

# Estudio y desarrollo de nuevos sistemas de identificación de productos

Ricard Sociats Rusinés

**Resumen—** Este proyecto surge del interés de estudiar nuevos sistemas de identificación de productos por parte de la empresa Zetes. Esta inquietud se genera debido a que las necesidades de trazabilidad, seguimiento global e identificación individual e única implica a las empresas encargadas de estos procesos a intercambiar información mediante canales paralelos.

Hoy en día los sistemas de identificación de productos almacenan una cantidad reducida de información. Con este pretexto se cree conveniente estudiar y analizar la posibilidad de aumentar estas capacidades, para no depender de canales paralelos, además de conocer y estudiar otras características relevantes para Zetes. Para ello se identifican *barcodes* actuales, se realiza un estudio de las necesidades de los sistemas de identificación y se escogen diversos candidatos para un estudio en profundidad, para finalmente obtener el QR Code como mejor opción. Además, para habilitar nuevas líneas de estudio futuras se proponen alternativas y mejoras a la especificación existente.

**Palabras clave—** Trazabilidad, identificación, código de barras, QR Code, Data Matrix, corrección de errores

**Abstract—** This research started due to the interest Zetes has on investigating new systems of identification for products. Nowadays companies need to trace, track globally and identify unique and individually products, this implies that firms in charge of these processes need to exchange information using parallel channels.

Currently the product identification systems store a small quantity of information. With this pretext it is convenient to study and analyze the possibility of increasing these capabilities, in order not to depend on parallel channels, besides studying and getting to know other characteristics that are relevant for Zetes. In order to do that, barcodes currently used are identified, a research of the systems of identification is conducted and different candidates are selected in order to run a study in depth to finally get the QR Code as best option. Furthermore, alternatives to the current specificities are proposed so that future lines of study are habilitated.

**Keywords—** Traceability, ID, barcodes, QR Code, Data Matrix, error correction

## 1 INTRODUCCIÓN

**A**CTUALMENTE las necesidades de trazabilidad, seguimiento global e identificación individual y única están obligando a las empresas encargadas de la cadena de suministro a colaborar e intercambiar datos de trazabilidad usando canales en paralelo. Debido a esto se requiere la colaboración y la confianza de los actores involucrados en estos escenarios [1].

Esta situación es especialmente crítica en el proceso de agregación. Es decir, cuando se incluyen diferentes artículos, con identificación única, dentro de un contenedor o cuando se comparte de manera temporal transporte logístico. En estos casos la identificación del contenedor no per-

te conocer qué productos individuales contiene de manera directa, sino que es necesario deducirlos.

Para evitar la deducción, se produce un proceso de agregación de confianza y se comparte información a través de la cadena de suministro, enviando en paralelo los datos de cada contenedor. De esta forma todos los actores de la cadena de suministro son capaces de saber qué elementos únicos están contenidos en el contenedor logístico.

Este proyecto está basado en el estudio de un método alternativo para solucionar la problemática de usar el envío de información en paralelo. Para ello se llevará a cabo un análisis de las capacidades máximas de diferentes códigos bidimensionales basados en estándares actuales u otros, que juntamente con la experiencia y conocimientos de la empresa colaboradora Zetes, permitan determinar cuáles son los principales beneficios y ventajas de estas alternativas propuestas.

- E-mail de contacto: ricard.sociats@e-campus.uab.cat
- Mención realizada: Tecnologías de la Información
- Trabajo tutorizado por: Mercè Villanueva Gay (DEIC)
- Curso 2015/16

## 1.1. Objetivos

Los objetivos del proyecto se pueden dividir en 4 puntos:

- Estudiar las características y diferencias entre los distintos tipos de *barcodes*.
- Analizar las necesidades y requerimientos de la empresa Zetes.
- Estudiar las ventajas y desventajas de los *barcodes* según las necesidades de Zetes.
- Describir nuevas alternativas para evolucionar los *barcodes*.

Estos puntos se definen, expanden y detallan en el apartado 2.1 Planificación.

## 1.2. Estado del arte

Todos los productos requieren de un identificador mediante el cual sea posible su control, pero a su vez se requiere de un soporte en paralelo para interpretar y entender este identificador. Hoy en día la mayoría de los sistemas de identificación hacen uso de una base de datos mediante la cual obtienen información del producto gracias a su identificador. Este identificador acostumbra a ser un código de barras (*barcode*) con información reducida.

Este proyecto quiere validar la opción de eliminar esta dependencia del soporte paralelo. Para ello se analizarán y estudiarán diversos *barcodes*. El propósito de dicho estudio es revelar las capacidades máximas de almacenamiento así como otras características relevantes para los sistemas de identificación de productos actuales.

Junto a la empresa Zetes se identificarán cuales de estas características tienen más importancia, para descartar o incluir distintos tipos de *barcodes* a un estudio más exhaustivo. Para ello se analizarán varios casos de uso de implementaciones reales, gracias a los que descubrirán y definirán distintos requisitos a cumplir por los *barcodes*.

## 2 METODOLOGÍA

La metodología que se sigue en este proyecto esta basada en una gestión orientada a objetivos (*Goal Directed Project Management*, GDPM) [2]. Este tipo de gestión usa un enfoque metodológico de arriba a abajo (top down) y está dirigida a obtener resultados, por lo cual se centra en tener siempre presente la orientación de los objetivos que se persiguen en el presente proyecto. A partir de aquí se baja hasta el nivel de los productos que se quieren obtener, líneas de trabajo o productos parciales en que se descompone el proyecto (*Work Breakdown Structure*, WBS) y seguidamente las actividades y tareas que se tienen que llevar a cabo.

La base de una metodología con una gestión orientada a objetivos es definir antes el “qué debe conseguirse” que el “cómo se debe conseguir”. Para ello se plantean unos hitos que darán lugar a unas actividades y tareas que conformarán el instrumento del trabajo diario para conseguir dichos resultados. En el siguiente apartado de planificación se descompondrá el proyecto en WBS, o en otras palabras se descompondrá en diferentes fases y actividades.

