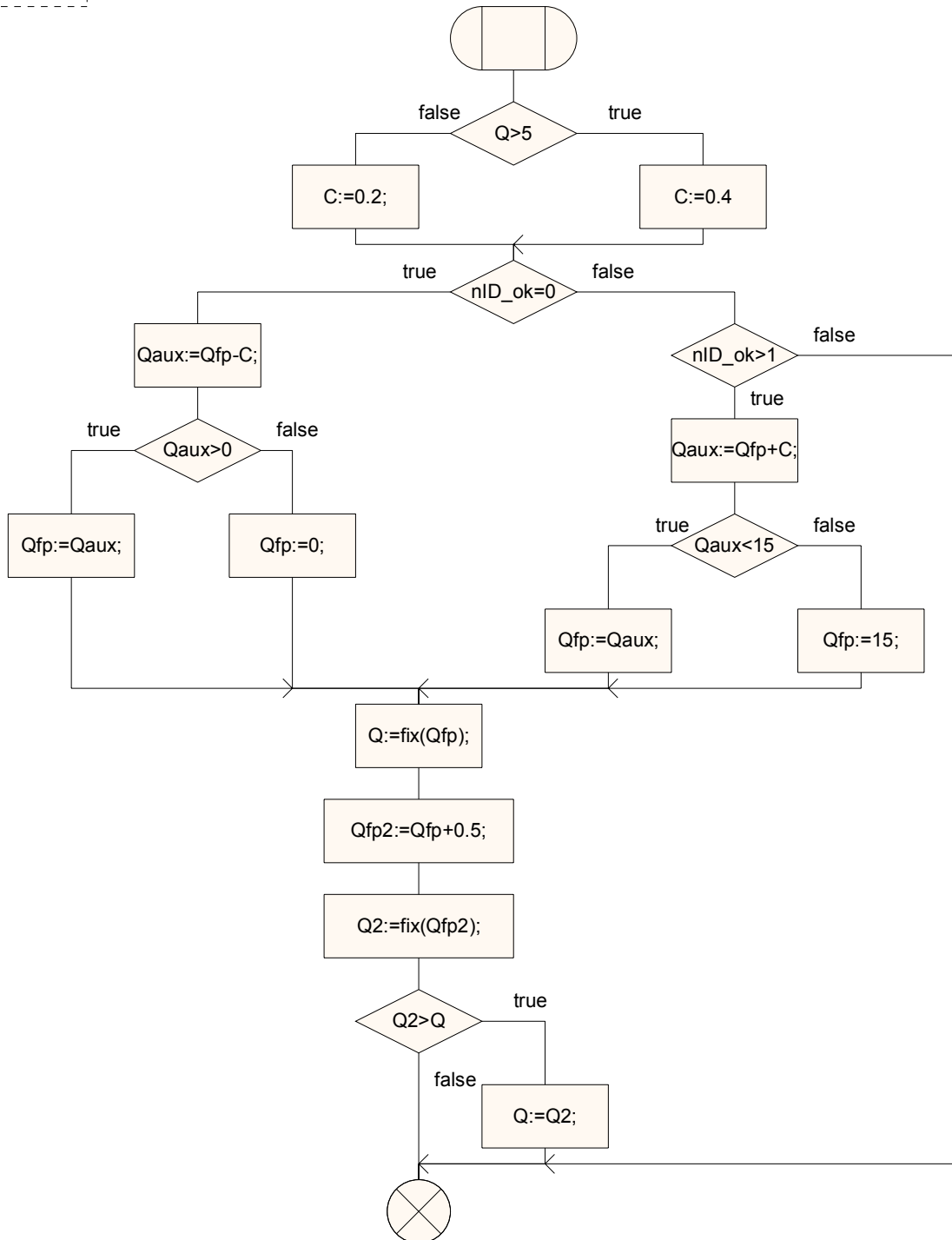


Figura 4.18: Algoritmo de Ajuste de la Q (ciclo a ciclo)

process Reader\_ciclo

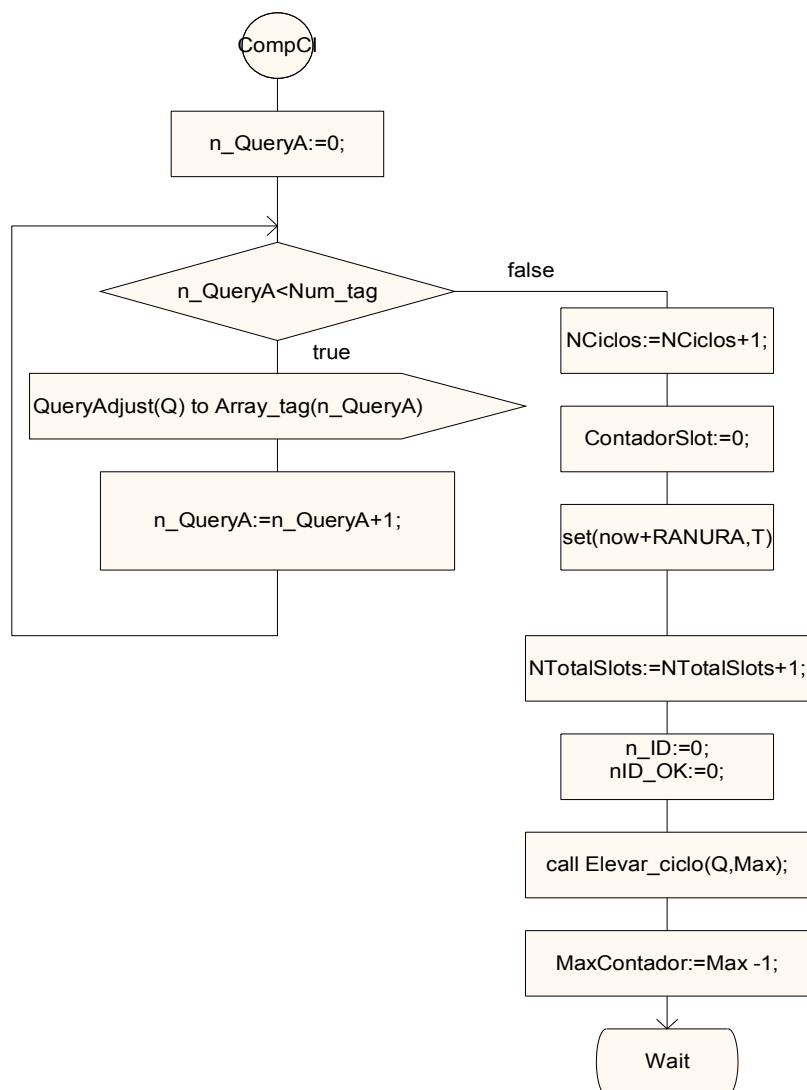


Figura 4.19: Nuevo ciclo (ciclo a ciclo)

#### 4.5.2.2.3.2.4 Hay varias identificaciones y no se ha alcanzado el número máximo de slots posibles en este ciclo

Se producirá una nueva colisión y por tanto, se incrementa NSlotsColision, como se observa en la figura 4.20. A continuación se envía QueryRep a todos los tags y comienza un nuevo Slot.

#### 4.5.2.2.3.2.5 Hay varias identificaciones y el número de slots es mayor que el posible en este ciclo

Esta situación, reflejada en la figura 4.19, dará lugar a un nuevo ciclo de igual forma que la descrita en el punto 4.5.2.2.3.2.3.

process Reader\_ciclo

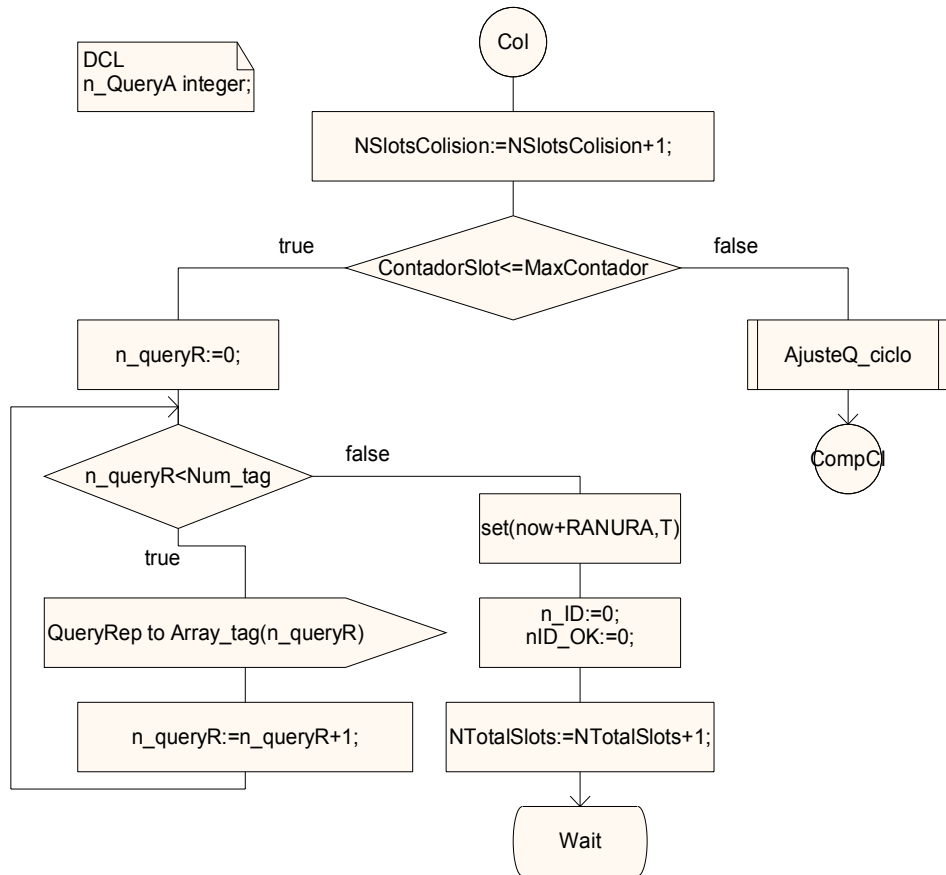


Figura 4.20: Esquema para varias identificaciones correctas y con número de slots dentro del rango permitido en el ciclo (ciclo a ciclo)

### 4.5.3 Reader en escenario con trama adaptativa slot a slot

Este proceso Reader consta de los procedimientos Emitir\_Broadcast, Recibe\_ID, RecibeDatosID y RecibeDatosCI, idénticos a los procedimientos Emitir\_Broadcast\_estatico, Recibe\_ID\_ciclo\_estatico, RecibeDatosID\_estatico y RecibeDatosCI\_estatico, respectivamente, del escenario con trama estática. Incluye también el procedimiento AjusteQ, con diseño similar al procedimiento AjusteQ\_ciclo detallado en el escenario con trama adaptativa ciclo a ciclo. Al igual que en el caso anterior, se muestra el esquema del proceso y se especifican sólo las diferencias con los otros dos escenarios que se han considerado oportunas.

En la figura 4.21 se muestra el comienzo del proceso Reader con la creación de las instancias de proceso tag, el envío de la señal Broadcast a cada uno de ellos y el estado en el que espera una señal:

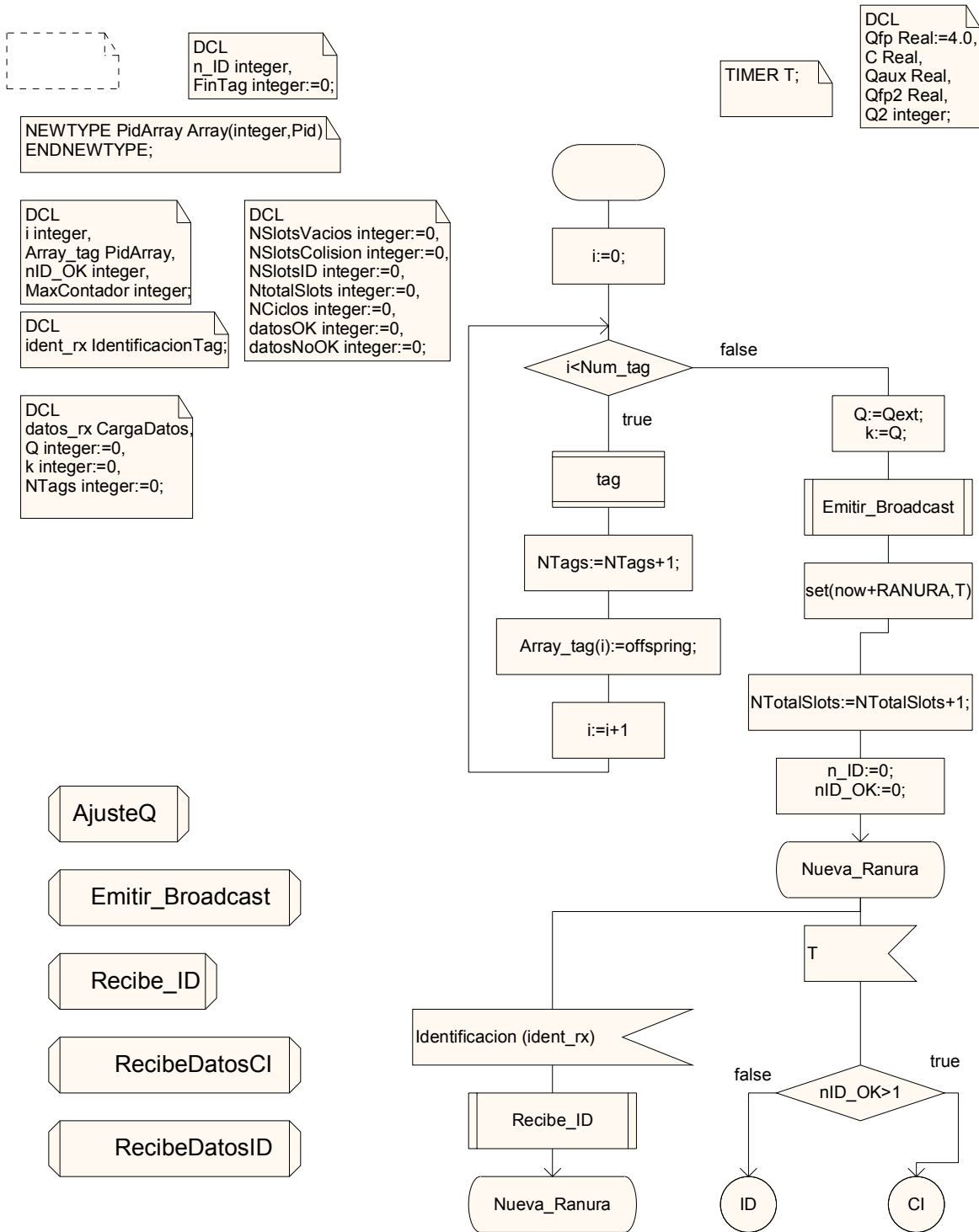


Figura 4.21: Inicio del proceso Reader (slot a slot)

### 4.5.3.1 Se recibe señal Identificación

Se pueden recibir una o varias señales de identificación dentro de un slot temporal (figura 4.21) comprobando si son correctas o no en el procedimiento `Recibe_ID`, idéntico al ilustrado y detallado en la sección 4.5.1.1.

