

Soporte multiformato de decodificación códigos 2D

Pablo Orduña, Aitor Almeida, Eduardo Castillejo (Universidad de Deusto).

Román Rodríguez, Marcos Sacristán (Treelogic)

Antonio Manuel Benjumea Conde (Servinform)

Abstract

Existen muchas librerías y proyectos de códigos dimensionales, todos con diferente codificación de la información y para diferentes plataformas y dispositivos. El propósito de este escrito es el desarrollo de un lector de códigos dimensionales para dispositivos móviles que permita a las aplicaciones su uso con diferentes códigos y librerías de forma transparente, e incluso la inclusión de nuevas librerías, sin modificar la aplicación original. Este paquete de trabajo se centra en proveer a la plataforma PIRAmIDE la habilidad de usar un lector de códigos como si de un sensor más del terminal se tratara.

Fueron analizados diversos códigos dimensionales, dentro de todos ellos los más interesantes son QR Code y Datamatrix: por el soporte con el que cuentan, por ser estándares internacionales, la capacidad de almacenaje de información y las librerías o proyectos existentes para trabajar con ellos, algunas de ellas para dispositivos móviles.

El sistema está dividido en dos capas, una para la adquisición, iterando por las librerías instaladas para decodificar las imágenes en datos en bruto para luego, en la segunda capa, recoger esa información y convertirla en objetos de alto nivel a través de un repositorio de formatos, repositorio que además es ampliable por el desarrollador de aplicaciones. Así cualquier código con el mismo tipo de información (localización, calendario, contacto) generará un mismo formato de alto nivel para que las aplicaciones originales no necesiten modificarse para entender un nuevo código.

Estos formatos se dividen en formatos padre, o de alto nivel, que aglutinan las características de los diferentes formatos hijo, o de bajo nivel, que son la definición de los formatos de los diferentes códigos instalados. También se presenta un tipo de código personalizable, que permite que para una misma imagen se pueda variar la información contenida en ella a través de un campo del formato.

Introducción

Este paquete de trabajo se centra en ofrecer a la plataforma PIRAmIDE la capacidad de utilizar códigos de barras y bidimensionales como un sensor más de las aplicaciones.

Esto es, una aplicación podría leer códigos bidimensionales indicando localización compleja dentro de un edificio para que al ser leído por el usuario le indique su posición en un mapa o indicaciones de cómo ir al lugar que desea. También podría indicarle automáticamente si un producto que ha comprado está caducado o su procedencia, o se podría utilizar para etiquetar objetos del entorno tal y como contemplan los desarrolladores de las aplicaciones construidas por encima de PIRAmIDE.

Sin embargo, existen en el mercado muchos tipos de códigos diferentes. Algunos pueden almacenar más información (como QR Code o Data Matrix), y pueden ser utilizados para situaciones complejas por los desarrolladores de aplicaciones. Otros códigos de barras (como

EAN-13 o GS1 Databar Expanded) almacenan menos información, pero están más extendidos, por lo que se pueden desarrollar aplicaciones que interactúen con productos ya etiquetados como los que ya podemos encontrar en el mercado. Además, existen diversas formas de almacenar una misma información dentro de un código.

Con objetivo de facilitar al desarrollador de aplicaciones la gestión de estos códigos y formatos, se ha desarrollado un subsistema capaz de abstraerle del código utilizado, de la librería utilizada y del formato utilizado de una manera extensible a dos niveles: soportando que se añadan nuevas librerías que puedan decodifican nuevos códigos y soportando que se añadan nuevos formatos.

En primer lugar se describe el estado del arte de los códigos y librerías decodificadoras analizadas. Posteriormente, se detalla la arquitectura general del subsistema, para posteriormente detallar el módulo que gestiona el soporte de múltiples formatos. Por último, se explican los resultados obtenidos del proyecto y las conclusiones.