## 2.1. Planificación

El objetivo de la planificación es descomponer los objetivos en distintas fases para alcanzar dichas metas. Para ello se ha dividido el proyecto en 8 fases:

- **Fase 1 – Planificación:** fase inicial donde se realizan actividades para conocer y concretar el ámbito y la temática del proyecto. A su vez se definen los objetivos, alcance, metodologías, planificación y fuentes de información.
- **Fase 2 – Investigación *barcodes*:** en esta fase se realiza una investigación a nivel general de diferentes *barcodes* bidimensionales que permita conocer sus principales características.
- **Fase 3 – Estudio necesidades Zetes:** fase en la que se estudian diferentes aspectos de la empresa para comprender sus necesidades y problemáticas específicas. En esta fase se realiza una evaluación de diferentes situaciones reales donde el proceso de agregación supone una problemática, de manera que conseguimos obtener de forma clara unos requisitos completos.
- **Fase 4 – Selección y estudio *barcodes*:** mediante los resultados obtenidos en las fases 2 y 3 se escogen varios *barcodes* para analizar en profundidad y se documentan sus características. En esta fase se aprovecha la colaboración de los conocimientos de Zetes para concretar que *barcodes* incluir en el estudio.
- **Fase 5 – Comparativa *barcodes*:** una vez documentados los *barcodes* y realizado el estudio de Zetes se compararán sus características teniendo en cuenta los aspectos relevantes de la fase 4.
- **Fase 6 – Selección mejor *barcode* según requisitos Zetes:** en esta fase se usa la comparativa de la fase 5, gracias a la que se obtiene una valoración del *barcode* que mejor se adapta a las necesidades de Zetes. Se tendrán en cuenta tanto las ventajas como las desventajas para dar una valoración al código.
- **Fase 7 – Alternativas/Mejoras a los estándares:** en esta fase se definen propuestas de modificación de los estándares para maximizar las capacidades de los *barcodes* analizados. Dichas propuestas son en su mayoría dirigidas al *barcode* con mayor valoración.
- **Fase 8 – Cierre del proyecto:** fase final donde se realiza la documentación a entregar, el artículo del proyecto y la defensa de éste ante el comité evaluador.

## 3 EMPRESA

Este proyecto se realiza con la colaboración de la empresa Zetes. Un grupo internacional especializado en soluciones de identificación y movilidad con sede en Bruselas. Actualmente cuenta con alrededor de más de 1.100 profesionales, junto con una red de oficinas distribuidas en 19 países de la zona EMEA (*Europe, the Middle East and Africa*).

Llevan más de 30 años ofreciendo a sus clientes soluciones innovadoras de movilidad e identificación automática, que garantizan la máxima visibilidad y protección de los datos, tanto en la optimización de la cadena de suministro como en la identificación de personas. Estas soluciones se consiguen valiéndose de la tecnología y ofreciendo

un método integrado en cuatro fases que se resumen en: análisis de necesidades, imprescindible para saber cuál es la solución más adecuada para cada cliente; soluciones llave en mano, para la cadena de suministro y administraciones públicas e instituciones; integración tecnológica, identificación y entrada de datos; integración con los sistemas existentes, soluciones que se integran con los sistemas internos y externos

### 3.1. Necesidades Zetes

Zetes requiere un conjunto de características que debe cumplir un *barcode*. Estas son extraídas de diferentes casos de uso e implementaciones reales de sistemas de identificación de productos y son las siguientes:

- **Capacidad máxima:** esta característica cumple con la necesidad de almacenar cantidades de datos variables según el tamaño de la agregación o de la información relevante del producto.
- **Corrección de errores:** el estado del propio identificador o de los entornos de lectura no pueden estar siempre garantizados, por lo cual es necesario proporcionar una capacidad correctora adaptable a los diferentes entornos y situaciones.
- **Control de sistemas de impresión:** en la mayoría de entornos empresariales e industriales las impresoras son en blanco y negro y con resoluciones mínimas. Es relevante tener en cuenta estas circunstancias y proporcionar compatibilidad con estos sistemas.
- **Uso global y estandarizado:** es importante que las distintas entidades reguladoras den soporte y estandaricen los *barcodes* para garantizar un uso global de estos.
- **Royalties/Licencias:** las tecnologías propietarias implican un handicap importante con el pago de royalties, tanto para el desarrollo como para el propio cliente final.

## 4 INVESTIGACIÓN *barcodes*

En este apartado explicamos de forma clara las principales características de los distintos *barcodes* existentes en la actualidad, para poder así conocer sus principales características. Mediante esta información, se evalúan y se profundiza en su estudio o se descartan. Estas decisiones se toman según las necesidades y requisitos básicos reclamados por Zetes.

En primer lugar, en función de su simbología y sus características para almacenar información existen distintos *barcodes*. Para ello se diferencian a nivel general, 2 tipos de *barcodes*; unidimensionales o lineales (1D) y bidimensionales (2D).

Partiendo de esta premisa, ahora conocida, nuestra investigación nos revela una subdivisión adicional en los *barcodes* 2D. Estos códigos 2D se deben diferenciar y clasificar en simbología apilada (*stacked*) o matricial (*matrix*). La diferencia principal entre ellos es la manera en la que se codifica y se lee su información tal y como se ve en la Fig. 1.

Los de tipo *stacked* están compuestos por varias filas de barras lineales y espacios. Esta tipología es una combinación de *barcodes* lineales, literalmente, apilados unos sobre otros. Tanto su codificación como su lectura se hace de forma horizontal.

Los de tipo *matrix*, en cambio, están implementados mediante el uso de patrones de celdas que pueden ser cuadradas, hexagonales o circulares. Todas ellas forman estos patrones mediante la presencia o ausencia de color en las celdas para codificar la información. Referentes a su forma de codificación y lectura, los códigos matriciales contienen información tanto en el eje X como en el eje Y del símbolo. Dado que los datos se codifican en dos dimensiones, los *barcodes* de tipo matricial permiten almacenar grandes cantidades de datos en menos espacio.

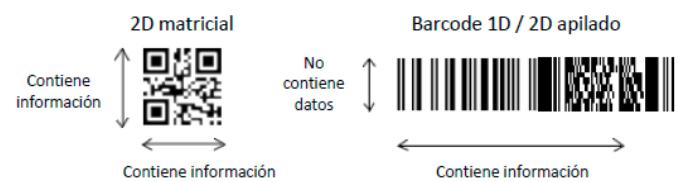


Fig. 1: Diferencias entre *barcodes* apilados y matriciales.

También se conoce la utilización de códigos detectores y correctores de errores [3] por parte de los distintos *barcodes*. Los detectores de errores añaden información redundante de forma que en su lectura es posible detectar si se ha producido un error, pero no qué tipo de error ni dónde. Por otra parte, los correctores usan la misma idea que los anteriores, incluir información redundante, pero en este caso lo suficiente como para permitir deducir cuál es el error y corregirlo. De esta forma, añadiendo esta redundancia a los distintos *barcodes* se consigue hacerlos más tolerantes a errores de descodificación, suciedad o daños físicos. Dependiendo del tipo de *barcode*, la capacidad de corrección de errores varía desde el 5 % al 35 % en la mayoría de los casos.