### 4.5.3.2 Se recibe una señal T

Se estudiará el número de identificaciones correctas recibidas.

#### 4.5.3.2.1 No hay identificaciones correctas

Al finalizar un slot, si no se han recibido identificaciones, comienza un nuevo slot tal y como se ilustra en la figura 4.22.

process Reader

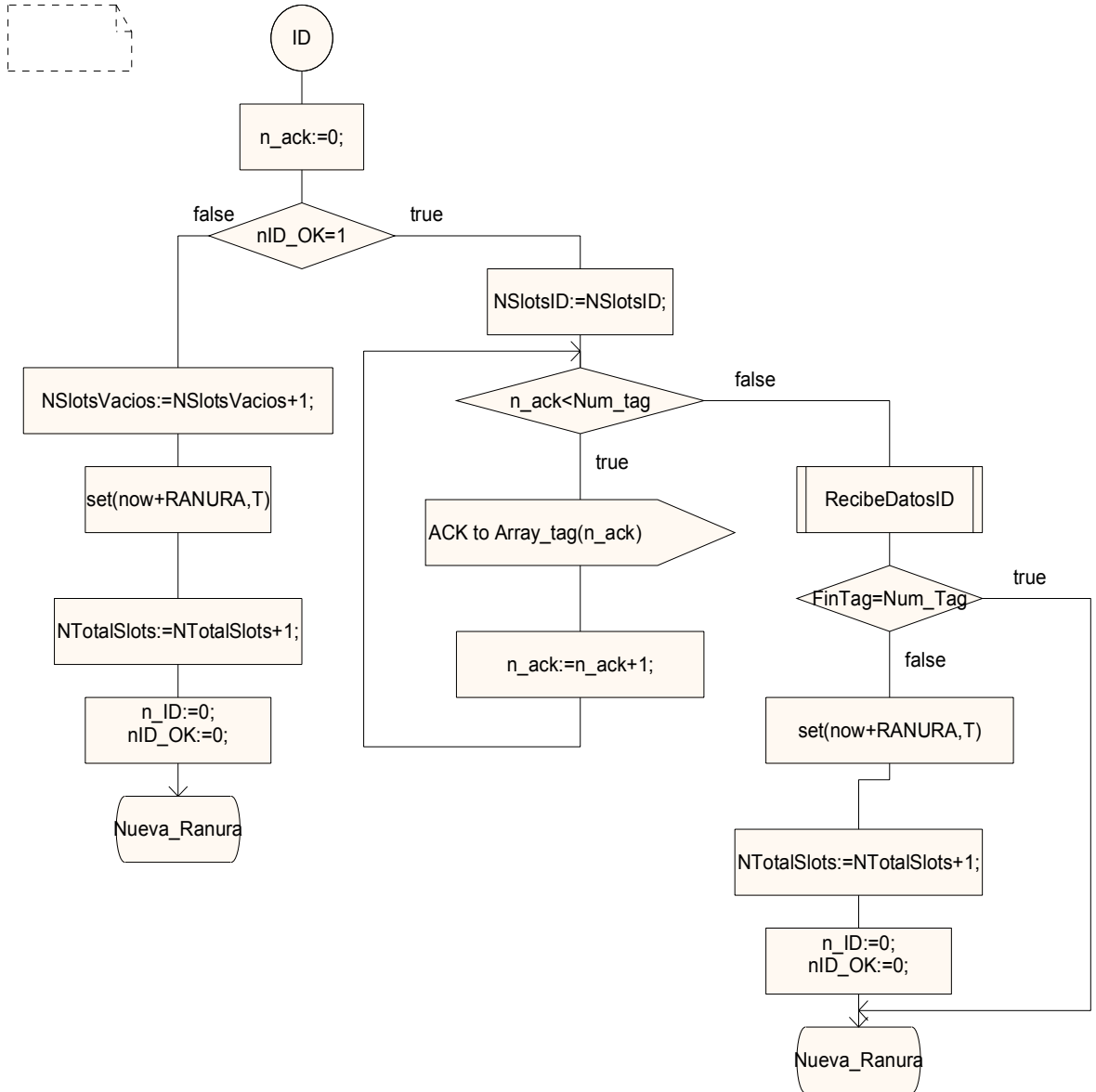


Figura 4.22: Una o ninguna identificación correcta (slot a slot)

#### 4.5.3.2.2 Hay una identificación correcta

Si hay una identificación correcta (figura 4.22) se le comunica a los tags y espera recibir los datos del tag que había enviado su identificación comprobando en el procedimiento RecibeDatosID\_ciclo (similar al detallado en el apartado 4.5.1.2.2) si los datos son correctos o no.

### 4.5.3.2.3 Hay más de una identificación correcta

Comienza un Ciclo de identificación puesto que se ha producido una colisión (figura 4.23) que se le comunicará a los tags mediante la señal Query (Q).

process Reader

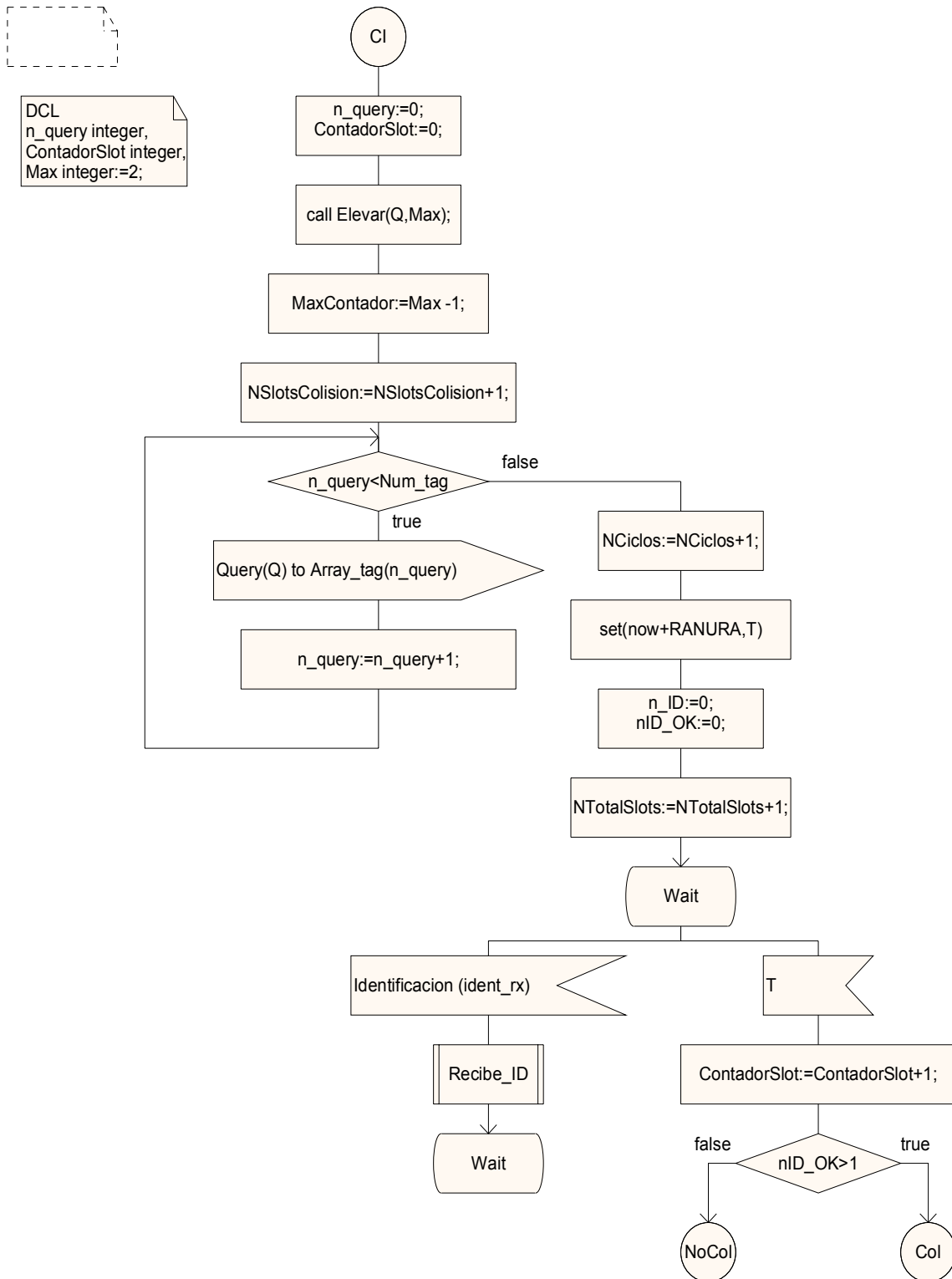


Figura 4.23: Varias identificaciones correctas (slot a slot)

El valor del número de slots del ciclo será  $2^Q - 1$ . Posteriormente, el Reader permanecerá a la espera en el estado Wait hasta recibir una señal Identificación o una señal T.

#### 4.5.3.2.3.1 Identificación

Se evaluará si la identificación es correcta o no mediante el procedimiento Recibe\_ID y al finalizar este vuelve al estado Wait (figura 4.23)

#### 4.5.3.2.3.2 Temporizador

Al expirar el temporizador (figura 4.23) se evalúa el número de identificaciones recibidas y se comprueba para cada caso si se ha llegado al número máximo de slots que contiene el ciclo.

##### 4.5.3.2.3.2.1 No hay identificaciones correctas y no se ha alcanzado el número máximo de slots en el ciclo

Como se aprecia en la figura 4.24, se realiza un reajuste de la trama Q, mediante el procedimiento AjusteQ (detallado e ilustrado en la sección 4.5.1.2.3.2.3). Si el valor de la Q obtenido tras el procedimiento es igual que el valor que tenía antes, se enviará un QueryRep a todos los tags indicando el comienzo de un nuevo slot, mientras que si es distinto comenzará un nuevo ciclo enviándole a los tags mediante la señal QueryAdjust el nuevo valor calculado para la Q (figura 4.25).

##### 4.5.3.2.3.2.2 Hay una identificación correcta y no se ha alcanzado el número máximo de slots en el ciclo

Se le comunica a los tags que el identificador es correcto y espera recibir los datos del tag identificado para comprobar si son correctos o no mediante el procedimiento RecibeDatosCI (similar al detallado en la sección 4.5.1.2.3.2.2 e ilustrado en las figuras 4.11a y 4.11b).

##### 4.5.3.2.3.2.3 Hay una o ninguna identificación y el número de slots es mayor que el posible en este ciclo

Como se ilustra en la figura 4.24, se ejecuta de nuevo el procedimiento AjusteQ. Una vez realizado, se envía el valor de la Q obtenido mediante la señal QueryAdjust (figura 4.25)



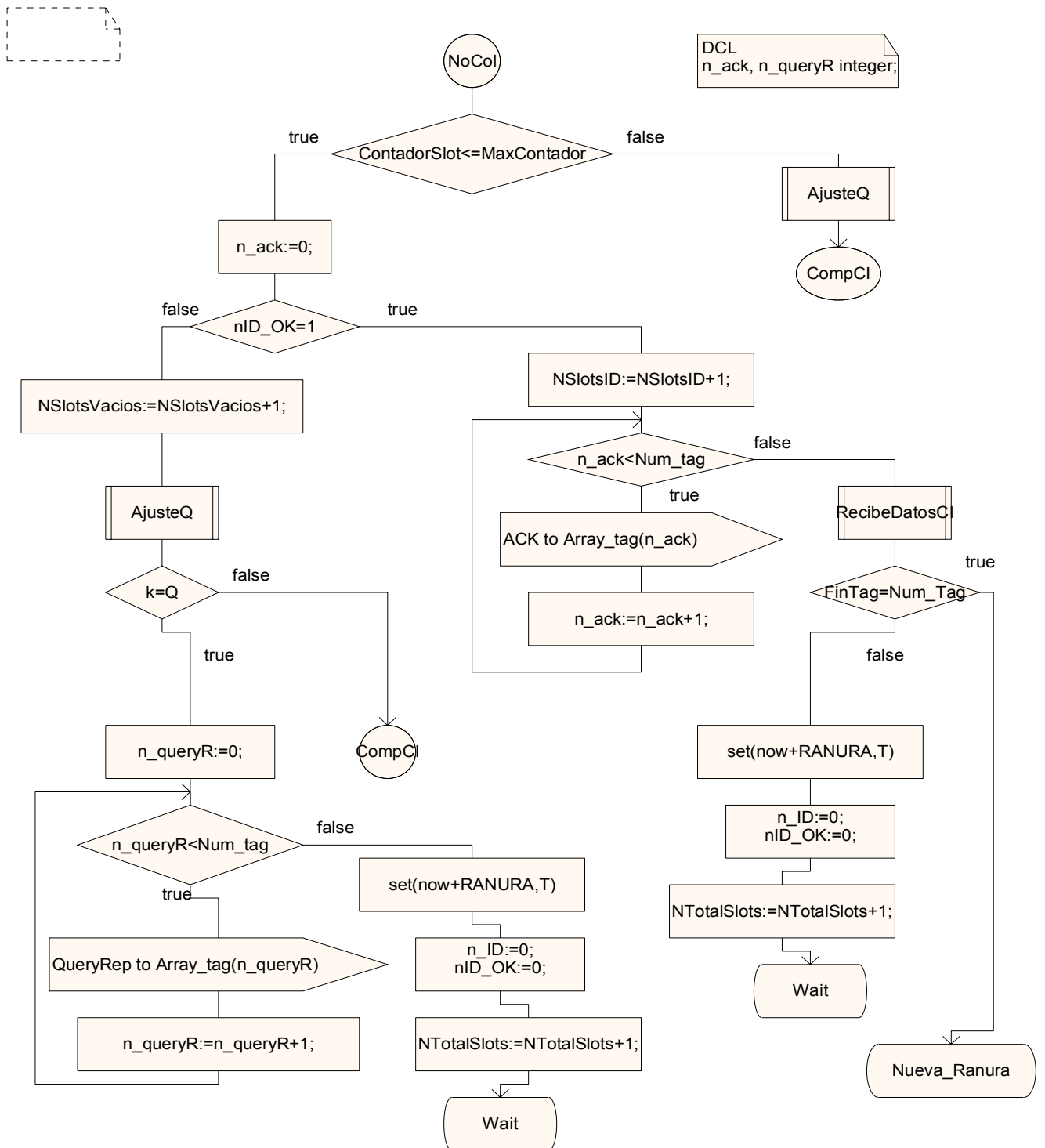


Figura 4.24: Esquema para una o ninguna identificación correcta y con número de slots dentro del rango permitido en el ciclo (slot a slot)