Estado del arte

En este apartado se estudian las características de los diferentes códigos dimensionales existentes y las librerías que pueden ser utilizadas para trabajar con ellos. La siguiente tabla resume la información clave de los códigos bidimensionales analizados:

	QR Code [QRC]	Data Matrix [DaMa]	PDF417 [PDF417]	Beetagg [BeTa]	GTIN-13 [GTIN-13]	ShotCode [ShCo]	TRIP [TRIP]	MaxiCode [MaCo]	Aztec [Aztec]	GS1-128 [GS1-128]	Databar [DaBa]	High Capacity Color Barcode [HCCB]	EzCode [Ezcode]	Moseycode [MoCo]
Tipo código	Matriz	Matriz	Apilado	Panel de abeja hexagonal	Barras	Circular	Circular	Rejilla hexagonal	Matriz	Barras	Barras	Rejilla de triángulos	Matriz	Área cuadrada con con diferentes tipos de marcado
Tamaño de imagen	La cantidad depende del nivel. La matriz es desde 21x21 del Nivel 1 hasta 177x177 del Nivel 40. El nivel 4 es de 33x33 pixels	Desde 8x8 hasta 144x144.	Máximo: 30 columnas x 90 filas.	Desde 100x100px hasta 500x500px.	15 bloques que tienen desde cada uno dos barras de diferentes tamaños.	2 circunferencias alrededor de la diana, con 24 porciones cada una	2 circunferencias alrededor de la diana, con 16 porciones cada una	33 filas x 30 columnas de módulos hexagonales	Desde 15x15 hasta 151x151	48 bloques por línea, cada uno podrá tener varias barras	14 bloques por línea, cada uno podrá tener varias barras	5x10 con separadores horizontales (5 filas de 10 triángulos)	11x11 celdas, siendo su tamaño más pequeño de 11x11 mm ²	Dividido conceptualmente formando una rejilla de 24x24.
Información almacenable	Tiene diferentes niveles. El máximo (40) puede almacenar hasta 2953 bytes. Sin embargo cada implementación leerá hasta un nivel determinado (en móviles suele ser hasta nivel 4 -78)	Hasta 1556 bytes (144x144). Otras medidas: 10x10 puede almacenar 1 byte. Con 36x36 84 bytes. Con 32x32 60 bytes.	1,100 bytes	Número binario de tamaño 90 → 2 ⁹⁰ códigos diferentes → 11 bytes + 2 bits de información.	13 dígitos (5 bytes)	Hasta 49 bits	Hasta 9 trits (equivale a entre 14-15 y bits)	Hasta 93 caracteres alfanuméricos / 138 caracteres numéricos	Desde 12 bytes hasta 1914 bytes	48 caracteres por línea, y el número de líneas no está limitado	74 números o 41 caracteres del alfabeto	12,5 bytes (1 byte cada 4 símbolos)	Desconocida.	96 bits en el código, siendo ilimitada la información a la que accede el usuario (información en un servidor)

	bytes- o nivel 6 -134 bytes-)													
Corrección de errores	Configurable. Tiene varios niveles: capaz de recuperar desde 7% hasta 30% (desciende la capacidad de almacenar en un código de un tamaño determinado).	ECC 220, capaz de recuperar hasta 30%	Hay 9 niveles (0 a 8) de recuperación de errores, llegando hasta un 30% en su nivel máximo (reduciendo a menos de la mitad la capacidad).	Desconocido.	Sólo detección de errores	Sólo detección de errores	Sólo detección de errores	Capaz de recuperar hasta 25%	Seleccionable entre 5% y 95%. Recomendable 23%	Sólo detección de errores	Sólo detección de errores	Reed Solomon. No se especifica el nivel de corrección de errores.	Desconocido.	Hasta un 12,5% usando código binario Golay Extendido
Estándar	ISO/IEC18004	ISO/IEC16022	ISO/IEC 15438:2006	No	EAN-13, actualmente GS1	No	No	ISO 16023:2000	No	ISO/IEC 15420	ISO/IEC 646	No	No	No
Comentarios adicionales	Es, junto a Data Matrix, el estándar más popular. QR Code es especialmente popular en Japón.			Es un código destinado al marketing.	Códigos de barras utilizados para almacenar ISBN en todos los productos.	Basado en TRIP.		Desarrollado y usado por UPS: formatos orientados a direcciones postales.				En fase beta. Las limitaciones en el uso de este código, tanto de almacenamiento en el código como en el servidor, son impuestas por Microsoft, no por el formato de codificación en si.	Desarrollado por ETH Zurich y con licencia para uso privado de la empresa Scanbuy. Es utilizado por Movistar en su aplicación para móviles denominada "Bidi".	