Cuando se hace referencia a capacidades máximas de almacenamiento cabe hablar de 3 tipos distintos de conjuntos/códigos de caracteres a los que nombrar. De forma general, los más utilizados y en los que nos basaremos en este proyecto son: numérico, alfanumérico y bytes (8-bit). A pesar de que algunos *barcodes* contemplan la codificación del conjunto de caracteres japoneses (Kanji) u otros alfabetos con codificaciones distintas, no haremos referencia a estos conjuntos.

En la Tabla 1 podemos ver algunos de los *barcodes* más significativos junto a sus capacidades de almacenamiento según los distintos conjuntos de caracteres. De todas las simbologías que conforman dicha tabla, el único *barcode* unidimensional es el Code 128.

Si hacemos referencia a las capacidades detectoras o correctoras de estos *barcodes*, todos pueden detectar pero la capacidad correctora queda fuera de sus características en: Codablock F, Code 16K y Code 128. Además, la simbología Code 1 y MaxiCode sólo permite un nivel de corrección fijo.

TABLA 1: CAPACIDADES BARCODES

	Numérico	Alfanumérico	Bytes
Aztec Code	3832	3067	1914
Codablock F	5450	2725	-
Code 1	3550	2218	-
Code 16K	15836	-	8025
Code 128	1 alfanúmerico = 2.79mm		
DataMatrix	3116	2355	1555
DataStrip Code	160bytes/cm <sup>2</sup>		
GridMatrix	1148,875 bytes		
HanXin	7827	4350	3261
HCC2D	8 colores: 291bytes/cm <sup>2</sup>		
Maxicode	138	93	-
PDF417	2710	1850	1018
QR Code	7089	4296	2953
SuperCode	5097	4079	2544

## 5 SELECCIÓN BARCODES

Este apartado tiene como finalidad escoger tres *barcodes*. Para ello se exponen distintas características y motivos que nos permitan hacer una selección de entre los *barcodes* presentados anteriormente en la tabla 1.

La característica más relevante y con un mayor peso en el momento de la selección es la capacidad máxima de almacenamiento. En segundo lugar, una característica con relevancia significativa es la capacidad de corrección de errores. En esta segunda característica no distinguimos entre las diferentes posibilidades de corrección, simplemente definimos la necesidad de corrección de errores además de la de detección.

Con estas dos características mencionadas podemos observar cómo distintos *barcodes* quedan fuera de nuestros requerimientos. Referente a la capacidad máxima de almacenamiento, se descarta el único *barcode* unidimensional, el Code 128. También quedaría descartado el *barcode* Maxicode ya que es el *barcode* 2D con capacidad inferior respecto al resto de los incluidos en la Tabla 1.

Haciendo referencia a la característica de corrección de errores, obtenemos dos simbologías de *barcode* que no cumplen con ella. Tanto Code 16K como Codablock F, no implementan esta característica. Si bien tanto el Code 16K como el Codablock F sí permiten la detección de errores, no admite su corrección. Además de estos dos, el Code 1, al permitir sólo un único nivel fijo de corrección, también se incluye como descarte. Esta decisión responde a la poca versatilidad y adaptación a diferentes entornos donde se requiera ajustar la corrección para así, garantizar una correcta lectura.

Otra característica relevante para la empresa Zetes es la necesidad de control de los sistemas de impresión, así como sus características técnicas. En la mayoría de entornos empresariales e industriales, como ya hemos indicado, las impresoras son en blanco y negro, por lo que el uso del *barcode* HCC2D sería de difícil implantación. Esta simbología hace uso de colores para codificar la información e implicaría disponer de impresoras a color. Además, el uso de colores en los sistemas de identificación de productos también aumentaría los controles necesarios en el momen-

to de la descodificación, ya que la varianza de coloración y tonalidades es muy superior al hecho de controlar color (negro) o ausencia de él (blanco). Por este motivo, y aun siendo el *barcode* con mayor capacidad de almacenamiento de datos, descartamos la simbología HCC2D, debido al uso de colores.

Siguiendo con la evaluación de características relevantes para la confección de un conjunto de *barcodes* que se ajusten de mejor forma a las necesidades establecidas en puntos anteriores, es conveniente descartar la simbología DataStrip. El motivo principal es que esta simbología es propiedad de la empresa DataStrip Inc., y su uso estaría sujeto al pago de los correspondientes royalties. Además, cabe decir, que DataStrip Inc. no muestra registros de actividad reciente por lo que se supone que en la actualidad esta empresa ha dejado de operar.

En este punto, de un conjunto inicial de 14 *barcodes*, una vez descartados los anteriores mencionados, obtenemos un total de 7 *barcodes* restantes de los cuales sólo 2 pertenecen a una simbología de tipo apilada.

Para nuestro estudio, creemos conveniente mantener una simbología de tipo apilada para, además de comparar características, tener una variante a la lectura por visión [4]. De esta forma se puede ofrecer una opción distinta en el caso que sea necesario hacer la lectura del *barcode* mediante lectores láseres. Por este motivo la primera simbología escogida para el estudio es la Supercode y también debido a esto, viendo las capacidades de almacenamiento inferiores, descartamos la otra simbología de tipo apilada: PDF417.

Como cabría esperar y siendo el requisito más importante la capacidad máxima de almacenamiento, se escogen, de los 5 restantes de tipología matriz, los de mayor capacidad. El primer *barcode* sería el HanXin, pero debido a varios motivos que exponemos a continuación, no lo incluiremos en nuestro estudio y nos decantamos por el QR Code.

Los motivos de esta decisión son que principalmente la simbología HanXin fue creada para el mercado chino y está orientada y muy focalizada al uso en este país [5]. En cambio, el QR Code es usado a día de hoy en todo el mundo además de estar estandarizado por GS1 [6] como uno de los dos únicos *barcodes* bidimensionales que se incluye en su estandarización. Además, hay dos variantes del QR Code con características muy interesantes para nuestro estudio, como son el iQR [7], que permite capacidades de hasta 40.000 caracteres y el SQR [7] con la capacidad de cifrado de una parte de los datos para que sólo pueda leerse desde un lector preparado para tal fin.

Hasta el momento no se ha mencionado la variante iQR o SQR debido a que son simbologías propietarias de Denso Wave y de las cuales no hay información pública que podamos analizar. Debido a esto, tampoco se analizan en profundidad en esta fase y se opta por utilizar para el estudio la simbología de la que parten, el QR Code.