process Reader

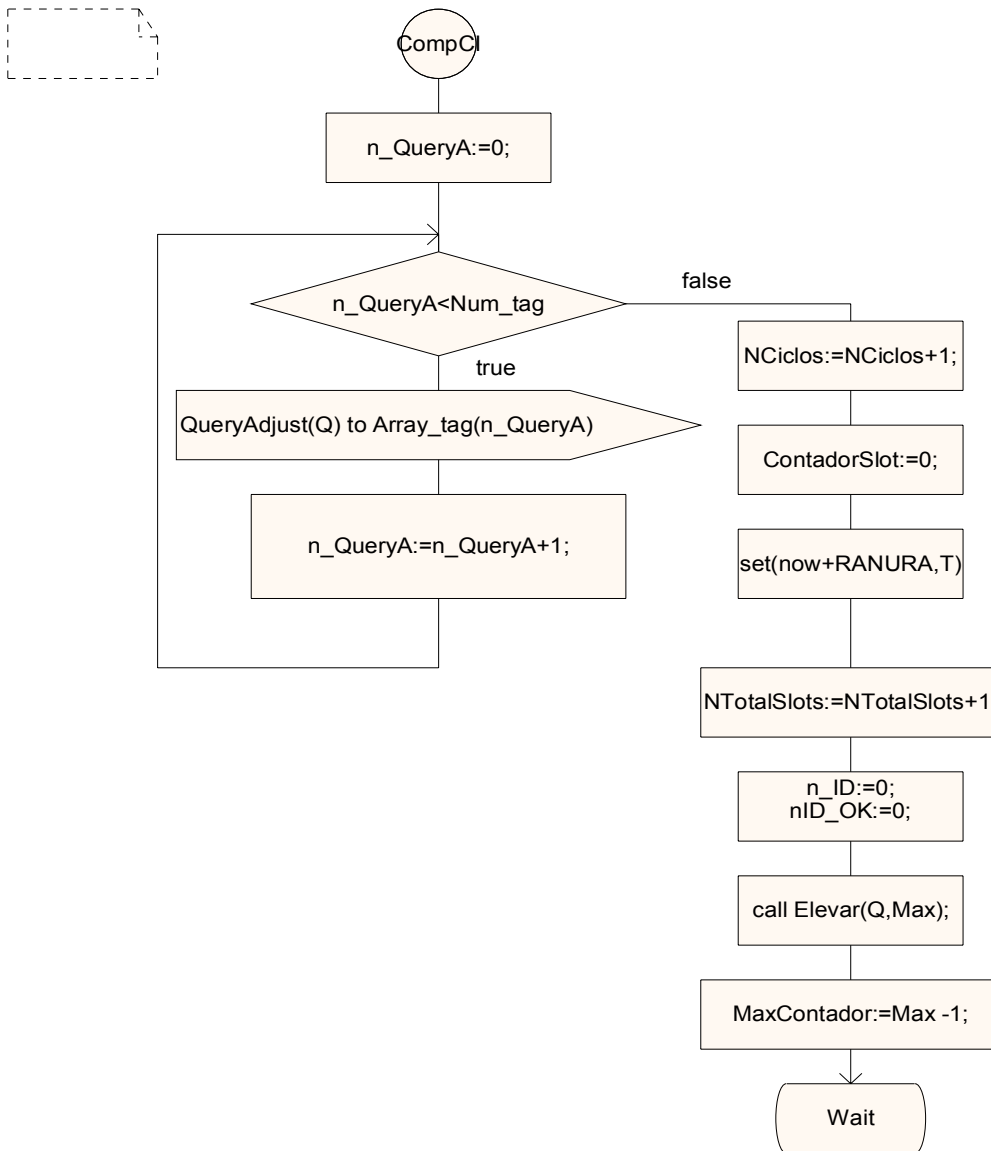


Figura 4.25: Nuevo Ciclo (slot a slot)

#### 4.5.3.2.3.2.4 Hay varias identificaciones y no se ha alcanzado el número máximo de slots posibles en este ciclo

Se producirá una colisión y como se aprecia en la figura 4.26, se realizará un reajuste de la Q. Si la Q obtenida es igual a la que había antes, se envía QueryRep puesto que comienza un nuevo slot, sin embargo, si es distinta, se enviará el nuevo valor de la Q mediante QueryAdjust y comenzará un nuevo ciclo (figura 4.25).

#### 4.5.3.2.3.2.5 Hay varias identificaciones y el número de slots es mayor que el posible en este ciclo

Esta situación, reflejada en la figura 4.16, dará lugar a un nuevo ciclo de igual forma que la descrita en el punto 4.5.1.2.3.2.3.

process Reader

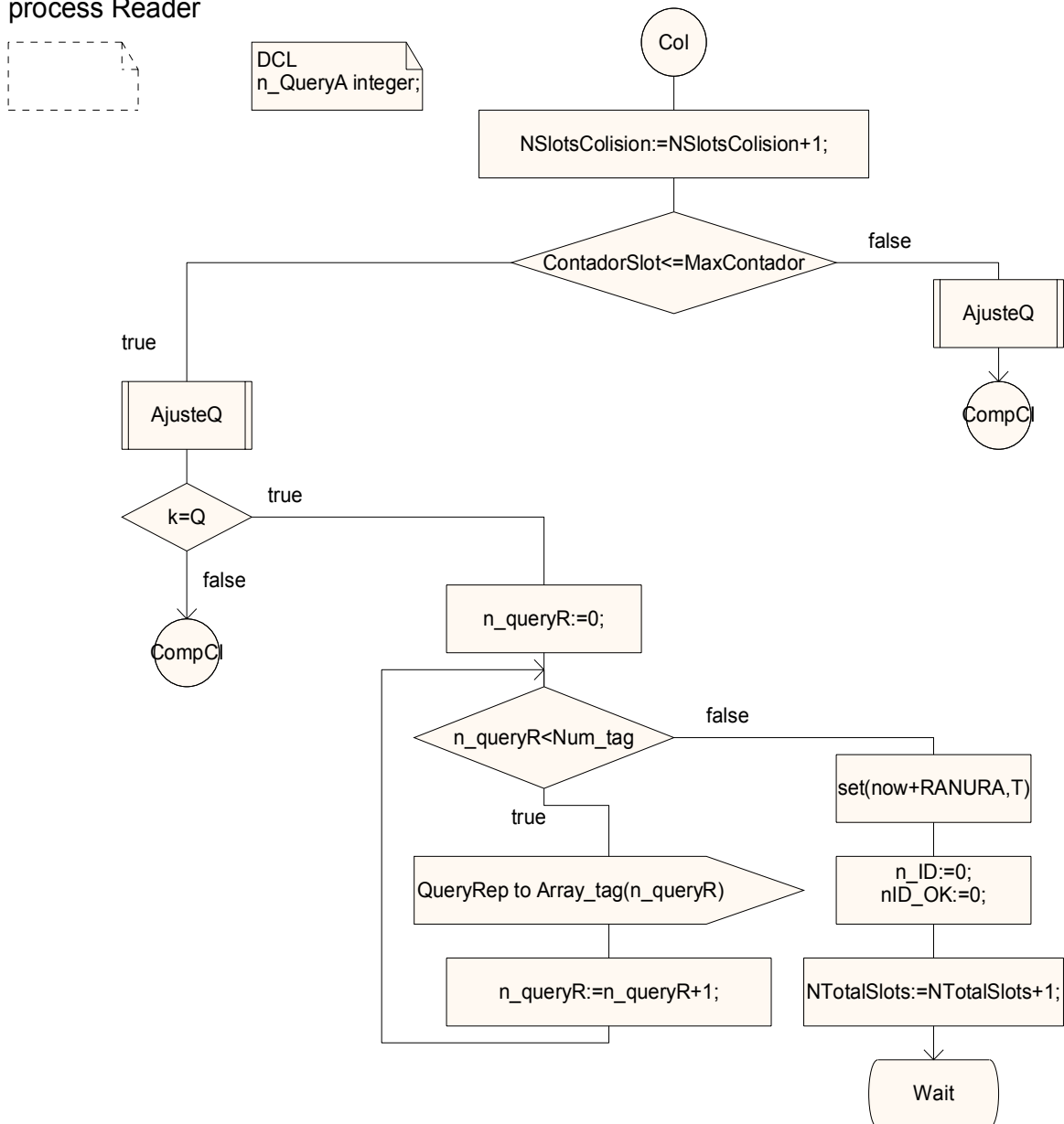


Figura 4.26: Esquema para varias identificaciones correctas y con número de slots dentro del rango permitido en el ciclo (slot a slot)

## 4.6 Diseño del Tag

En el sistema RFID, el proceso tag se comporta de igual forma independientemente del escenario en el que se encuentre, luego es suficiente una única implementación.

Cada tag comienza esperando recibir una señal Broadcast por parte del Reader, tal y como muestra la figura 4.27, al detectar dicha señal, un tag interpreta que tiene que enviar su identificador. Una vez enviado, quedarán a la espera de recibir la confirmación de si el identificador enviado es correcto o no en el estado EsperaConf\_ID.

En este momento existen dos posibilidades (figura 4.28), o sólo hay un tag en la zona de cobertura del Reader y recibirá una señal ACK, o al enviar todos los tag su identificador en el mismo slot, se produce una colisión, circunstancia en la que se recibe una señal Query con un valor de Q.

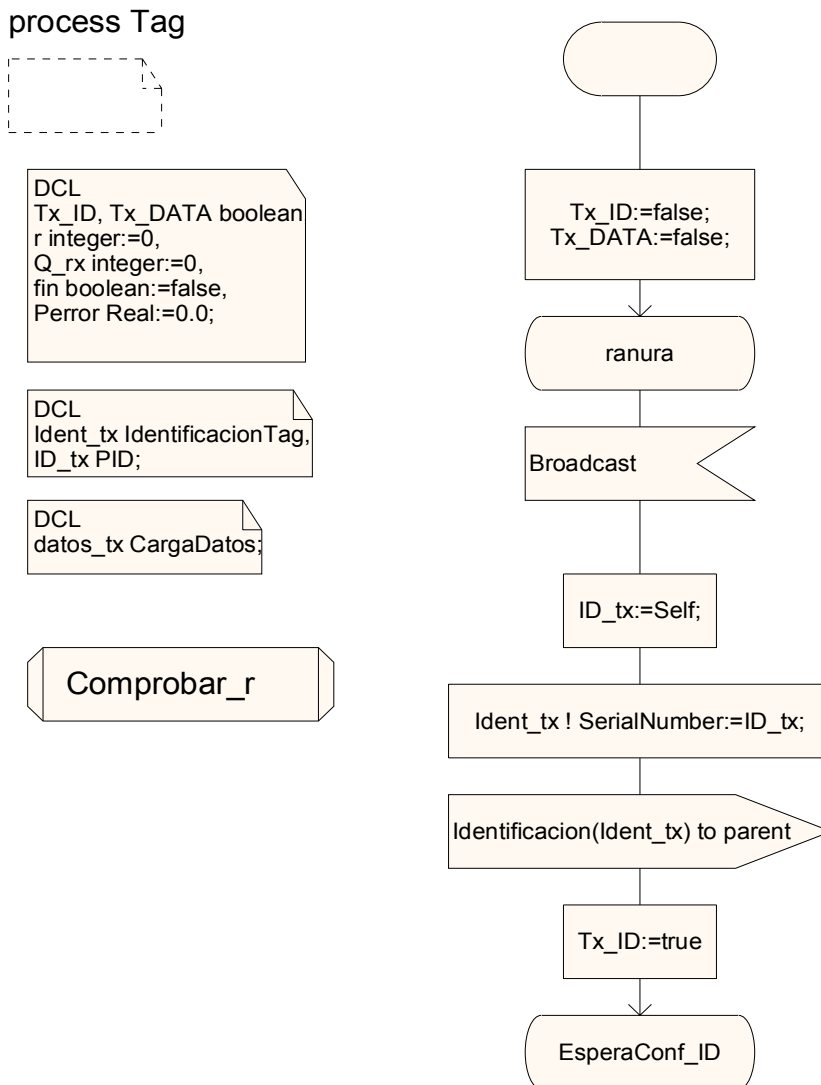


Figura 4.27: Inicio proceso Tag

### 4.6.1 Se recibe ACK

Si recibe una señal ACK el tag que había enviado su identificador, envía ahora sus datos y espera en el estado EsperaConf\_DAT\_ID la respuesta del Reader, pudiendo ser NACK o QueryRep.

Hay dos opciones de datos. A la hora de elegir una, se calcula un número aleatorio entre 0.0 y 1.0, el cual se va a almacenar en la variable Perror, y que simboliza la probabilidad de que se transmitan datos incorrectos. Si Perror es menor que el valor de la probabilidad PEP introducida externamente, se completará con OtroCodigo (datos incorrectos), mientras que si es mayor, se optará por CodigoEPC (datos correctos).

### 4.6.1.1 Se recibe NACK

Al recibir NACK, interpreta que los datos enviados son incorrectos (figura 4.29). El tag que había enviado sus datos trata la variable Tx\_ID y Tx\_DATA para poder volver a apostar por otro slot y vuelve al estado anterior. (Tx\_ID y Tx\_DATA indican que tags han enviado su identificación y sus datos, respectivamente, y cuales no. Así cada vez que enviemos información, pondremos la variable correspondiente a true, por el contrario, cada vez que el tag necesite volver a competir por un slot, se pondrán a false).

### 4.6.1.2 Se recibe QueryRep

En el caso de recibir QueryRep, comienza un nuevo slot, así que cada tag envía de nuevo su identificador.