La siguiente tabla resume la información clave de las implementaciones analizadas:

	libdmtx	zxing	jjil	Nokia Barcode Reader	EzCode	ShotCode	TRIP	Zebra	QR Midlet	Moseycod	Idautomation	Partitek PtBarcodeDec	BeeTagg
--	---------	-------	------	----------------------	--------	----------	------	-------	-----------	----------	--------------	-----------------------	---------



PERSONALIZABLE INTERACTIONS WITH RESOURCES
ON AMI-ENABLED MOBILE DYNAMIC ENVIROMENTS

Códigos soportados	Data Matrix	QR Code: Hasta nivel 6 EAN13 DataMatrix: experimental UPC-A, UPC- E Code 39 Code 128	Ninguno, es una librería para el tratamiento de imágenes	DataMatrix QR Code	EzCode QR Code DataMatrix	ShotCode	TRIP	EAN-13/UPC-A UPC-E EAN-8 Code 128 Code 39 Interleaved 2 of 5	QR	Moseycode	Aztec Data Matrix PDF417 MacroPDF417 Maxicode QR-Code	PDF417 QR Code DataMatrix	BeeTagg QR-Code Datamatrix
Plataformas software	Linux Windows Mac OS X Symbian Python S60 iPhone	JavaSE Android JavaME	JavaSE Android JavaME	NOKIA (Symbian 60)	Android BlackBerry Palm OS Windows Mobile iPhone OS	iPhone Compact .NET Symbian Palm OS JavaME	JavaME Compact .NET Framework	C/C++ Linux OS Portable	J2SE J2ME MIDP2.0/CLDC1.0	Android	Windows DOS Macintosh AS/400 Unix Linux Microsoft .NET Framework C/C++ JAVA	Windows Unix/Linux WinCE/Windows Mobile Symbian	BlackBerry Java Java light Palm Symbian S60 2nd Symbian S60 3rd Symbian UIQ Symbian UIQ3 Windows Mobile
Utilizable como librería	Sí	Sí	Sí	No	No	No	Sí	Sí	Sí	Sí	Sí	Sí	No
Licencia	Open Source y gratuito (GNU LGPL)	Open Source y gratuito (Apache License 2.0)	Open Source y gratuito (GPL 2.0)	Libre distribución	Libre descarga del lector	Cerrado	Open Source y gratuito	Open Source y gratuito (GNU LGPL)	Open Source y gratuito (GNU GPL)	Open Source y gratuito	De pago	De pago	Cerrado
Versión	0.6	1.2	12/22/2008	-	-	3.0.0	-	0.5	1.0	0.2.1	-	3.0	2.1
Comentarios	Desarrollada en C++ pero utilizable en varias plataformas incluyendo Symbian e importable desde Python S60	Desarrollado por Google	Es una librería para el tratamiento de imágenes que incluye dos aplicaciones: Lector de códigos EAN-13 y Detección de rostros.	Utiliza los servidores de NOKIA para obtener la información completa del código capturado.	Es un código bidimensional propiedad de Scanbuy.			Desarrollada en C/C++ que incluye interfaces para Python y Perl y Plugins para navegadores tipo Mozilla.			Librería junto con aplicaciones desarrollada en C/C++/C#/JAVA y de pago	Librería en formato DLL para windows y SO para sistemas UNIX/LINUX con una gran capacidad para reconocer los códigos bidimensionales y realizar corrección de errores.	Sólo se puede utilizar este código por medio de los servicios ofrecidos por la compañía propietaria del código bidimensional.

Los códigos que más información pueden almacenar y más soporte cuentan, además de ser estándares internacionales, son DataMatrix y QR Code. Entre las implementaciones analizadas, ambos cuentan con 7 proyectos que decodifican estos códigos, de los cuales 4 pueden ser utilizados como librerías, y de estos, 3 pueden ser utilizados por terminales móviles. PDF417 también está estandarizado y es capaz de almacenar tanta información como los dos códigos anteriores, aunque el soporte en las implementaciones analizadas es menor.