Para la selección de la tercera simbología optamos entre Aztec y Data Matrix. Aztec tiene una capacidad de almacenamiento superior en 766 bytes respecto al Data Matrix, pero nos encontramos con un caso parecido al anterior. Data Matrix junto con el QR Code son las dos únicas simbologías 2D que GS1 incluye en su especificación. Data Matrix tiene 2 particularidades por las cuales optamos a su análisis en lugar de Aztec además de que está estandarizado por GS1. La primera es que Data Matrix está diseñado para poder

utilizarse en tamaños reducidos y aunque este punto no es muy significativo respecto a la característica de capacidades máximas, la segunda particularidad si repercute en ella. Data Matrix, a medida que requiere de mayor capacidad para almacenar información, replica su módulo base, es decir si con un módulo no tiene suficiente para almacenar dicha información, replica esta estructura hasta que adquiere dicha capacidad. Por este motivo creemos que es relevante incluirlo en nuestro estudio.

## 6 QR CODE

El *barcode* QR Code viene del inglés Quick Response code. Este fue creado en 1994 por la compañía japonesa Denso Wave y estandarizado a nivel internacional [8] en junio del 2000. Actualmente Denso Wave ha renunciado a los derechos de la patente para los QR Code estandarizados por lo cual el uso de este es de carácter público.

Originalmente el QR Code era capaz de codificar 1.167 caracteres numéricos mediante una matriz de 73 x 73 módulos. Esta versión original es la llamada Model 1. Posteriormente y con la intención de mejorar esta primera versión se definió la actual versión y llamada Model 2. Esta es, como ya hemos mencionado, capaz de almacenar hasta 7.089 caracteres numéricos mediante una matriz de 177 x 177 módulos.

Nuestro estudio se basa en esta segunda versión e incluye las diferentes características añadidas respecto a la Model 1.

### 6.1. Conceptos básicos

Para introducir de manera clara y paulatina los diferentes aspectos de los que se compone el QR Code, así como su estructura, se definen algunos de los aspectos básicos de su estandarización.

El QR Code es capaz de codificar distintos conjuntos de caracteres y entre ellos se encuentran:

- ✓ Numéricos (0-9)
- ✓ Alfanuméricos (0-9, A-Z y los caracteres: espacio, \$ % \* + - . / : )
- ✓ Conjunto de caracteres de 8 bits - JIS X 0201
- ✓ Caracteres Kanji - JIS X 0208

Para la representación de la información se usa como unidad mínima el módulo, el cual representa un 1 binario cuando es negro y un 0 binario cuando es blanco.

Considerando el módulo como unidad mínima, QR Code admite en su simbología básica la llamada versión 1 que está compuesta por 21 x 21 módulos. Así mismo su mayor versión es la 40 y está compuesta por 177 x 177 módulos. El incremento de versión y módulos guarda una relación de  $\frac{1}{4}$  por lo que en cada versión superior se aumentarán 4 módulos por lado.

Relativo a sus capacidades técnicas, hay 2 aspectos a destacar. El primero hace referencia a sus capacidades correctoras de errores. QR Code dispone de cuatro niveles distintos según su capacidad correctora:

- ✓ L - 7 %
- ✓ M - 15 %
- ✓ Q - 25 %

- ✓ H - 30 %

Estos porcentajes hacen referencia a la capacidad de corrección aproximada dentro del conjunto total del símbolo QR Code.

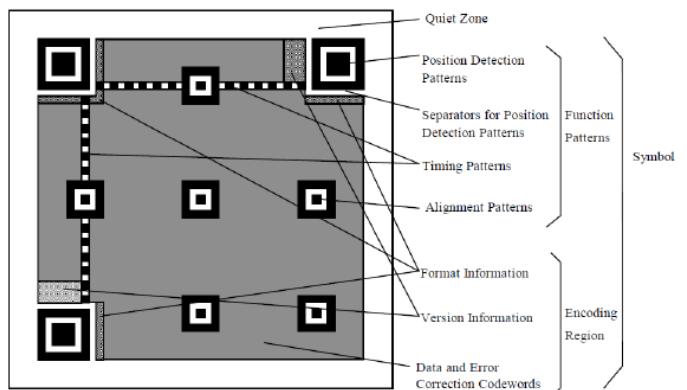
El segundo aspecto relevante es la capacidad máxima de almacenamiento de datos. En su versión 40-L que corresponde al tamaño máximo con el mínimo de corrección, el QR Code puede almacenar:

- ✓ 7.089 numéricos
- ✓ 4.296 alfanuméricos
- ✓ 2.953 datos en formato 8-bit (JIS X 0201)
- ✓ 1.817 caracteres Kanji

### 6.2. Estructura del símbolo

La estructura del QR Code como conjunto se compone de distintas partes además de la información que queremos almacenar. En la Fig. 2 podemos ver de manera completa todas estas partes.

Fig. 2: Elementos que forman la estructura del QR Code.



- **Quiet zone:** corresponde a una zona libre de cualquier marca que rodea el símbolo. Esta zona tiene un tamaño de cuatro módulos.
- **Position detection patterns:** están compuestos por tres patrones de detección de posición idénticos en las esquinas superiores y la inferior izquierda que sirven para alinear el símbolo para su correcta lectura.
- **Separators for position detection patterns:** del tamaño de un módulo y en formato blanco, son una zona utilizada para separar los patrones detectores de posición de la región de datos.
- **Timing patterns:** están formados alternando módulos negros y blancos, empezando y terminando en negro. Siempre se encuentran en la fila y columna 6 de cualquier versión y sirven como puntos de referencia para determinar las coordenadas de los módulos de datos.
- **Alignment patterns:** son similares a los patrones de detección de posición, pero en tamaño reducido. Además de su parecido, también sirven como referencia para alinear el símbolo. El número total de patrones de alineamiento depende de la versión del símbolo.

- **Format information:** contiene información sobre el formato, donde los 2 primeros bits corresponden al nivel de corrección de errores utilizado para los datos del mensaje. Los 3 siguientes indican la máscara y los bits restantes, 10, son para la corrección de errores de los 5 iniciales y son calculados mediante un código BCH(15,5). En total se generan 15 bits entre datos y redundancia. Esta información está duplicada y situada en tres ubicaciones distintas.
- **Versión information:** contiene información sobre la versión y es almacenada en 18 bits. Los 6 bits iniciales representan la versión y los 12 restantes son para la corrección de errores calculados mediante un código BCH(18,6). En total se generan los 18 bits entre datos y redundancia. Esta información está duplicada y situada en dos ubicaciones distintas. Además, sólo está presente a partir de la versión 7 hasta la 40.
- **Data and error correction codewords:** región donde se ubican los datos codificados y su redundancia. Esta redundancia se calcula según el nivel de corrección indicado mediante un código Reed-Solomon. Tanto los datos como la redundancia son colocados en agrupaciones de 8 bits a los que llamamos *codewords*. Esta distribución varía según la versión del símbolo.