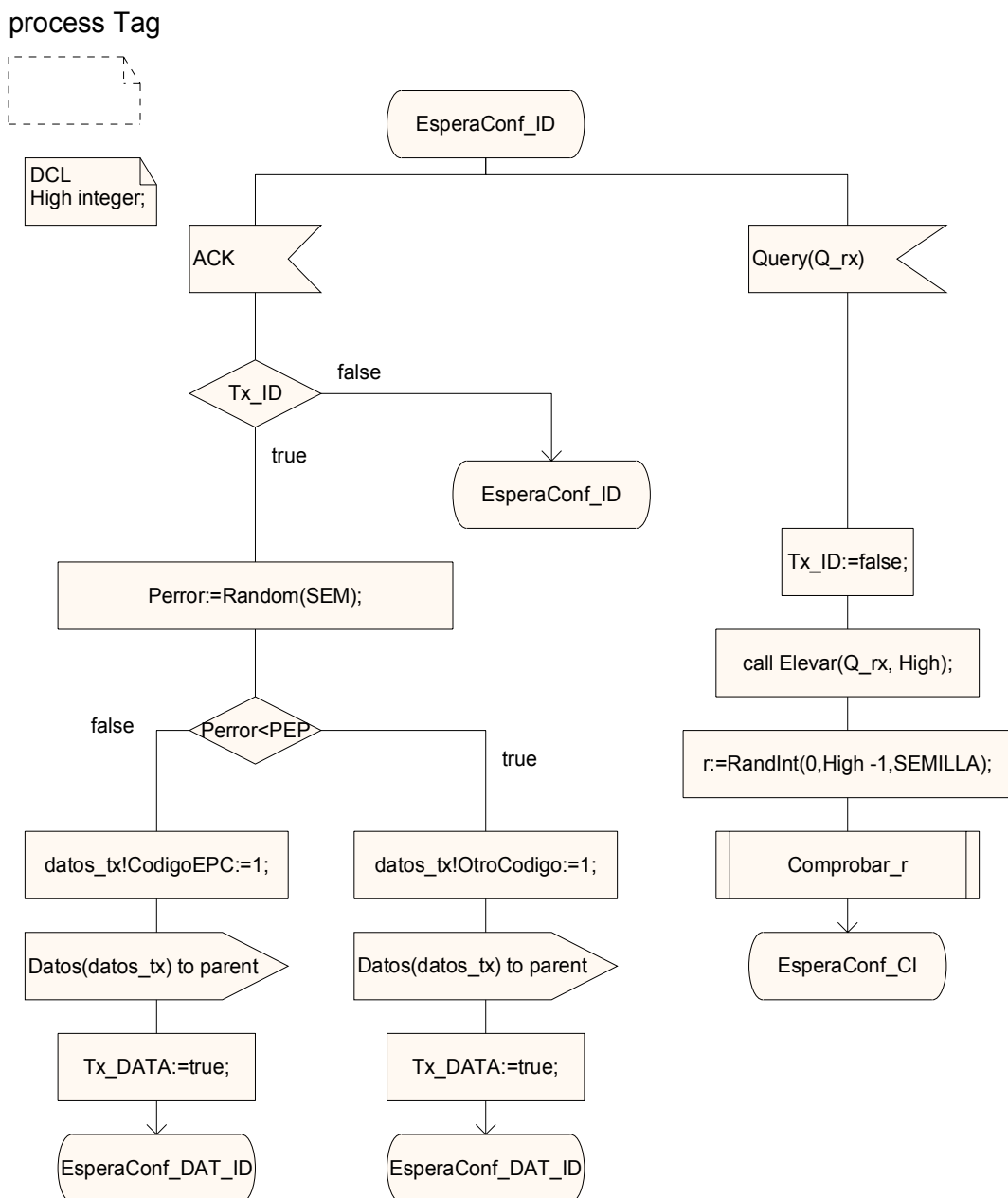


Figura 4.28: Confirmación de Identificación

process Tag

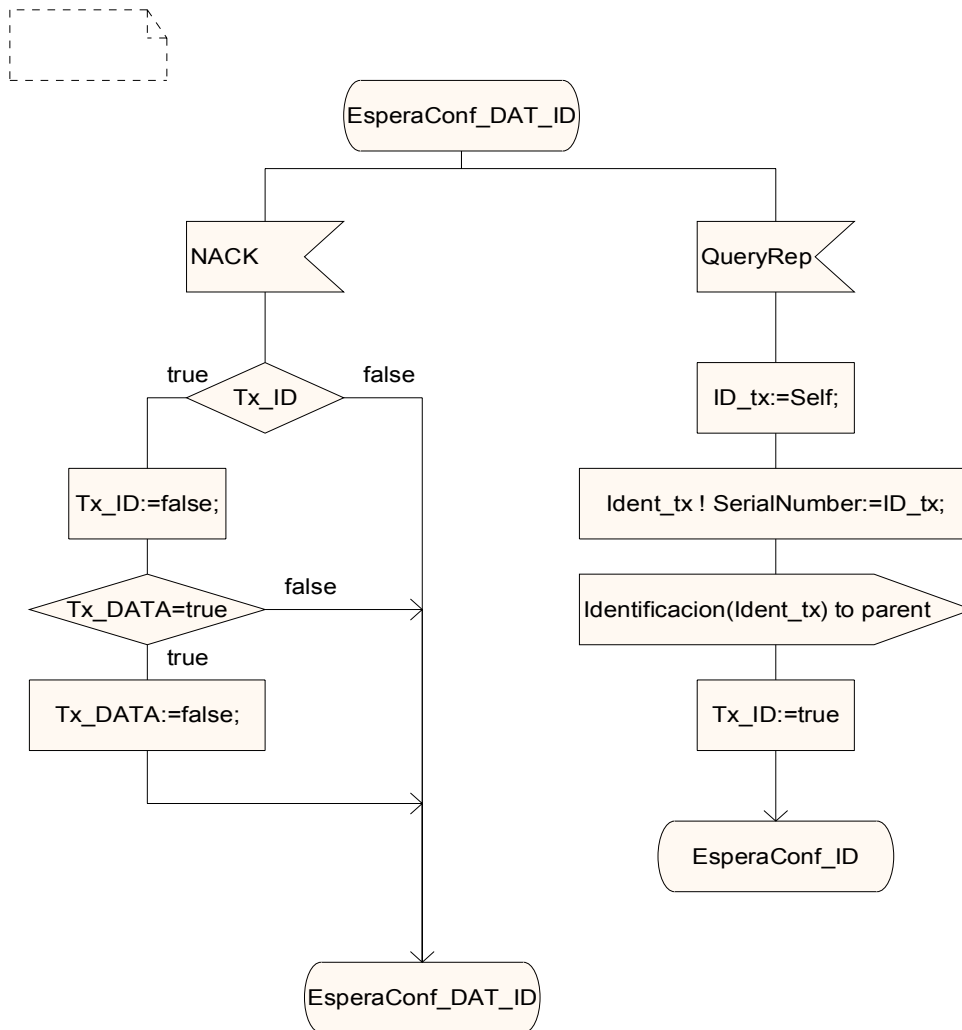
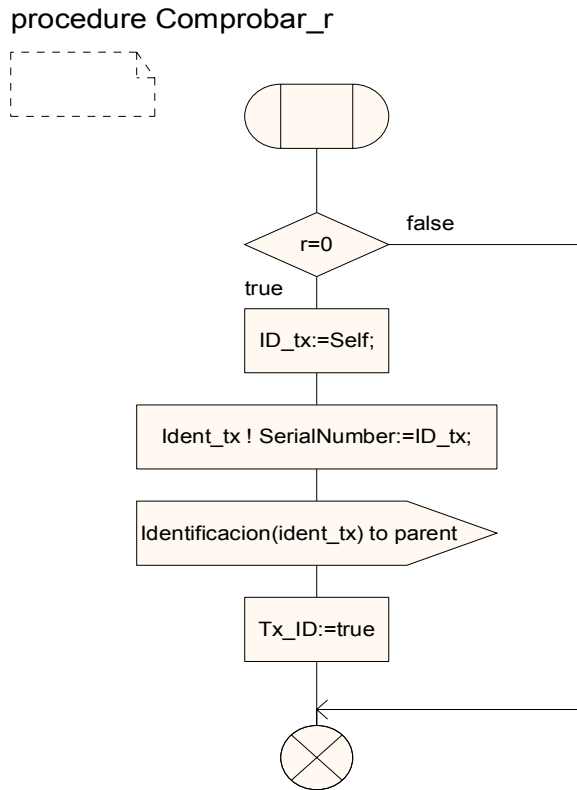


Figura 4.29: Confirmación de datos

## 4.6.2 Se recibe Query

Si se recibe una señal Query (figura 4.28) primeramente se guarda el valor de la Q recibida en la variable Q\_rx y cada tag trata la variable Tx\_ID para poder volver a competir por un slot. A continuación, se almacena en la variable r el resultado que nos devuelve la función Elevar al pasarle como parámetro Q\_rx, que será un valor aleatorio entre 0 y  $2^Q - 1$  (explicada en el apartado anterior). A continuación, se ejecuta el procedimiento Comprobar\_r, en el que se va a determinar cuando un tag tiene que enviar su identificador. Si r es distinta de 0 finaliza el procedimiento sin realizar acción alguna, mientras que si r es igual a 0 el tag envía su identificación (figura 4.30).



4.30: Procedimiento Comprobar\_r

Al finalizar este procedimiento, se vuelve al estado EsperaConf\_CI (figura 4.30.a y 4.30.b ). En este estado, se comprueba el valor de la variable fin, dentro del ciclo de identificación, cada vez que se recibe una señal. Será true cuando el tag ya no compita por un slot, y es necesaria para que el sistema no quede bloqueado.

#### 4.6.2.1 QueryRep

Los tags interpretan que va a comenzar un nuevo slot (figura 4.31a). El tag que había enviado su identificador actualiza el valor de su contador  $r$  a  $2^q - 1$ , consiguiendo así que este tag no vuelva a apostar en este ciclo. Los tags que no habían enviado su identificador, decrementan dicho contador. En ambos casos, y antes de volver al estado anterior, se comprueba el valor de  $r$ .

#### 4.6.2.2 ACK

El tag que había enviado su identificador, ahora envía sus datos y espera en el estado EsperaConf\_DAT (figura 4.32) la respuesta de si los datos enviados son correctos o no, mientras, el resto de datos vuelve al estado de EsperaConf\_CI (figura 4.31a y 4.31b).

##### 4.6.2.2.1 Confirmación de datos. QueryRep

En este caso, el tag que se había identificado y que había enviado sus datos, finalizará y ya no volverá a competir. El resto de tags, decrementan la variable  $r$  y ejecutan Comprobar\_r, volviendo en ambos casos al estado EsperaConf\_CI.

##### 4.6.2.2.1 Confirmación de datos. NACK

Si la señal que se recibe es NACK, el tag que se ha identificado y ha enviado sus datos, actualiza su valor de  $r$  a  $2^q - 1$ , y vuelve al estado anterior.

### 4.6.2.3 QueryAdjust

Al recibir esta señal (figura 4.31b) todos los tags entienden que va a dar comienzo un nuevo ciclo, así que se trata Tx\_ID para permitir que vuelvan a competir, y se calcula un nuevo valor aleatorio para r pasando como parámetro a la función Elevar el nuevo valor Q recibido. Se accede al procedimiento Comprobar\_r y se vuelve al estado EsperaConf\_CI.

### 4.6.2.4 NACK

El tag que había enviado su identificación, actualiza su r a  $2^Q - 1$  y en cualquier caso, se vuelve al estado anterior (figura 4.31b)

process Tag

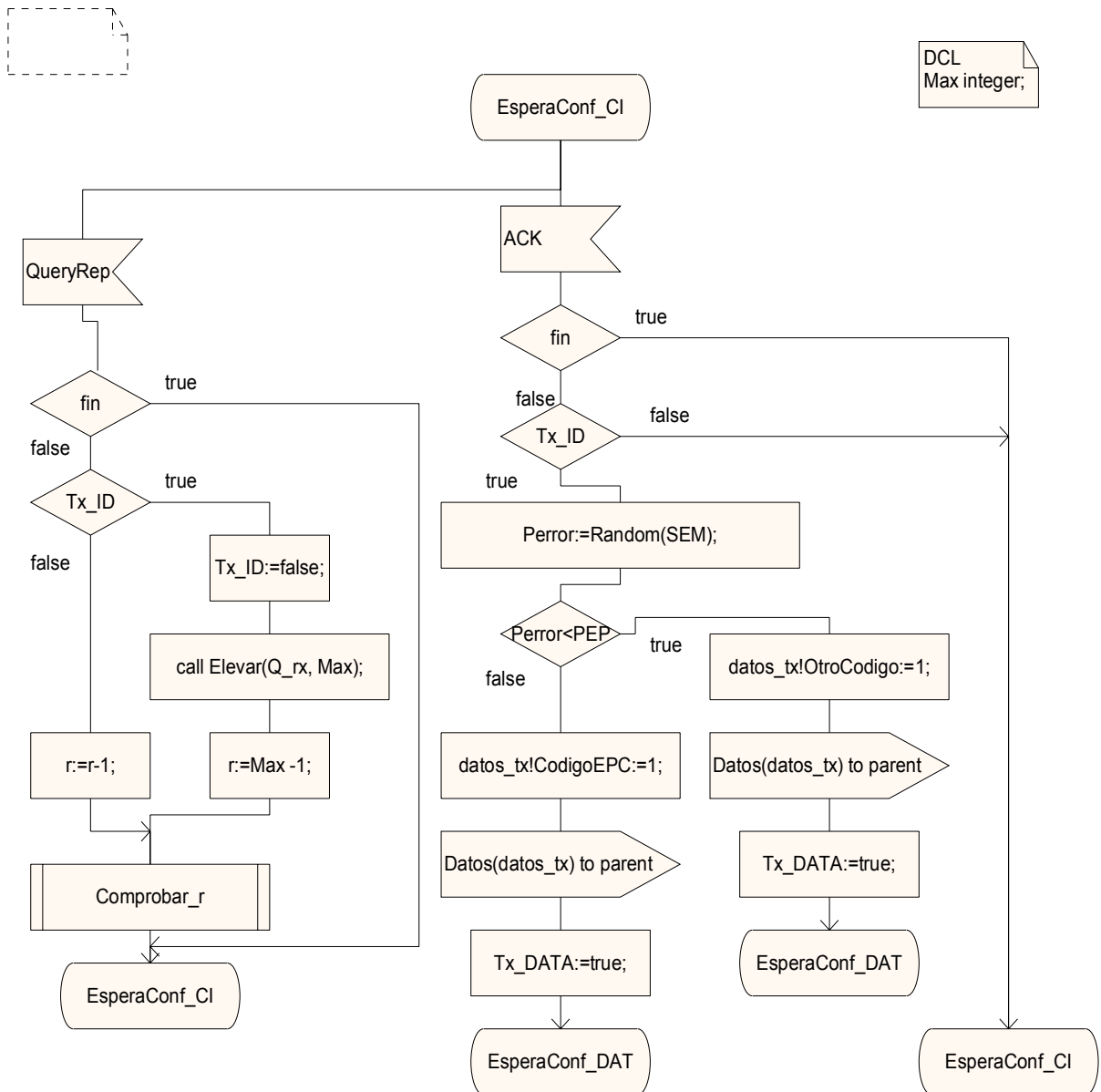


Figura 4.31a: Espera confirmación de identificación dentro del Ciclo de Identificación