Además, los códigos GTIN-13, GS1-128, MoseyCode y TRIP, si bien son capaces de almacenar considerablemente menos información, cuentan con implementaciones analizadas que podrían ser utilizadas como librerías incluso en dispositivos móviles. Los códigos a tener en cuenta serían los siguientes:

Código	Implementaciones analizadas que lo soportan	Implementaciones que pueden ser utilizadas como librería	Implementaciones que pueden ser utilizadas como librería en dispositivos móviles	Información almacenable
QR Code	7	4	3	2953 bytes. En dispositivos móviles hasta 78 o 134
DataMatrix	7	4	3	1556 bytes. Menor soporte en dispositivos móviles
PDF417	1	1	1	1100 bytes. Menor soporte en dispositivos móviles
GTIN-13	2	1	1	5 bytes
GS1-128	1	1	1	27 bytes por fila
MoseyCode	1	1	1	12 bytes
TRIP	1	1	1	11 bytes

Los códigos Databar y HCCB son estándares de los cuales no se ha analizado ninguna implementación, pero aún así son códigos estándares que podrían ser utilizadas en caso de que encontrásemos implementaciones interesantes. De Maxicode (que también es estándar) y Aztec también hay implementaciones entre las analizadas que los soportan incluso como librería, aunque no pueden ser utilizadas en dispositivos móviles. Se debería vigilar la futura evolución de los siguientes códigos:

Código	Implementaciones analizadas que lo soportan	Implementaciones que pueden ser utilizadas como librería	Implementaciones que pueden ser utilizadas como librería en dispositivos móviles	Información almacenable
DataBar	0	0	0	30 bytes
HCCB	0	0	0	12,5 bytes
Aztec	1	1	0	1914 bytes

MaxiCode	1	1	0	53 bytes
----------	---	---	---	----------

Los códigos ShotCode, Beetag y EzCode son códigos privados no estandarizados de los cuales hay una única implementación que no puede ser utilizada como librería. Es por ello que hemos descartado su uso dentro del proyecto PIRAMIDE.

Código	Implementaciones analizadas que lo soportan	Implementaciones que pueden ser utilizadas como librería	Implementaciones que pueden ser utilizadas como librería en dispositivos móviles
Beetag	1	0	0
EzCode	1	0	0
ShotCode	1	0	0

Cotejando todas las tablas, en Mayo de 2009, la situación de las implementaciones es la siguiente:

	DataMatrix	QR Code	Databar
Symbian	libdmtx, Partitek PtBarcodeDec (de pago)	Partitek PtBarcodeDec (de pago)	Ninguna
Java ME	zxing (en alpha)	zxing, QR midlet	Ninguna
Windows Mobile	Partitek PtBarcodeDec (de pago)	Partitek PtBarcodeDec (de pago)	Ninguna
iPhone	libdmtx		Ninguna
Android	zxing (en alpha)	zxing	Ninguna