### 6.3. Corrección de errores

Los códigos detectores y correctores de errores [3] permiten generar información complementaria para, a posteriori, detectar la veracidad y poder recuperar la información original en casos con sucesos que dañan o no permitieran recuperar la información inicial correctamente. Estos sucesos pueden ir desde errores de lectura a daños en el propio símbolo y por lo tanto falta de información física.

Estos códigos se rigen por tres parámetros  $c$ ,  $k$  y  $r$ , donde cada uno de ellos tiene un valor significativo. La  $c$  representa el total de bits o *codewords*, la  $k$  son los bits o *codewords* de información y por último la  $r$  hace referencia al total de errores que podremos corregir mediante la redundancia generada por este código.

#### 6.3.1. Códigos correctores

En el símbolo QR Code se usan dos tipos de códigos distintos: el BCH y el Reed-Solomon [3].

El BCH se usa para las áreas destinadas a almacenar la información y la versión del propio símbolo. En el caso del *format information*, encontramos dos bits para almacenar el nivel de corrección y otros tres para la máscara. Mediante el uso de un código BCH(15,5) se obtienen diez bits adicionales de la redundancia. Es decir, mediante los cinco iniciales el código BCH nos proporciona diez de redundancia. En el caso de la información de la versión, *version information*, donde tenemos seis bits para almacenar la versión se aplica un BCH(18,6) por lo cual a partir de los seis bits de información obtenemos doce de redundancia. En el caso del código BCH la definición se indica sólo con los parámetros  $(c,k)$ .

Para generar la redundancia de la información se usa un Reed-Solomon. Este se aplica de forma similar al anterior, pero en lugar de mediante  $k$  bits de información se generan  $c-k$  bits de redundancia, en este caso mediante  $k$  *codewords*

(8k bits) se generan  $c-k$  *codewords* (8(c-k) bits) de redundancia. El valor  $r$  resultante varía según el código concreto Reed-Solomon que se use y que está definido en la especificación según la versión del símbolo y el nivel corrector (L, M, Q o H).

Además, para reducir la complejidad de la descodificación y la corrección de errores en el momento de la lectura, los datos organizados en *codewords* son divididos en uno o distintos bloques a los que pueden aplicarse distintos códigos Reed-Solomon para calcular la redundancia total.

#### 6.3.2. Capacidades correctoras

La capacidad correctora se obtiene mediante los *codewords* de redundancia generados por un código Reed-Solomon o BCH.

En los casos donde se usa un código BCH el valor  $r$  es de 3 para el código BCH(15,5) y 2 para BCH(18,6) por lo cual obtenemos un  $(3/15) \times 100 = 20\%$  y un  $(2/18) \times 100 = 11\%$  de corrección de errores de forma respectiva para cada código.

Los 4 niveles de los que dispone QR Code son:

- ✓ L - 7 %
- ✓ M - 15 %
- ✓ Q - 25 %
- ✓ H - 30 %

El valor en % de cada nivel de corrección es el total de errores que puede corregir sobre el total de *codewords* en el conjunto del símbolo, ya sean de datos o de propia redundancia.

TABLA 2:  
CARACTERISTICAS DE CORRECCION DE ERRORES  
PARA QR CODE, CONCRETAMENTE LA VERSIÓN 4.

Número total de codewords	Nivel de Corrección	Número total de codewords de redundancia	Número de bloques de corrección	Corrección de errores por bloque $(c,k,r)$
100	L	20	1	(100,80,10)
	M	36	2	(50,32,9)
	Q	52	2	(50,24,13)
	H	64	4	(25,9,8)

En la Tabla 2 vemos un ejemplo de la definición para un QR Code versión 4, el cual tiene un total de 100 *codewords* disponibles. Según el nivel de corrección a usar, los *codewords* de redundancia aumentan y en consecuencia se reducen los *codewords* destinados a información. En la cuarta columna podemos ver, según la versión, cuantos *codewords* de redundancia se generan. En la cuarta columna vemos en cuantos bloques se dividen los *codewords* de información para aplicar el código Reed-Solomon definido para dicha versión y nivel de corrección.

Por último, en la quinta columna, encontramos la definición del código Reed-Solomon. La  $c$ , es el número total de *codewords*, la variable  $k$  significa el número total de *codewords* de información que se proporcionan al código corrector para calcular los *codewords* de redundancia y la variable más relevante,  $r$ , indica cuantos errores se pueden corregir dentro del bloque donde se ha aplicado el código corrector,

de ahí, es de donde se obtiene el % aproximado para cada nivel de corrección.

Para cada versión del símbolo, y para cada nivel, se especifica en cuantos bloques se dividen los *codewords* de información y para cada uno los parámetros del código Reed-Solomon a usar. Además, en la especificación ISO 18004 también se describen los polinomios generadores para los distintos valores que pueden tomar los parámetros *c*, *k* y *r*.

## 7 DATA MATRIX

El nombre que recibe este *barcode* es la traducción del español de matriz de datos. Esta simbología fue desarrollada en 1982 por la empresa I.D. Matrix [8] con intención principal de poder almacenar información en símbolos de tamaños reducidos. A finales del año 1996 se publicó su especificación de la mano de AIM/TSC y más tarde, en el año 2000, se adoptó esta especificación a nivel internacional como ISO/IEC 16022 [9]. Esta simbología es de uso libre y gratuito por lo cual está exento de pago de royalties.

Actualmente Data Matrix tiene 2 versiones distintas definidas. La versión 1 está compuesta por las simbologías ECC 000, ECC 050, ECC 080, ECC 100 y ECC 140, mientras que la versión 2 está compuesta por la simbología ECC 200. Estas se diferencian según sus capacidades de almacenamiento de información y sus códigos correctores de errores. Aunque detallaremos algunos aspectos de las simbologías de la versión 1, este apartado se centrará principalmente en el estudio de la versión Data Matrix ECC 200. A partir de este punto se usa Data Matrix para hacer referencia a la versión ECC 200 y en los casos referentes la otra versión se especifica cual.

### 7.1. Conceptos básicos

Del mismo modo que con la simbología QR Code, en este apartado introduciremos de manera clara y paulatina los diferentes aspectos de los que se compone el símbolo Data Matrix y su estructura.