process Tag

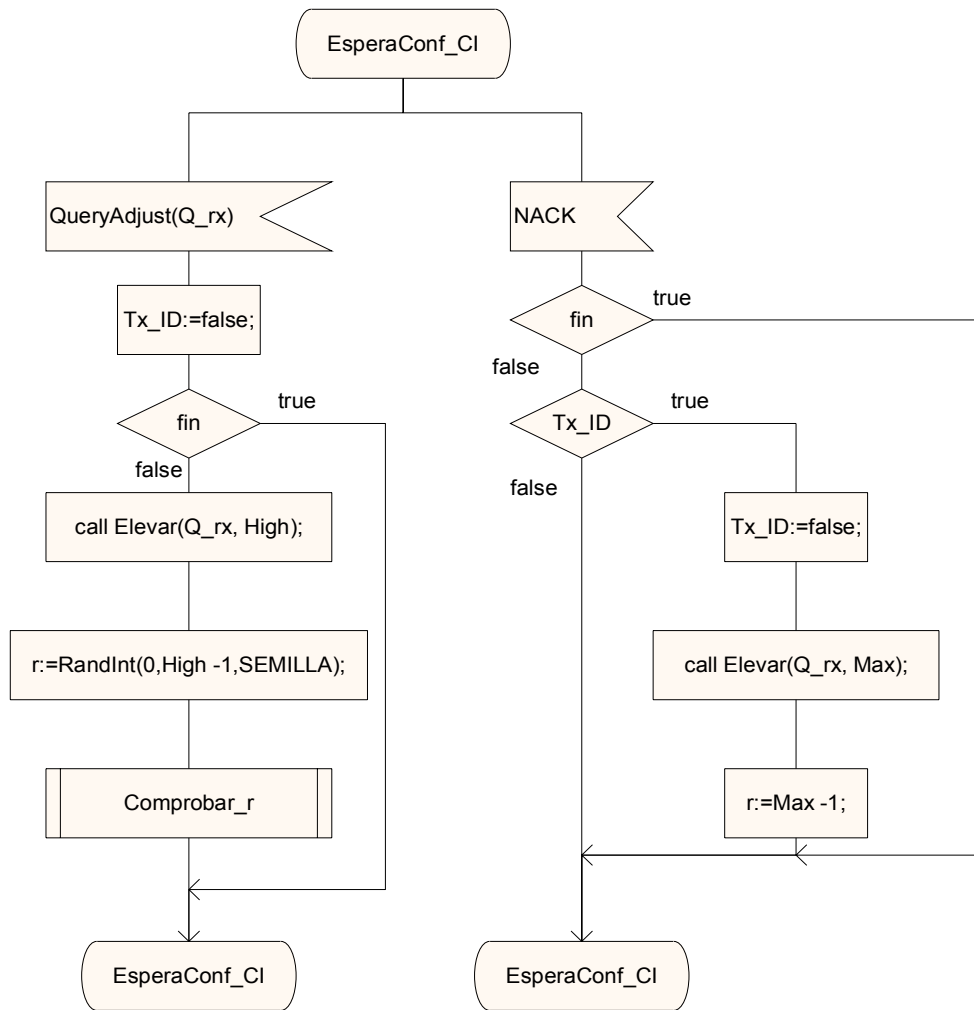


Figura 4.31b: Espera confirmación de identificación dentro del Ciclo de Identificación

process Tag

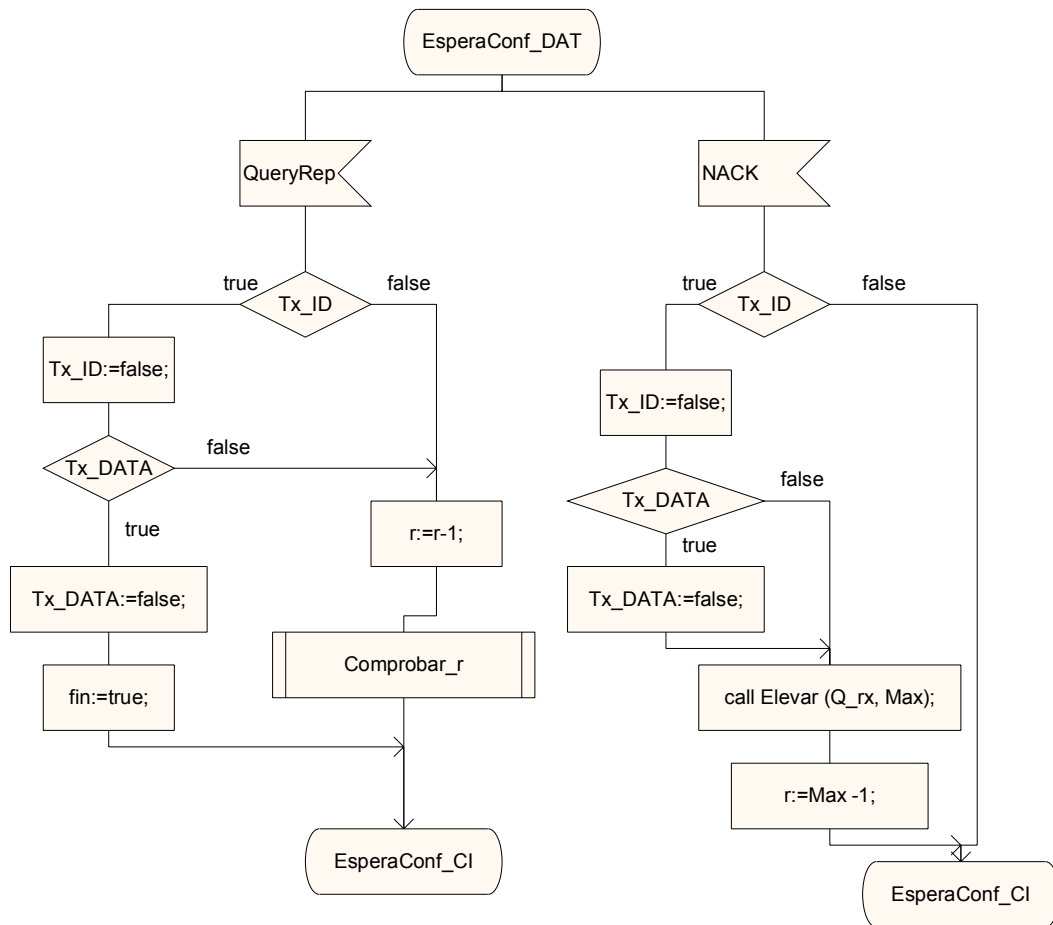


Figura 4.32: Espera confirmación de datos dentro del Ciclo de Identificación

## 4.7 Puesta en marcha del Validador / Simulador

### El validador

El validador comprueba que nuestro sistema es consistente, es decir, que no se producen bloqueos y que todas las señales se consumen. Para ello, mediante distintos métodos de evaluación (Bit-state, Random Walk, Tree Walk...) escanea todos los posibles estados de nuestro sistema buscando cualquier tipo de inconsistencia. Antes de compilar la validación, establecemos el compilador adecuado 'Microsoft Validation'.

Otro aspecto a tener en cuenta en la validación es la longitud de la cola de señales. Por defecto, su valor es 3. Al ejecutar la validación, si nuestro sistema emitiera más de 3 señales sin consumirlas, la cola se llenaría y el validador mostraría un error, así que en nuestra implementación hemos decidido sobredimensionar el número máximo de señales que se pueden encolar y hemos puesto un valor de 1000.

### El simulador

Para realizar la simulación, al igual que con el validador, se selecciona el compilador que vamos a utilizar 'Microsoft Simulation'. Este compilador, genera un

ejecutable (RFID\_smc.exe) que como tal, puede ser invocado desde el shell de MS-DOS. También podemos invocar este ejecutable (.exe) desde la herramienta gráfica que proporciona Telelogic y realizar trazas con MSC

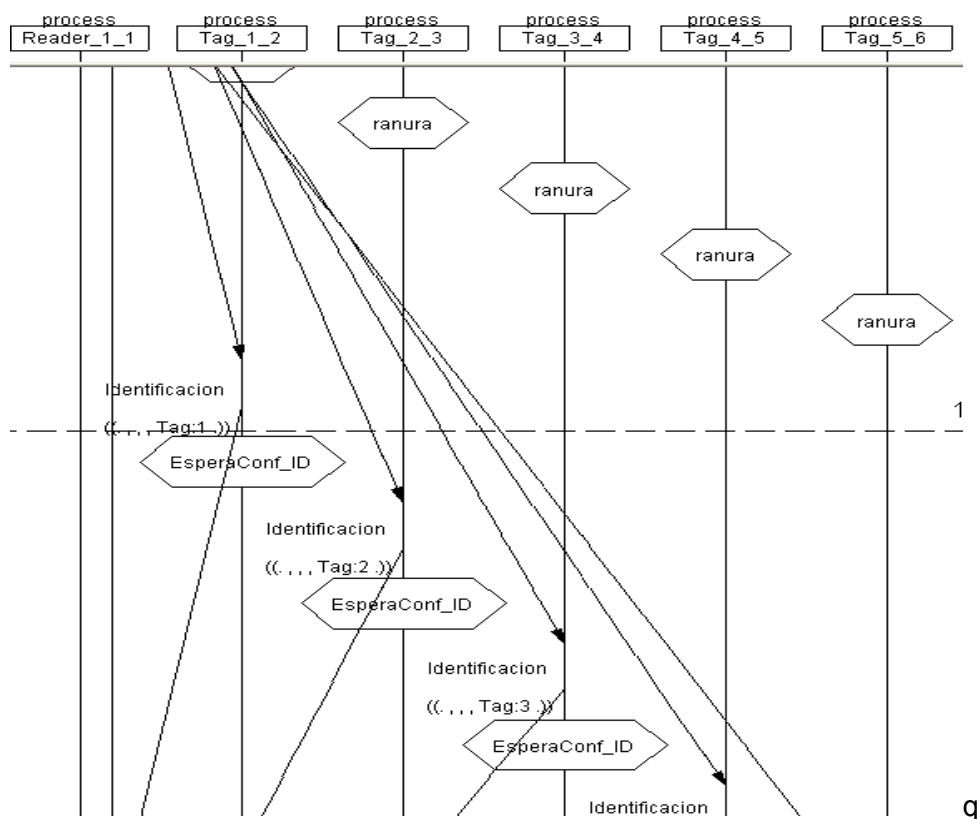


Figura 4.33: Ejemplo de trazas con MSC

Esta última opción es más sencilla, pues proporciona de manera interactiva los comandos a ejecutar. Sin embargo, es menos práctica, sobre todo si queremos ejecutar simulaciones de manera programada (automática).

La forma más práctica de realizar simulaciones de manera automática es escribiendo los comandos a ejecutar en el fichero de texto siminit.com. Este fichero contendrá los comandos a ejecutar en nuestra simulación de forma secuencial.

```

Go

Examine-Variable ( Reader Q)
Examine-Variable ( Reader NTags)
Examine-Variable ( Reader NSlotsVacios)
Examine-Variable ( Reader NSlotsColision)
Examine-Variable ( Reader NSlotsID)
Examine-Variable ( Reader NtotalSlots)
Examine-Variable ( Reader NCiclos)
Examine-Variable ( Reader datosOK)
Examine-Variable ( Reader datosNoOK)
Exit
    
```

Declarar sinónimos externos, nos ayuda a ejecutar varias simulaciones sin necesidad de modificar los valores de nuestras constantes en el código fuente, y por lo tanto, tener que recompilar para realizar una simulación con los nuevos valores. Los pasos a realizar han sido los siguientes:

1. Definir en nuestro sistema los sinónimos de forma externa.
2. Proporcionar esos valores por teclado o fichero. Al ejecutar la simulación, el sistema indicará que proporcionemos el valor de los sinónimos de forma externa, bien desde teclado, o bien desde fichero. Si elegimos introducir el valor por fichero, deberemos tener un fichero con extensión .syn que contenga una lista de sinónimos y sus valores. Por ejemplo:

Qext 4    Num\_tag 20    PEP 0.0

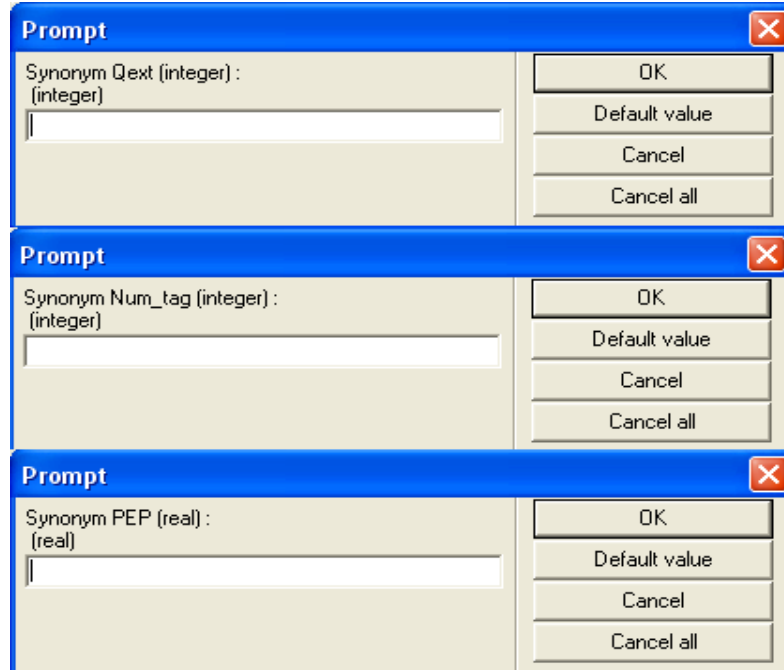


Figura 4.34: Introducción manual de sinónimos

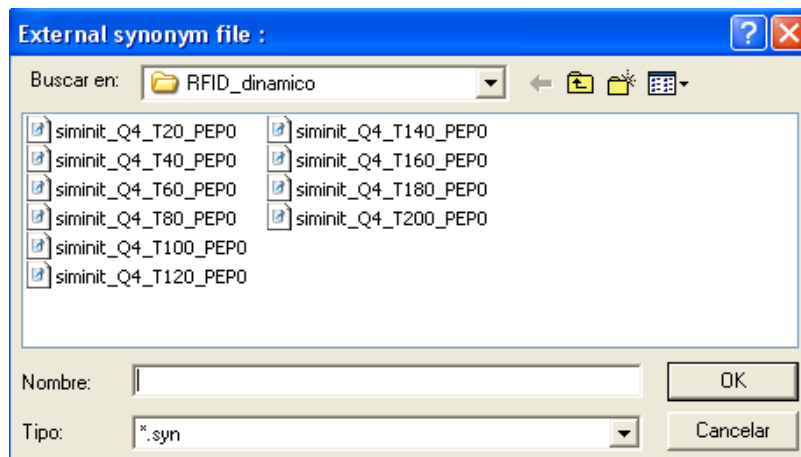


Figura 4.35: Introducción de sinónimos a través de ficheros

Para que nuestro ejecutable (RFID\_smc.exe) sepa que nuestro fichero de sinónimos se llama '.syn', debemos crear la variable de entorno SDTEXTSYNFILE y asignarle como valor el nombre del fichero donde están contenidos los sinónimos. Desde la línea de comandos (desde el cmd.exe) teclearíamos por ejemplo:

```
C:\> set SDTEXTSYNFILE = siminit_Q4_T20_PEP0.SYN
```

```
C:\> RFID_smc.exe > resultado_Q4_T20_PEP0.txt
```

Y el resultado de lanzar el ejecutable, se almacena en un fichero de texto .txt

Para poder automatizar aún más la simulación, creamos un fichero lanzador.bat (del que incluimos un ejemplo), que, tecleándolo en la ventana de MS\_DOS desarrolla todo el trabajo.

```
set SDTEXTSYNFILE=siminit_Q4_T10_PEP0.syn  
RFID_smc.exe > resultados_Q4_T10PEP0.txt
```

```
set SDTEXTSYNFILE=siminit_Q4_T10_PEP0.syn  
RFID_smc.exe > resultados_Q4_T20PEP0.txt
```

```
set SDTEXTSYNFILE=siminit_Q4_T10_PEP0.syn  
RFID_smc.exe > resultados_Q4_T30PEP0.txt
```

```
set SDTEXTSYNFILE=siminit_Q4_T10_PEP0.syn  
RFID_smc.exe > resultados_Q4_T40PEP0.txt
```

```
set SDTEXTSYNFILE=siminit_Q4_T10_PEP0.syn  
RFID_smc.exe > resultados_Q4_T50PEP0.txt
```



# Capítulo 5

## Resultados Obtenidos

### 5.1 Parámetros establecidos

Como se comentó en los capítulos anteriores, se han tenido en cuenta tres escenarios distintos, el primero de ellos atendiendo a una implementación con trama estática, el siguiente según una adaptación de trama ciclo a ciclo, y por último, un escenario con trama adaptativa slot a slot.

Para cada uno de los escenarios se han realizado 10 pruebas en cada punto de simulación (a mayor número de puntos de simulación simulaciones se realizará un mejor ajuste de la media). Además, cada una de estas pruebas se ha realizado para tres probabilidades de error distintas. Se ha medido la probabilidad de error para realizar unas simulaciones más realistas, ya que los sistemas RFID se instalan en entornos con ruido, pérdidas de propagación, difracción de la señal, etc.. lo que puede hacer que no siempre lleguen los datos. As pues, tendremos por ejemplo, para el escenario con trama estática, 10 simulaciones en cada uno de los puntos de simulación atendiendo a una probabilidad de error  $PEP=0$ , otras 10 simulaciones para cada punto de simulación según una probabilidad de error  $PEP=10e-6$  y por último, pruebas similares para una probabilidad de error  $PEP=10e-3$ . Estas simulaciones descritas son idénticas para todos los escenarios, diferenciándose entre sí, en que en la trama estática, los puntos de simulación se han tomado desde 10 a 100 tags con incrementos de 10 en 10, mientras que en los otros dos escenarios se ha considerado un rango de pruebas desde 20 hasta 200 tags con incrementos de 20 en 20. El motivo de tener en cuenta intervalos diferentes se debe a la no adaptación de la trama, lo que provoca en el caso del escenario estático unos tiempos de simulación y una memoria consumida que colapsa los ordenadores disponibles para poder realizar el proyecto.

### 5.2 Resultados obtenidos de trama estática

En las siguientes tablas se muestran los resultados medios obtenidos en el escenario de trama estática al realizar 10 pruebas para cada valor de Ntags, el cual, variará de 10 a 100 (primera columna de cada tabla) e introduciendo un valor inicial de  $Q=4$ . Se puede observar información acerca del estado de los contadores descritos en el diseño.

Q ESTATICA CON PROBABILIDAD DE ERROR $PEP=0$							
Ntags	NSlotsVacios	NSlotsColision	NSlotsID	NtotalSlots	Nciclos	datosOK	datosNoOK
10	24.3	2.9	10	<b>38.9</b>	2.7	10	0
20	23.8	8.3	20	<b>54.4</b>	3.6	20	0
30	42.5	27	30	<b>103.7</b>	6.6	30	0
40	46.9	68.4	40	<b>160.4</b>	10.1	40	0
50	56	127.1	50	<b>242.8</b>	15.5	50	0
60	49.7	216.9	60	<b>332.8</b>	20.9	60	0
70	49.2	411.6	70	<b>537.9</b>	33.9	70	0
80	55.2	701.8	80	<b>845.8</b>	53.2	80	0
90	52.6	1191.9	90	<b>1341.5</b>	84.1	90	0
100	53	1856.8	100	<b>2018.8</b>	126.2	100	0

Tabla 5.1 Resultados de Q estática con PEP=0

Estos nos van a indicar, por ejemplo, cuantos slots y ciclos han sido necesarios en total para que todos los tags se hayan identificado correctamente, así como el número de slots vacíos o con colisión y el número de datos enviados correctos e incorrectos. De los resultados obtenidos se puede deducir el tiempo de identificación, si se conoce la duración temporal de los slots.

Q ESTATICA CON PROBABILIDAD DE ERROR PEP=10e-6							
Ntags	NSlotsVacios	NSlotsColision	NSlotsID	NtotalSlots	Nciclos	datosOK	datosNoOK
10	29.8	3.7	11	<b>46.7</b>	3.2	10	1
20	32	12.3	22	<b>69.3</b>	4.6	20	2
30	33.3	32.5	35.2	<b>107.2</b>	6.9	30	5.2
40	48.5	79.4	43.9	<b>177.3</b>	11.4	40	3.9
50	54.8	148.3	54.5	<b>263.6</b>	16.8	50	4.5
60	52.9	238.1	63.4	<b>361.1</b>	22.8	60	3.4
70	56.8	436.5	74.4	<b>576.1</b>	36.3	70	3.2
80	53.4	724.1	84.6	<b>870.2</b>	54.7	80	4.6
90	65.2	1610.7	103.8	<b>1788.9</b>	112.1	90	5.7
100	54.5	2235.4	104.7	<b>2403.6</b>	150.5	100	4.7

Tabla 5.2 Resultados Q estática con PEP=10e-6

Q ESTATICA CON PROBABILIDAD DE ERROR PEP=10e-3							
Ntags	NSlotsVacios	NSlotsColision	NSlotsID	NtotalSlots	Nciclos	datosOK	datosNoOK
10	85.2	4.7	22	<b>117.9</b>	7.9	10	12
20	82	21	43.3	<b>153.9</b>	10	20	23.3
30	90	57.5	65.2	<b>221.8</b>	14.5	30	35.2
40	96.3	125.1	81.6	<b>313</b>	19.9	40	41.6
50	100.1	269.7	97.4	<b>465.7</b>	19.5	50	47.4
60	116.7	447.1	122.7	<b>700.5</b>	44.2	60	62.7
70	128.5	785.4	144.1	<b>1074.1</b>	67.6	70	74.1
80	134.4	1448.5	166.5	<b>1767.9</b>	110.8	80	86.5
90	107.4	2621.7	181.9	<b>2929.1</b>	183.2	90	92.1
100	170	4405.1	204.3	<b>4802.5</b>	300.5	100	104.3

Tabla 5.3 Resultados Q estática con PEP=10e-3

En la Figura 5.1 se puede apreciar la variación del número total de slots empleados con respecto al número de tag según la probabilidad de error.



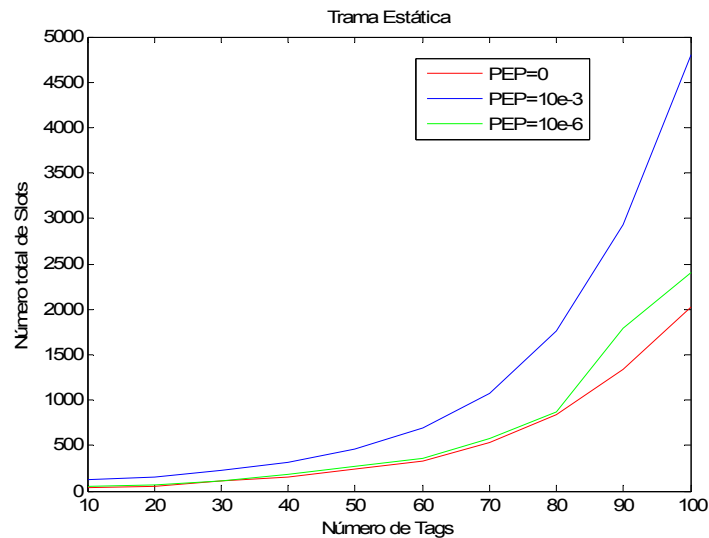


Figura 5.1: Distribución del Numero Total de Slots en trama estática

De la figura anterior se destaca que, como es de esperar, si la PEP=0, la respuesta del sistema es mejor que para cualquier valor de PEP>0. Se comprueba como el aumento de número de tags en la zona de cobertura del lector implica un incremento exponencial en número de slots para la total identificación.

En la Figura 5.2 se observa la variación del número de ciclos empleados con respecto al número de tags diferenciando cada probabilidad de error. En este caso, al estar trabajando con ciclos de tamaño estático, las gráficas de la figura 5.1 y 5.2 son similares.

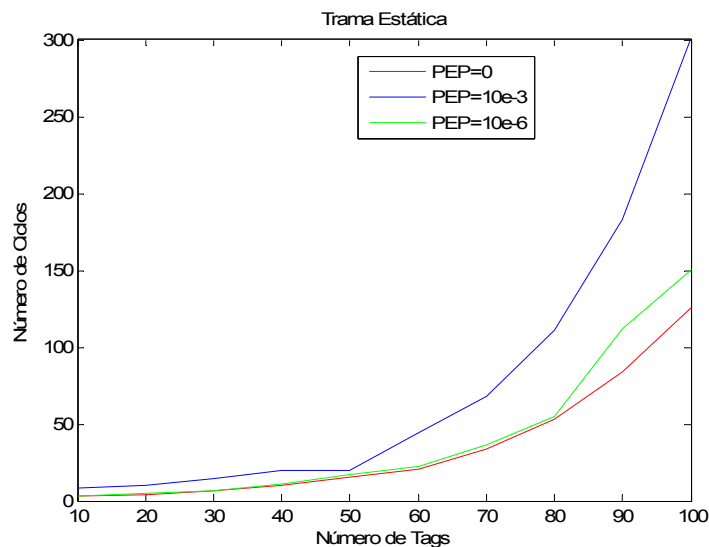


Figura 5.2: Distribución del Número de Ciclos en trama estática

### 5.3 Resultados obtenidos de trama adaptativa ciclo a ciclo

En las siguientes tablas se muestran los resultados medios obtenidos en el escenario de trama adaptativa ciclo a ciclo al realizar 10 pruebas para cada valor de Ntags, el cual, varía de 20 a 200 y con un valor de Q=4.

En estos escenarios, la trama se va modificando ciclo a ciclo, mediante el algoritmo ampliamente explicado en el capítulo anterior.