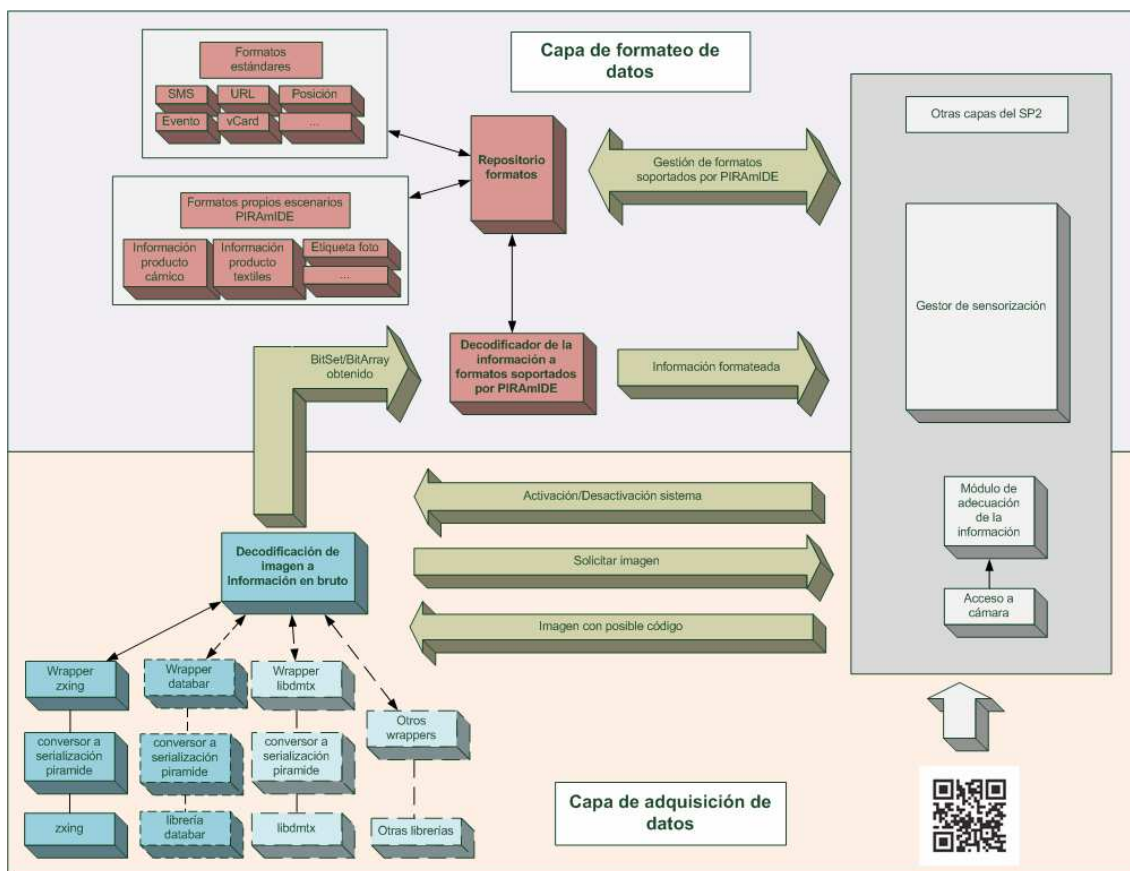
Coloreando de rojo librerías de pago, de verde las Open Source y gratuitas y de amarillo las que todavía no están soportadas. Cuando zxing implemente DataMatrix (se ven avances en los últimos meses), el mapa pasaría a ser el siguiente:

	DataMatrix	QR Code	Databar
Symbian	libdmtx, Partitek PtBarcodeDec (de pago)	Partitek PtBarcodeDec (de pago)	Ninguna
Java ME	zxing	zxing, QR midlet	Ninguna
Windows Mobile	Partitek PtBarcodeDec (de pago)	Partitek PtBarcodeDec (de pago)	Ninguna
iPhone	libdmtx		Ninguna
Android	zxing	zxing	Ninguna

Por lo que DataMatrix pasaría a ser especialmente interesante.

Arquitectura general

Tal y como se aprecia en la siguiente figura, el subsistema está dividido en dos capas independientes. La primera, la capa de adquisición de datos, se encarga de obtener la información representada en el código bidimensional. La segunda, la capa de formateo de datos, se encarga de procesar esta información de manera que puede se puede expresar en formatos de alto nivel.



Para ello, el usuario utilizará una aplicación que internamente activará el subsistema. Este solicitará imágenes que podrán contener un código. Estas imágenes son pasadas a un módulo de decodificación de imágenes, que llamará a una serie de librerías disponibles para que intenten decodificarlas. Así, el sistema consultará a la librería zxing para obtener información de códigos QR o Databar, y consultaría a la librería libdmtx para códigos Data Matrix si el wrapper de esta librería fuese implementado.

Posteriormente, el módulo decodificador de la información tomará la información en bruto obtenida de las librerías, e intentará convertirla en objetos de alto nivel. Para ello, consultará con un repositorio local de formatos, que pueden ser añadidos por el propio desarrollador de aplicaciones. Estos formatos, tal y como se detalla en la siguiente sección, están definidos en XML, y contienen instrucciones para que el módulo decodificador tome la información en bruto y vaya convirtiendo la información en diferentes estructuras de datos complejas que posteriormente devolverá a la aplicación del usuario.

De esta manera, un código QR con información de localización formateada para ser entendida por zxing, pasará a ser formateado como información de localización tal y como está expresado en el proyecto. De esta manera, para entender otro código que exprese información de localización para ser entendida por otra librería se podrá añadir un nuevo XML que genere la información de localización. Así, la aplicación original no tendrá que ser modificada para poder procesar este nuevo formato.