Data Matrix es capaz de codificar distintos conjuntos de caracteres y por defecto se encuentra establecida con el siguiente conjunto:

- ✓ Valores de 0 – 127 de acuerdo con ISO/IEC 646. Un ejemplo claro sería los 128 caracteres ASCII.
- ✓ Valores de 128 – 255 definidos de forma particular por Data Matrix.

Como anteriormente, para la representación de la información se usa como unidad mínima el módulo, el cual representa un 1 binario cuando es negro y un 0 binario cuando es blanco. Además, Data Matrix provee una variación estandarizada que permite invertir esta representación y así el 1 binario es representado por blanco y el 0 binario por negro.

Considerando el modulo como unidad mínima, Data Matrix ECC 200 admite un mínimo de 10 x 10 módulos hasta un máximo de 144 x 144. El incremento de módulos no es constante según la necesidad de capacidad de información, en esta simbología además de incrementar el número de módulos se duplican los símbolos de forma completa. Es decir, cuando se requiere más capacidad de almacenamiento, se añaden más módulos (columnas y filas) pero al llegar

al máximo de 24 x 24 módulos, en lugar de seguir añadiendo sólo módulos (columnas y filas) se reducen los módulos y se duplica la estructura completa del símbolo de la forma que muestra la Fig. 3.



Fig. 3: Ejemplo de duplicación de la estructura para ampliar la capacidad de almacenamiento.

Gracias a este procedimiento es posible generar símbolos en formato rectangular en lugar de cuadrado. En nuestro estudio sólo se tienen en cuenta las características relativas al formato cuadrado y no al rectangular. Esto es debido a que las características básicas de la simbología son las mismas, pero con la única diferencia de la colocación y expansión del símbolo resultante. Además, el formato cuadrado permite una mayor capacidad de almacenamiento respecto al rectangular.

Por otro lado, la versión 1 está formada por un mínimo de 9 x 9 módulos y un máximo de 49 x 49 y a diferencia de la versión 2, ECC 200, no se duplica su estructura básica, simplemente se añaden más módulos (columnas y filas).

Relativo a sus capacidades técnicas, hay 2 aspectos a destacar. El primero hace referencia a sus capacidades correctoras de errores mientras que el segundo a sus capacidades de almacenamiento. Data Matrix dispone de seis niveles distintos según su capacidad correctora.

- ✓ ECC 000 – 0 %
- ✓ ECC 050 – 3 %
- ✓ ECC 080 – 5 %
- ✓ ECC 100 – 12.5 %
- ✓ ECC 140 – 25 %
- ✓ ECC 200 – Entre 13 % y 25 %

Estos porcentajes hacen referencia a la capacidad de corrección aproximada dentro del conjunto total del símbolo Data Matrix.

En cuanto a capacidades máximas de almacenamiento de datos, nos encontramos con la peculiaridad que la mayor capacidad la obtenemos sólo con la versión 2, la cual tiene una corrección variable según el tamaño. Esto es debido a como anteriormente hemos mencionado, para los cinco primeros niveles, es decir la versión 1, el máximo definido es de 49 x 49 módulos, mientras que para la versión 2 se definen hasta 144 x 144.

Las capacidades máximas para la versión 1 para la simbología ECC 000 (sin corrección) con formato máximo 49 x 49 y sin duplicación de estructura básica son las siguientes:

- ✓ 596 numéricos
- ✓ 413 alfanuméricos.
- ✓ 271 datos en formato 8-bit

Por el contrario, para la versión 2 con formato máximo 144 x 144, duplicando la estructura básica y con un nivel de corrección del 13,7 %, las capacidades máximas son:

- ✓ 3116 numéricos
- ✓ 2335 alfanuméricos.
- ✓ 1555 datos en formato 8-bit

## 7.2. Estructura del símbolo

La estructura del Data Matrix tiene ciertas diferencias entre las versiones 1 y 2. Como ya hemos comentado anteriormente en el punto de conceptos básicos la estructura de ECC 200 es diferente de las otras versiones, aunque todas ellas comparten los elementos básicos.

Data Matrix se compone de 3 elementos además de la información que almacena.

- **Quiet zone:** corresponde a una zona libre de cualquier marca que rodea el símbolo. Esta zona tiene un tamaño de cuatro módulos.
- **Timing pattern:** este patrón tiene la función de proporcionar un recuento del numero de filas y columnas del que está formado el total del símbolo.
- **Finder pattern:** este patrón se usa para determinar y localizar la orientación del símbolo.

## 7.3. Corrección de errores

Del mismo modo que en el apartado de corrección de errores de QR Code, en este punto se detallan los aspectos relevantes de los códigos detectores y correctores de errores que utiliza Data Matrix, así como se genera la redundancia para permitir recuperar la información en caso de que el símbolo sea dañado o se produzcan errores de lectura.

### 7.3.1. Códigos correctores

El símbolo Data Matrix para la versión que nos ocupa, ECC 200, utiliza un código corrector Reed-Solomon. Para la versión 1 que engloba las otras simbologías y niveles de corrección se usa un código convolucional [3].

El código Reed-Solomon permite generar a partir de los datos de información, organizados en *codewords*, la redundancia. En el caso del Data Matrix la redundancia generada por dicho código permite recuperar un número fijo de errores. A diferencia del QR Code, en esta simbología y buscando siempre las mayores capacidades de almacenamiento, sólo estudiamos la versión ECC 200. Esto es debido a que los otros niveles de corrección de errores están vinculados a la versión con capacidades de almacenamiento reducidas.

Data Matrix distingue dos metodologías para el cálculo de la redundancia. En símbolos con menos de 255 *codewords* totales se usan estos mismos sin hacer uso de interleaving. Por otra parte, en los casos que el número total de *codewords* supere 255, la redundancia se calcula usando un procedimiento específico llamado interleaving descrito en el anexo A de la especificación ISO 16022 [9].

En términos generales el interleaving es una técnica que consiste en organizar los datos de una forma determinada para así proteger la información frente a errores de ráfaga. Es decir, errores que afecten a bits de forma consecutiva.

### 7.3.2. Capacidades correctoras

La capacidad correctora se obtiene mediante los *codewords* de redundancia generados por el código Reed-Solomon.

Para la simbología Data Matrix el nivel de corrección está fijado en: ECC 200 – Entre un 13 % y 25 %. Este tanto por ciento nos indica que aproximadamente podremos corregir este porcentaje de los *codewords* totales del símbolo una vez generada la redundancia e incluida con la información inicial. Este porcentaje se obtiene mediante la expresión:  $2t \leq d$ . Donde  $t$  es el número de errores y  $d$  el número de *codewords* de redundancia.