Q ADAPTATIVA SLOT A SLOT CON PROBABILIDAD DE ERROR PEP=0							
Ntags	NSlotsVacios	NSlotsColision	NSlotsID	NtotalSlots	Nciclos	datosOK	datosNoOK
20	27.2	10.2	20	<b>59.8</b>	3.9	20	0
40	68.2	63.5	40	<b>170.7</b>	9.5	40	0
60	77.2	101.7	60	<b>243.2</b>	10	60	0
80	89.8	168.3	80	<b>344.2</b>	12.6	80	0
100	131.9	242.8	100	<b>478.6</b>	14.4	100	0
120	159.8	305.3	120	<b>589.3</b>	15.2	120	0
140	193.7	357.7	140	<b>696.8</b>	17.4	140	0
160	236.9	448.2	160	<b>850.5</b>	17.6	160	0
180	292.1	476.1	180	<b>949</b>	17.6	180	0
200	305.5	538.4	200	<b>1049.5</b>	20.5	200	0

Tabla 5.4: Resultados Q adaptativa ciclo a ciclo con PEP=0

Q ADAPTATIVA CICLO A CICLO CON PROBABILIDAD DE ERROR PEP=10e-6							
Ntags	NSlotsVacios	NSlotsColision	NSlotsID	NtotalSlots	Nciclos	datosOK	datosNoOK
20	40.5	14.6	24.2	<b>83.3</b>	5.6	20	4.2
40	78	63.4	44.2	<b>189.8</b>	9.4	40	4.2
60	85	118.7	64.2	<b>274.2</b>	11.4	60	4.2
80	111	183.4	84.4	<b>384.9</b>	13.4	80	4.4
100	152.5	234.3	104.5	<b>494.9</b>	13.7	100	4.5
120	166.8	299.3	124.9	<b>594.4</b>	14.7	120	4.9
140	256.3	368.7	148.1	<b>777.9</b>	17.9	140	7.1
160	283.4	451.2	166.3	<b>906.2</b>	18	160	6.3
180	288.7	496.7	186.7	<b>977.5</b>	18.2	180	6.7
200	287.6	549.4	207.5	<b>1050.1</b>	18.5	200	7.5

Tabla 5.5: Resultados Q adaptativa ciclo a ciclo con PEP=10e-6

Q ADAPTATIVA CICLO A CICLO CON PROBABILIDAD DE ERROR PEP=10e-3							
Ntags	NSlotsVacios	NSlotsColision	NSlotsID	NtotalSlots	Nciclos	datosOK	datosNoOK
20	87.6	25	41.8	<b>151.1</b>	11.1	20	21.8
40	168.3	84.7	84.4	<b>348</b>	16.6	40	44.4
60	200.7	170.5	122	<b>506.1</b>	20.4	60	62
80	261.5	264.6	165.6	<b>704.2</b>	22.8	80	85.6
100	426.7	318.4	202	<b>958.7</b>	23.6	100	102

120	395	411.1	244.6	<b>1061.3</b>	23.9	120	124.6
140	642.7	462.1	281	<b>1396.3</b>	24.1	140	141
160	625.4	535.5	321.9	<b>1492.6</b>	24.3	160	161.9
180	898.5	572.7	357.5	<b>1841.2</b>	26.9	180	177.5
200	733.2	661.3	362.7	<b>1800.5</b>	27.4	200	192.7

Tabla 5.6: Resultados Q adaptativa ciclo a ciclo con PEP=10e-3

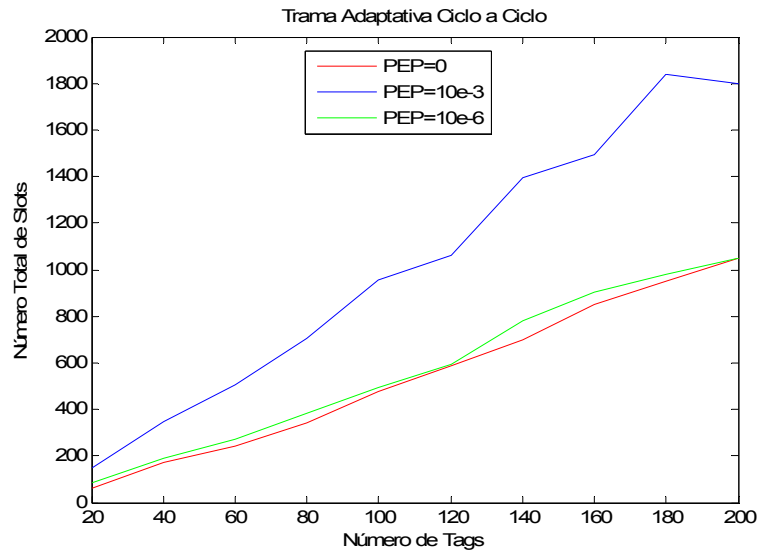


Figura 5.3: Distribución del Numero Total de Slots en trama adaptativa cada ciclo

En la Figura 5.3 se puede apreciar la variación del número total de slots empleados con respecto al número de tag según la probabilidad de error. Como es de esperar, cuanto más restrictivo es la PEP, peores resultados se obtienen (PEP = 10e-3). Si se comparan los resultados de la figura 5.3 con los obtenidos en la figura 5.1, se deduce que, por ejemplo, para 100 tags, el número de slots utilizados para identificar a toda la población de tags es mucho menor si se utiliza el algoritmo de trama adaptativa.

En la Figura 5.4 se observa la variación del número de ciclos empleados con respecto al número de tags según la probabilidad de error. Esta grafica no debe ser de interés para extraer conclusiones para este algoritmo ya que el número de slots por ciclo varía.

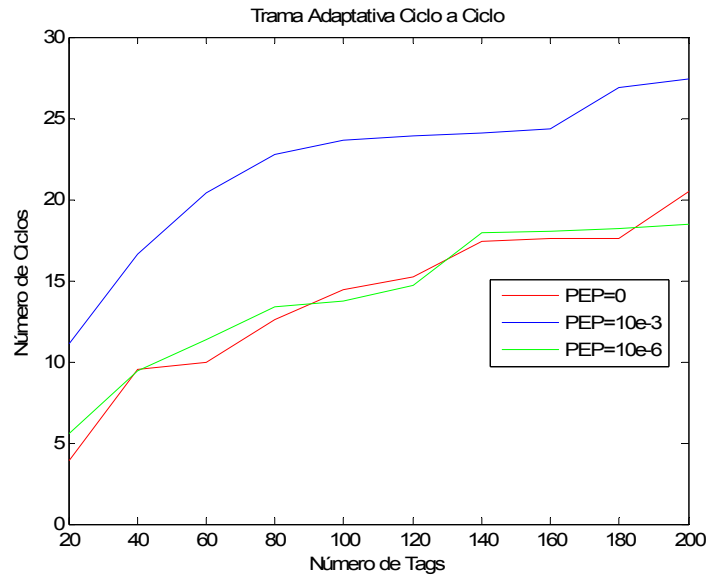


Figura 5.4: Distribución del Número de Ciclos en trama adaptativa ciclo a ciclo

## 5.4 Resultados obtenidos de trama adaptativa slot a slot

Por último, al igual que en los puntos anteriores, se muestran las tablas que contienen los resultados medios obtenidos, en este caso, según el escenario de trama adaptativa slot a slot al realizar 10 pruebas para cada valor de Ntags, el cual, varía de 20 a 200.

Q ADAPTATIVA SLOT A SLOT CON PROBABILIDAD DE ERROR PEP=0							
Ntags	NSlotsVacios	NSlotsColision	NSlotsID	NtotalSlots	Nciclos	datosOK	datosNoOK
20	22.4	13.1	20	<b>56.1</b>	19.1	20	0
40	43.6	34.2	40	<b>112.8</b>	38	40	0
60	57.8	54.1	60	<b>172.8</b>	79	60	0
80	76.1	77.8	80	<b>234.7</b>	122.8	80	0
100	98.8	99	100	<b>278.6</b>	162.2	100	0
120	121.6	125.9	120	<b>368.6</b>	207.8	120	0
140	142.6	145.6	140	<b>429</b>	257.4	140	0
160	157.8	160.1	180	<b>478.8</b>	276.8	160	0
180	179.3	180.2	180	<b>540.5</b>	322.5	180	0
200	208.5	215.5	200	<b>625.2</b>	384.9	200	0

Tabla 5.7: Resultados Q adaptativa slot a slot con PEP=0

Q ADAPTATIVA SLOT A SLOT CON PROBABILIDAD DE ERROR PEP=10e-6							
Ntags	NSlotsVacios	NSlotsColision	NSlotsID	NtotalSlots	Nciclos	datosOK	datosNoOK
20	24.3	15	22.1	<b>62.2</b>	21.1	20	2.1
40	43.7	34.9	41.4	<b>121</b>	38	40	1.4
60	60.7	56.1	61.6	<b>179.3</b>	76.7	60	1.5

80	79.3	79.8	81.7	<b>241.8</b>	124	80	1.6
100	108.9	108	102.2	<b>320.7</b>	178.1	100	2.2
120	107.3	123.6	122.9	<b>366.4</b>	201.5	120	2.9
140	150.9	153.5	143.3	<b>449.1</b>	266.6	140	3.3
160	169.8	172.2	164.2	<b>503.3</b>	304.6	160	4.2
180	189.1	190.4	183.8	<b>563.7</b>	335.4	180	3.6
200	210.5	215.3	304.4	<b>631.5</b>	384.9	200	4.4

Tabla 5.8: Resultados Q adaptativa slot a slot con PEP=10e-6

Q ADAPTATIVA SLOT A SLOT CON PROBABILIDAD DE ERROR PEP=10e-3							
Ntags	NSlotsVacios	NSlotsColision	NSlotsID	NtotalSlots	Nciclos	datosOK	datosNoOK
20	44.1	30.9	42.1	<b>119.7</b>	42.6	20	22.1
40	80	66.5	79	<b>228</b>	71	40	39
60	116	117.3	119.5	<b>355.3</b>	160	60	59.5
80	157.7	163.4	157.5	<b>480.9</b>	235.4	80	77.5
100	192.4	203.8	199.4	<b>599.4</b>	331.3	100	99.4
120	236.2	242.6	242.3	<b>726.8</b>	395.4	120	122.3
140	259.3	269.1	262.7	<b>803.5</b>	459.1	140	132.2
160	303.3	318.8	310.7	<b>935.2</b>	559	160	150.7
180	349.2	353.3	352.8	<b>1058.2</b>	635	180	172.8
200	415.6	423.7	395.1	<b>1236.2</b>	742.5	200	195.1

Tabla 5.9: Resultados Q adaptativa slot a slot con PEP=10e-3

En la Figura 5.5 se comprueba la variación del número total de slots empleados con respecto al número de tag para cada probabilidad de error. Si se comparan los resultados obtenidos en esta gráfica con los obtenidos en la figura 5.3 se comprueba de nuevo que esta adaptación slot a slot tiene un mejor comportamiento (en menor número de slots para la identificación de todos los tags), que en el caso de trama adaptativa ciclo a ciclo y slot a slot.

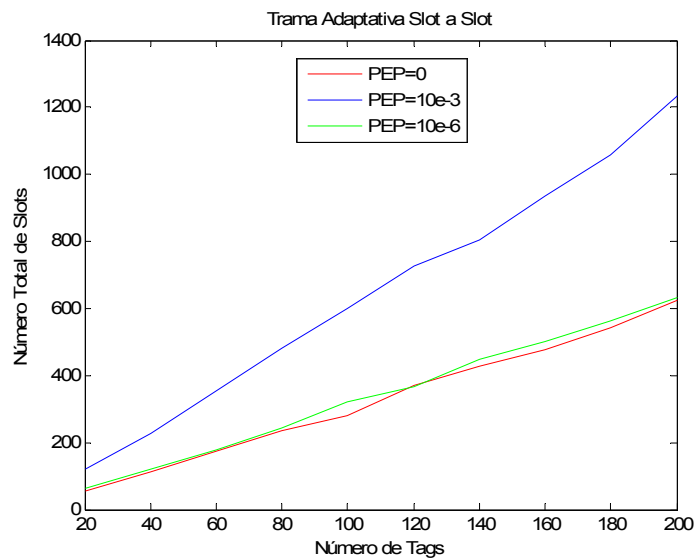


Figura 5.5: Distribución del Numero Total de Slots en trama adaptativa slot a slot

En la Figura 5.6 se aprecia el comportamiento del número de ciclos empleados con respecto al número de tags teniendo en cuenta las distintas probabilidades de error. Como en el caso anterior esta grafica no debe ser de interés para extraer conclusiones para este algoritmo ya que el número de slots por ciclo varía. Además, en este algoritmo slot a slot, los ciclos pueden finalizar en cualquier momento para comenzar uno nuevo, por lo que es obvio que el número de ciclos empleado es mucho mayor.

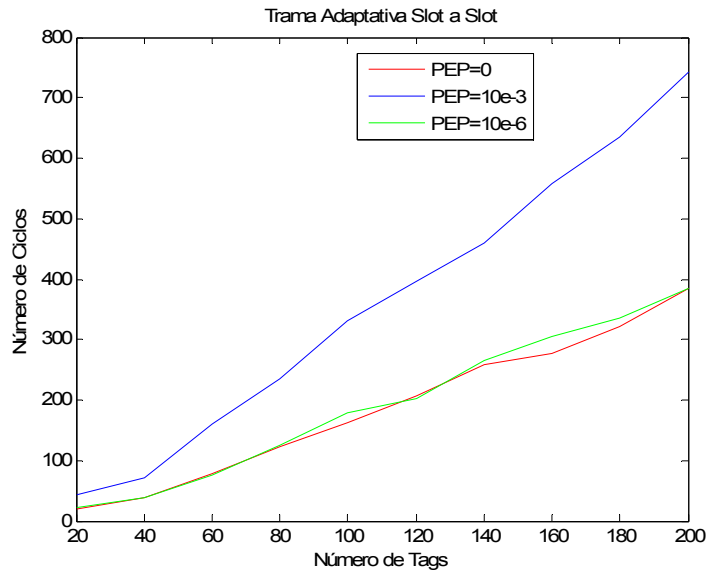


Figura 5.6: Distribución del Número de Ciclos en trama adaptativa slot a slot

# Capítulo 6

## Conclusiones y líneas futuras

---

### 6.1 Conclusiones

En la literatura científica se pueden encontrar numerosos protocolos para sistemas RFID de identificación/anticolisión, cuyo principal objetivo es minimizar el tiempo medio de identificación de los tags. Sin embargo, los sistemas RFID actuales se basan en el estándar mundialmente reconocido y adoptado EPCglobal Class-1 Gen-2. Este estándar, define, sin entrar en detalles, los requisitos hardware y software a seguir en la implantación del mismo, dejando al fabricante e ingeniero la tediosa tarea de deducir o resolver las cuestiones que dicho estándar deja en el aire, como son, por ejemplo, la implementación total del protocolo anti-colisión.

En este proyecto se ha llevado a cabo la validación y simulación del protocolo EPCglobal Class1 Gen2 con el fin de mostrar la respuesta de dicho protocolo ante las posibles alternativas que el fabricante puede tomar a la hora de implementarlo. Se han tenido en cuenta tres configuraciones distintas que se pueden llevar a cabo: trama de contienda estática, trama de contienda adaptativa ciclo a ciclo y trama de contienda adaptativa slot a slot. Los resultados obtenidos ante diversos escenarios (con y sin probabilidad de error) permite deducir cual de las opciones presenta los mejores resultados en tiempo medio de identificación ya que es lo que se persigue minimizar con dichos algoritmos.

De las validaciones y simulaciones realizadas a lo largo de este proyecto hay que destacar la respuesta del algoritmo de adaptación de trama slot a slot frente a los dos algoritmos también estudiados: ciclo a ciclo y trama estática. Por ejemplo, la diferencia en número de slots para una población de 100 tags y PEP = 0 es de más de 200 slots si se compara con el algoritmo ciclo a ciclo y unos 1800 slots comparado con el algoritmo de trama estática. Si la duración de un slot es, según el estándar, de aproximadamente 2.505ms, esta diferencia se traduce a: 0.75 s para identificar a 100 tags con el algoritmo slot a slot frente a 1.25 s para identificar a la misma población con el algoritmo de adaptación de trama ciclo a ciclo y 5.01 s para identificar a la misma población con el algoritmo de trama estática.