Soporte multiformato

Con el soporte multiformato se consigue abarcar cualquier tipo de código utilizado por las aplicaciones construidas sobre PIRAmIDE.

Este soporte consiste en la definición de una serie de formatos en XML que detallan las características de la información contenida en los códigos. Con este paso sería posible decodificar cualquier formato soportado por las librerías utilizadas por PIRAmIDE.

La información codificada puede ser dependiente de las librerías o del desarrollador, ya que por ejemplo algunos de los formatos definidos por NTT Docomo son diferentes que los definidos por zxing para la codificación QR y para el mismo tipo de información. Por este motivo es necesario avanzar un paso más: la solución propuesta plantea la definición de unos formatos generales de nivel superior (formatos “padre”) que engloben las características comunes y no comunes de los diferentes formatos existentes para un tipo de información equivalente (formatos “hijo”).

Así pues, se realiza una nueva iteración desarrollando nuevos formatos que aglutinan las características comunes de los anteriores a la vez que enlazan con ellos, pero esta vez completamente independientes. Con ello es posible que diferentes códigos, para un mismo tipo de información, sean interpretados de igual forma por las aplicaciones PIRAmIDE. Esta doble definición de formatos permite que los desarrolladores puedan utilizar cualquier tipo de formato de información en sus códigos y que a la vez pueda ser consumido por otras aplicaciones a las que les resulte transparente el tipo de formato utilizado. Para ello sólo habrá de definirse un formato PIRAmIDE general para ese tipo de información (formato “padre”).

Los formatos han sido diseñados para que aunque no sea posible integrar toda la información en la imagen del código se provea un mecanismo de acceso a información extendida, a través de una url.

También han sido definidos, a modo de demostración y como extensión de los formatos de información existentes en los dos estándares fundamentales a día de hoy, formatos específicos para los casos de uso inicialmente considerados en los dominios de aplicación clave del proyecto, para etiquetado de alimentos en entornos de e-Inclusión, etiquetado de prendas en entorno textil.

Y como novedad se reservó un tipo de código genérico que sirve para etiquetar cualquier tipo de producto o lugar, de manera que el código sólo contenga los parámetros necesarios para acceder a cierta información, y no contenga información en sí. Esta aproximación pretende ser una solución al problema de códigos bidimensionales personalizables.