Por ejemplo, en el caso del símbolo Data Matrix de tamaño 72x72 y con un total de 512 *codewords* de los cuales 368 sean de datos y 144 de redundancia, la capacidad correctora sería de  $2t \leq 144$  por lo cual se podrían corregir 72 de los 512 *codewords* totales. Esta operación refleja la capacidad correctora, en este caso de 14 %.

Del mismo modo que con el uso de Reed-Solomon para el QR Code, según el número de *codewords* de redundancia que sean necesarios generar, se usa uno de los 16 polinomios generadores definidos en la especificación.

De este modo, según el polinomio generador, el número de bloques, el número de *codewords* de redundancia y el número de *codewords* de información, se calcula el porcentaje de *codewords* que podremos recuperar en caso de error. Estos valores varían según el tamaño del símbolo. Para el símbolo más pequeño obtenemos una capacidad correctora del 25 % mientras que para el símbolo más grande el porcentaje es del 13,7 %.

## 8 SUPERCODE

Esta simbología fue inventada por Ynjiun Wang en 1994 y es de dominio público [10]. Aun siendo de dominio público ha sido imposible obtener su especificación [11] de manera gratuita. Por este motivo y debido a que en la planificación de este estudio no se ha considerado ningún presupuesto para acceder a recursos de pago, se ha optado por obtener la información de otras fuentes [11, 12]. Dichas fuentes a veces son contradictorias, debido a que es posible que existan revisiones de esta simbología y que dichas fuentes se refieran a una u otra revisión sin hacer mención de ello. Teniendo en cuenta todo esto y sabiendo que esta simbología se incluyó en el estudio como opción para mantener la posibilidad de usar lectores láseres, el estudio de este símbolo es significativamente más reducido con respecto a QR Code y Data Matrix.

### 8.1. Conceptos básicos

Esta simbología usa una estructura de paquetes la cual es una variante de otras simbologías multilínea. Para almacenar sus datos y su redundancia usa varios paquetes organizados en filas que se pueden leer de forma independiente. Estos paquetes una vez leídos son reestructurados en su posición para su correcta descodificación. Este procedimiento se da gracias a que cada paquete contiene información de su número en la secuencia y a la vez permite desordenar dichos paquetes para garantizar que el símbolo final no tendrá secuencias de ceros (blanco) o unos (negro) de forma consecutiva que puedan complicar su lectura. Además,

el hecho de organizar la información en paquetes y no en líneas completas permite al lector leer la información con un ángulo mucho más inclinado [12] que no si tuviera que leer todo el ancho completo del símbolo.

La capacidad de almacenamiento máxima con la mínima capacidad correctora es de 2544 bytes. SuperCode hace uso de un código Reed-Solomon para calcular *codewords* de redundancia y así proporcionar la capacidad correctora de errores. Esta capacidad es configurable y varía en torno a 32 niveles distintos [12].

## 9 SELECCIÓN MEJOR *barcode* SEGÚN REQUISITOS ZETES

El objetivo de este apartado es ratificar las evidencias que se han ido manifestando a lo largo del desarrollo de los apartados anteriores y a su vez relacionarlas con las características más significativas que precisa la empresa Zetes en un *barcode*.

La característica más importante es, sin lugar a duda, la capacidad máxima. Siendo conscientes de esta, entre las simbologías analizadas la que aporta mayores capacidades, según su especificación, y sin tener en cuenta la capacidad correctora de errores es la de QR Code. Es cierto que para una correcta comparación es necesario equiparar los niveles de corrección para así, obtener unos resultados de capacidad máxima comparables. Haciendo este ejercicio, obtenemos los valores de la Tabla 3, donde el QR Code es claramente superior.

TABLA 3: COMPARATIVA QR CODE VS DATA MATRIX CON LOS MISMOS NIVELES DE CORRECCIÓN.

Simbología	Versión/Tamaño	Nivel de corrección	Capacidad numéricos
QR Code	40-M	18,5 %	5596
Data Matrix	144 x 144	13,7 %	3116

Además, el hecho de que Data Matrix sólo tenga un único nivel de corrección para niveles superiores de almacenamiento reduce significativamente la adaptación a diferentes entornos. Enlazando con estas capacidades de adaptación, Zetes trabaja en diferentes entornos donde cada uno de ellos puede requerir un nivel de corrección distinto, en este aspecto el QR Code también posee una ventaja significativa respecto al Data Matrix al poseer 4 niveles de corrección.

Otra característica importante es la estandarización. Tanto la simbología Data Matrix como la QR Code ostentan el apoyo y certificación por parte de GS1 [6]. En este aspecto las dos simbologías aportan el mismo nivel de garantías.

Teniendo en cuenta las 3 características más relevantes, se puede observar de forma clara, como la simbología que se adapta mejor a las necesidades de Zetes es la QR Code. Además, esta simbología aporta un valor añadido ya que su estructura fue diseñada para una lectura de forma rápida, de tal forma que es posible que su uso sea beneficioso en entornos donde el tiempo sea una característica crucial.

## 10 CONCLUSIONES

La identificación de productos, concretamente mediante *barcodes*, tiene un gran potencial aún sin explotar gracias

a los *barcodes* bidimensionales. En este proyecto se ha trabajado junto con la empresa Zetes y gracias a sus conocimientos se ha podido obtener unos requisitos básicos de la identificación de productos, mediante los cuales se han filtrado los *barcodes*.

En este proyecto se han querido reflejar las posibilidades, así como las características, de los distintos *barcodes* existentes. A la vez, se han analizado en profundidad aquellos que reflejaban en sus características el cumplimiento de los requisitos obtenidos de Zetes.

Como reflejo del estudio y mediante los resultados obtenidos, podemos ver como la simbología QR Code ha destacado entre las otras por sus características, principalmente por su capacidad de almacenamiento. Este *barcode* es el mejor candidato de los analizados para desarrollar un nuevo sistema de identificación de productos.

Cabe decir que es posible que en algunos casos o entornos la especificación actual del QR Code no pueda cumplir y adaptarse con las necesidades de Zetes. Para ello se ha incluido un apartado con alternativas y mejoras a esta especificación para un posible estudio futuro. Estas alternativas hacen referencia tanto a la simbología QR Code como a las otras analizadas.