Para el caso de PEP >0, el algoritmo de adaptación de trama slot a slot también se presenta como la mejor alternativa frente a los otros dos protocolos, siendo la diferencia temporal entre ellos mucho mayor, como se visualiza en las siguientes gráficas.

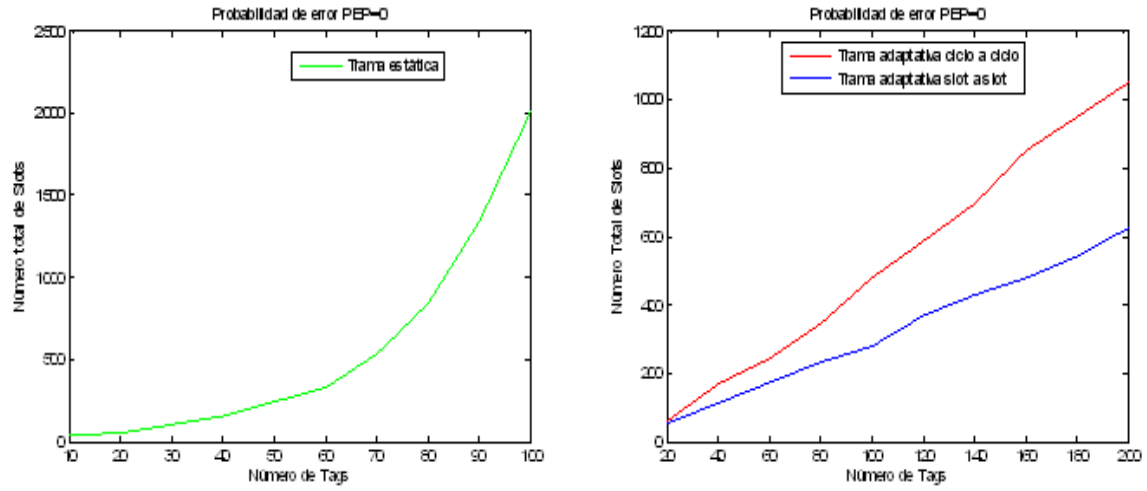


Figura 6.1: Número Total de Slots para una probabilidad PEP=0

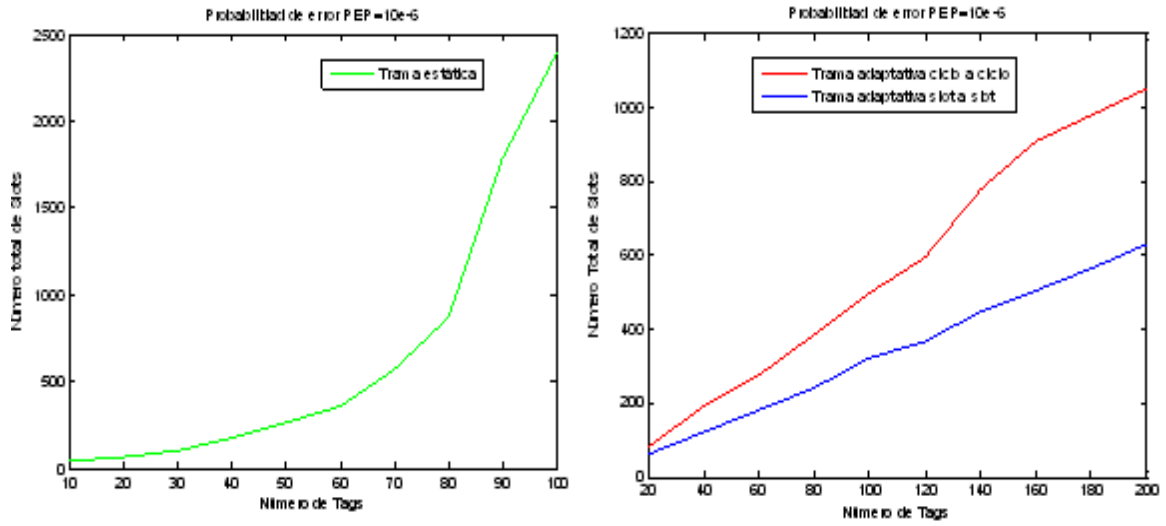


Figura 6.2: Número Total de Slots para una probabilidad PEP=10e-6

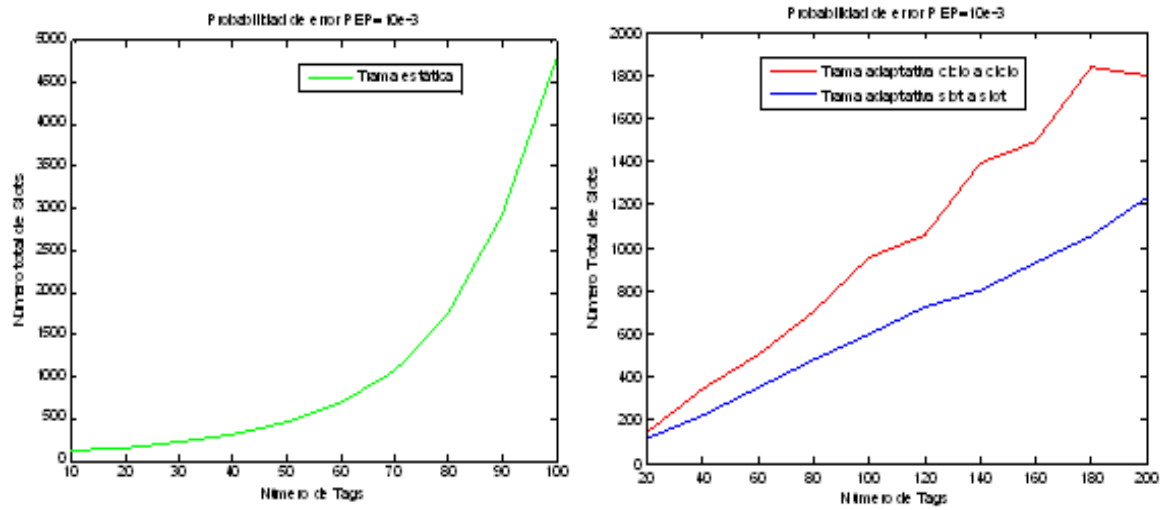


Figura 6.3: Número Total de Slots para una probabilidad PEP=10e-3



Hay que destacar que en el algoritmo de trama estática, el número de slots necesarios para identificar a los tags crece de forma exponencial, conforme aumenta el número de tags en cobertura, lo que justifica el hecho de haber realizado las pruebas de este escenario solo hasta 100 tags. También cabe destacar que el algoritmo de trama adaptativa ciclo a ciclo, si el número de slots es elevado, el reajuste de trama es más lento puesto que no se realiza hasta que no finaliza un ciclo, además, si el número de tags varía rápidamente, tarda en adaptarse a los cambios. Sin embargo, en el algoritmo de trama adaptativa en cada slot, el reader calcula un nuevo valor de  $Q$  al finalizar un slot, así, evita que en el siguiente ciclo haya un gran número de slots vacíos y se evita un mayor retardo en la identificación y por tanto una mayor probabilidad de error si era un slot con colisión.

Por tanto, considerando el número total de slots necesarios para que los tags se identifiquen correctamente, se concluye que el escenario con trama adaptativa slot a slot emplea menor número de slots, y por tanto, el menor tiempo de identificación, independientemente del valor de PEP.

## 6.2 Líneas futuras

Las líneas futuras del trabajo realizado son múltiples, ya que la tecnología RFID es, hoy en día, uno de los *hot-topics* en el mundo de la investigación. Entre las diversas opciones, cabe destacar:

- Sería interesante poder ampliar la validación y simulación del estándar EPCglobal Class-1 Gen2 teniendo en cuenta otros parámetros que influyen en la comunicación. Entre los parámetros físicos, la frecuencia de operación, el número de antenas del lector, la distancia de los tags al lector, las pérdidas de propagación según el entorno de trabajo, etc.
- También se podría implementar un mecanismo de tráfico de tags. Emular la entrada y salida de tags de la zona de cobertura siguiendo diversos patrones de tráfico (Pareto, Poisson, Determinista, etc...). El objetivo de este mecanismo sería el de medir la Probabilidad de Pérdida de Tag, es decir, la probabilidad de que salga un tag de la zona de cobertura sin haberse identificado, y que, a nuestro juicio, es una situación que debe evitarse en la medida de lo posible.
- Por otro lado, también sería interesante poder ampliar el mecanismo de validación y simulación a otros protocolos anti-colisión que se proponen en la literatura científica. Muchos de estos mecanismos no se han validado previamente y han provocado errores de simulación en las pruebas realizadas por el grupo de investigación del Área de Ingeniería Telemática.



# ANEXO I. ESTÁNDARES ISO

---

ISO, posee una lista de comités y concilios técnicos relacionados con RFID:

- ISO JTC1 SC31
- ISO JTC1 SC17
- ISO TC 104 / SC 4
- ISO TC 23 / SC 19
- ISO TC 204
- ISO TC 122

De estos comités surgieron una cantidad importantísima de estándares relacionados con RFID, y utilizados en aplicaciones del mundo real. A continuación, mostramos una lista completa:

- **ISO 6346.** Normas para containers, codificación, identificación y marcado.
- **ISO 7810.** Tarjetas de identificación. Características físicas, criterios de rendimiento, equipamiento para intercambios internacionales, y criterios sobre control hombre-máquina.
- **ISO 7816.** Tarjetas de identificación. Circuitos integrados, tarjetas con contactos. Está dividida en 12 partes.
- **ISO 9798.** Información tecnológica. Técnicas de seguridad y autenticación. Está dividida en 5 partes.
- **ISO 9897.** Normas para containers. Equipamiento, intercambio de información, códigos de comunicación.
- **ISO 10373.** Tarjetas de identificación. Métodos de test. Está dividido en 6 partes
- **ISO 10374.** Normas para containers. Identificación automática.
- **ISO 10536.** Tarjetas de identificación. Circuitos integrados para tarjetas sin contactos. Está dividido en 3 partes.
- **ISO 11784.** RFID para identificación de animales. Estructura del código.
- **ISO 11785.** RFID para identificación de animales. Conceptos técnicos. Especifica el proceso de transmisión entre *tag* y *reader*.
- **ISO 14223.** RFID para identificación de animales. Tags avanzados. Contiene el protocolo de interfaz aire.
- **ISO 14443.** Tarjetas de identificación. Circuitos integrados para tarjetas sin contactos. Tarjetas de proximidad. Está dividida en 4 partes.
- **ISO 14816.** Normas para teletráfico. Equipamiento y automatización de vehículos. Numeración y estructuración de datos.
- **ISO 15434.** Información tecnológica. Sintaxis para transferencia de información ADC.
- **ISO 15459.** Información tecnológica. Identificación de unidades de transporte. Está dividida en 2 partes.
- **ISO 15961.** Información tecnológica. RFID para gestión de objetos. Protocolo de datos y interfaz de aplicación.
- **ISO 15962.** Información tecnológica. RFID para gestión de objetos. Protocolo

- de codificación de datos y funcionalidades de la memoria.
- **ISO 15963.** Información tecnológica. RFID para gestión de objetos. Identificación única para tags RF
- **ISO 17358.** Aplicación para cadenas de suministro. Requerimientos de aplicación.
- **ISO 17363.** Aplicación para cadenas de suministro. Contenedores.
- **ISO 17364.** Aplicación para cadenas de suministro. Unidades de transporte.
- **ISO 17365.** Aplicación para cadenas de suministro. Objetos reutilizables.
- **ISO 17366.** Aplicación para cadenas de suministro. Empaquetamiento.
- **ISO 17367.** Aplicación para cadenas de suministro. Etiquetado de productos (tagging)
- **ISO 18000.** Información tecnológica. RFID para gestión de objetos. (dividido en 6 partes):
  - **18000-1.** Parámetros generales para las interfaces aire y correspondencia con las frecuencias mundialmente admitidas.
  - **18000-2.** Interfaz aire para 135 KHz
  - **18000-3.** Interfaz aire para 13.56 MHz
  - **18000-4.** Interfaz aire para 2.45 GHz
  - **18000-5.** Interfaz aire para 5.8 GHz
  - **18000-6.** Interfaz aire desde 860 MHz hasta 930 MHz
  - **18000-7.** Interfaz aire para 433.92 MHz
- **ISO 18001.** Información tecnológica. RFID para gestión de objetos. Perfiles de aplicaciones.
- **ISO 18047.** Información tecnológica. RFID para testeo. Similar al ISO 18000. Se divide en 3 partes.
- **ISO 18185.** Normas para contenedores. Protocolo de sellado eléctrico. (En desarrollo). Está dividido en 7 partes.
- **ISO 19762.** Información tecnológica. Técnicas AIDC. Dividida en 3 partes.
- **ISO 23389.** Normas para contenedores. Normas de lectura/escritura RFID.
- **ISO 24710.** Información tecnológica. Técnicas AIDC para gestión de objetos con interfaz ISO 18000. Funcionalidades elementales en interfaz aire.

De todas estas normas, las más utilizadas actualmente son la 18000-3 y 6.

# REFERENCIAS

---

- [1] G.Holzmann, "Design and Validation of Computer Protocols", Ed.Prentice Hall, 1991
- [2] Doldi L. "Validation of Telecom Systems with SDL. Wiley, 2003.
- [3] ITU-T Recomendación Z.100 11/99: "Specification and Description Language"SDL.
- [4] [www.telelogic.es/products](http://www.telelogic.es/products)
- [5] [www.rfid-spain.com](http://www.rfid-spain.com)
- [6] [www.igt.es/rfid/](http://www.igt.es/rfid/)
- [7] [www.rfidjournal.com](http://www.rfidjournal.com)
- [8] Evaluación de CSMA No Persistente como protocolo anticolidión en sistemas RFID activos. [ait.upct.es/~eegea/pub/jorRFID.pdf](http://ait.upct.es/~eegea/pub/jorRFID.pdf)
- [9] Class 1 Generation 2 UHF Air Interface Protocol Standard Version 1.0.9: ``Gen 2". [www.epcglobalinc.org/standards](http://www.epcglobalinc.org/standards).
- [10] [www.epcglobalsp.org](http://www.epcglobalsp.org)
- [11] [www.wikipedia.org](http://www.wikipedia.org)