### 10.1. Alternativas/Mejoras

En este apartado se plantean distintas hipótesis para remediar o ampliar aspectos que no se adapten suficientemente a la demanda de Zetes. El principal objetivo es el de presentar las hipótesis que mediante un posible posterior estudio permitan ampliar las capacidades del *barcode* elegido o de otra forma paliar los aspectos negativos de sus rivales analizados.

#### Eliminar características innecesarias

Eliminar características de las distintas simbologías será posible en el caso de que los escenarios de uso se garanticen. Por ejemplo, en un escenario donde se garantice que los símbolos se encuentran siempre en posición vertical y su lectura será en esa posición, los elementos que permiten alinear el símbolo serían innecesarios.

#### Aumentar tamaño

Las dos simbologías analizadas se encuentran encapsuladas dentro de unos límites de tamaño definidos por el número total de filas y columnas, añadiendo más número de estas se aumentaría la capacidad de datos a almacenar.

#### Reducir corrección de errores

El hecho de modificar las capacidades correctoras significa de manera directa aumentar las capacidades de almacenamiento. Esta alternativa o mejora sería relevante en el caso de poderse aplicar en las distintas simbologías, pero, en el caso de Data Matrix permitiría adquirir la capacidad de adaptación de la cual no dispone. Además, en entornos totalmente controlados donde la corrección de errores es poco significativa o innecesaria la eliminación de esta aportaría un extra de capacidad.

#### Cambiar los modos de codificación

Los modos de codificación son estándares definidos y en algunos casos no se adaptan correctamente al espacio de

caracteres que realmente se usa en nuestro escenario. Una posible mejora sería definir un espacio de caracteres propio donde solo se definan los caracteres que realmente sean necesarios. Es decir, si nuestros identificadores de producto en lugar de usar 20 dígitos numéricos usaran 10 dígitos numéricos y una letra comprendida entre A-B-C necesitaríamos hacer uso del modo alfanumérico que contempla muchos más caracteres en nuestro caso innecesarios.

### Modificar la densidad de caracteres por bits

Los distintos modos de codificación usan agrupaciones distintas para posteriormente convertir dichas agrupaciones a binario. Cabe la posibilidad que modificando el número de caracteres a agrupar aumentemos la densidad de carácter por bits.

Tomando como ejemplo el modo numérico usado en el QR Code, que agrupa de tres en tres y obtiene una densidad de aproximadamente 3 caracteres por 10 bits. Es decir, 3 caracteres codificados en 10 bits permiten representar  $2^{10} = 1024$  números por lo cual se obtiene una ratio de  $\frac{10}{3} = 3,33$  bits/número. Agrupando de dos en dos, nos harían falta 7 bits ( $2^7 = 128$ ) y obtendríamos una ratio de  $\frac{7}{2} = 3,5$  bits/número.

### Usar el modo anexado

Sin modificar la especificación de ninguna de las 2 simbologías es posible usar el modo anexado [5, 7] para concatenar distintos símbolos de la misma tipología para almacenar más información. Este modo permite dividir la información en distintas partes para codificarla en varios símbolos y recuperando la información inicial una vez se hayan leído todos los símbolos generados y que forman dicho conjunto.

Tanto en la simbología QR Code como en Data Matrix el máximo de símbolos que se pueden anexar es de 16. Calculando de forma aproximada con QR Code se podría llegar a obtener una capacidad de 47.000 bytes al anexar los 16 símbolos. Del mismo modo con la simbología Data Matrix y anexando un total de 16 se obtendría una capacidad de 24.500 bytes.

### Comprimir información

Cabe la posibilidad de comprimir la información a almacenar antes de generar la redundancia y colocarla de esta forma en el símbolo final. Para poder aplicar esta propuesta posiblemente sería necesario reservar un espacio para identificar si los datos están comprimidos o no, además del método de compresión usado y del mismo modo que hay un espacio para la versión o el formato habría que reservar un espacio para indicar esta información adicional.

### Simbología iQR Code

Esta simbología de carácter privado, forma parte del catálogo de productos de la empresa Denso Wave, creadora de la simbología QR Code. Según los detalles que muestran en su web y en una presentación de GS1 de Japón[5] iQR Code es la evolución de QR Code con algunos cambios que recuerdan a características usadas en la simbología Data Matrix. Esta simbología tiene una capacidad máxima de almacenamiento de aproximadamente 40.000 dígitos numéricos o de aproximadamente 5 veces más que el QR Code descrito en este proyecto. Además, incluye un nivel de corrección de errores que permite corregir el 50 % y puede almacenar la

misma información e imprimirse en un 30 % de espacio menos.

## 11 AGRADECIMIENTOS

Quiero agradecer a mi tutora Mercè Villanueva Gay, el tiempo que ha invertido en ayudarme y solucionar las cuestiones y dudas que me han ido surgiendo durante todo el proyecto, así como su voluntad en corregir y aconsejarme en cada una de las fases del mismo.

También quiero agradecer a la empresa Zetes, especialmente a Jordi Soler Pla, por brindarme la oportunidad de desarrollar este proyecto, además del trato amable que han tenido conmigo y la ayuda proporcionada para entender y obtener recursos que han facilitado el desarrollo del trabajo.

## REFERENCIAS

- [1] “Guía sobre las TIC en el sector de la logística y transporte en la PYME y la MICROPYME,” Aug 2014.
- [2] R. J. Ramon., M. J. Garcia, and I. L. Orozco, *Gestión de proyectos informáticos*. UOC, 2007.
- [3] W. C. Huffman and V. Pless, *Fundamentals of error-correcting codes*. Cambridge University Press, 2003.
- [4] “Evaluación de las tecnologías de lectura de códigos de barras,” Oct 2014.
- [5] “Han xin code.” [Online]. Available: <http://www.renuzhzhui.com/index.php/hanxincode>
- [6] “Two-dimensional (2D) barcodes.” [Online]. Available: <http://www.gs1.org/barcodes/2d>
- [7] S. Yokota and G. Japan, “QR code overview & progress of QR code applications,” 2009.
- [8] “Information technology — automatic identification and data capture techniques — QR code 2005 bar code symbology specification,” 2009.
- [9] “Information technology — automatic identification and data capture techniques — bar code symbology specifications — data matrix,” 2005.
- [10] A. S. B. Raj, *Bar codes: technology and implementation*. Tata McGraw-Hill Pub., 2007.
- [11] “ISS - SuperCode - AIM.” [Online]. Available: <https://aimglobal.site-ym.com/store/viewproduct.aspx?id=2761089>
- [12] Y. Wang, “Packet bar code with data sequence encoded in address/data packets,” Jan. 1996, uS Patent 5,481,103. [Online]. Available: <https://www.google.ch/patents/US5481103>