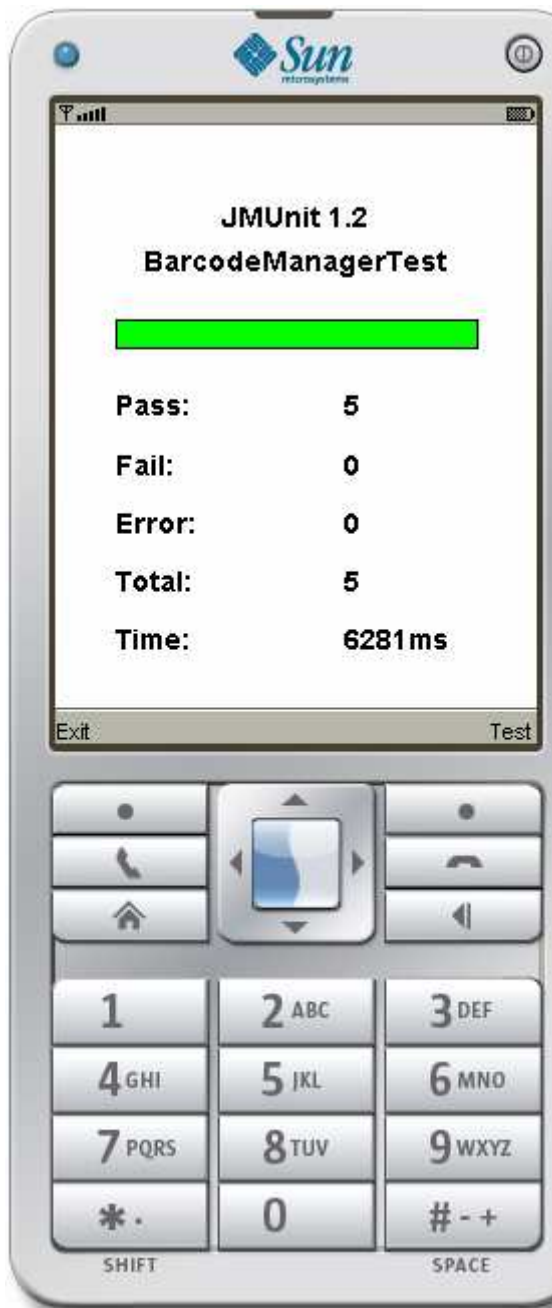
Resultados obtenidos

a) Módulo decodificador

El módulo decodificador mantiene un listado actualizado de formatos a utilizar, que puede ser ampliado en cualquier momento por cualquier aplicación que se instale en el sistema. Esta ampliación se produce de manera sencilla, simplemente copiando un fichero XML con los nuevos formatos en un directorio predefinido. El sistema lo detectará automáticamente y cargará los nuevos formatos.

Se provee además toda una funcionalidad de gestión de formatos para permitir su registro, eliminación y actualización. También hay que destacar que se proveen funcionalidades para la gestión de los módulos de las librerías decodificadoras. Todas estas funcionalidades han sido diseñadas para proveer una gestión eficiente de los recursos, pudiéndose eliminar de memoria decodificadores o formatos con poco uso. Y sumadas a un uso austero de los recursos, permiten

que el módulo decodificador sea capaz de cargar todos los datos necesarios para su puesta en funcionamiento y decodificar una imagen en tan sólo medio segundo*.



Carga de sistema para conjunto de pruebas.



Carga de sistema y decodificación de imagen.

Las pruebas fueron realizadas sobre emulador Sun WT Versión 2.5.2_01 para CLDC.

* Con baja carga de formatos.

Debido a las escasas capacidades de algunos terminales del mercado este dato indica una eficiencia que permitirá que cualquier dispositivo pueda utilizar esta funcionalidad con relativa soltura.

b) Databar en zxing

Tal y como se ha detallado en el estado del arte, al comienzo del desarrollo del proyecto se detectó la necesidad de soportar el formato databar, y se observó que no existía ninguna implementación que lo soportara en teléfonos móviles.

Para ello, se tomó el proyecto zxing [zxing10], que es un proyecto Open Source de decodificación de diferentes códigos de barras y bidimensionales desde diferentes plataformas software como Android, Java SE o Java ME, y se comenzó a soportar diferentes formatos de GS1 Databar. Al ponernos en contacto con los desarrolladores del proyecto, informaron que estaban ya soportando dos de los formatos (GS1 Databar Omnidireccional y GS1 Databar Omnidireccional Stacked), por lo que nos centramos en contribuir el formato GS1 Databar Expanded. El código ha sido aceptado y es ahora parte del proyecto zxing [zxing-contributions10].

Asimismo, durante el desarrollo de este formato, se observaron fallos en la librería de codificación Zint [zint10], de los que se informaron a los desarrolladores e incluso se enviaron parches para arreglar algunos de estos problemas. Estos fallos han sido corregidos de manera que en la siguiente versión del proyecto genera correctamente los códigos GS1 Databar Expanded [zint-contributions10].

Conclusiones

Los resultados obtenidos permiten la decodificación de códigos QR y Databar, haciendo uso para ello de la librería zxing. Las ampliaciones no conllevarían apenas tiempo de desarrollo siempre y cuando las librerías de decodificación hayan sido desarrolladas en las tecnologías de la plataforma. La incorporación futura del formato Datamatrix resulta de gran interés para las aplicaciones, por su uso masivo, al igual que QR, y que está integrada de forma experimental en la librería zxing.

La definición de formatos XML para los diferentes códigos puede ser realizada por cualquier desarrollador y utilizada por todas las aplicaciones que hagan uso del módulo. Si la plataforma no contiene esas definiciones y se ven necesarias, sería posible ampliarlo lanzando nuevos formatos genéricos PIRAmIDE.

Agradecimientos

El Proyecto Singular y de carácter Estratégico PIRAmIDE (TSI020301-2008-2) ha sido aprobado por el subprograma Avanza I+D dentro de la convocatoria de ayudas de Acción Estratégica de Telecomunicaciones y Sociedad de la Información 2008, habiendo sido financiado por el Ministerio de Industria, Turismo y Comercio (MITYC) y el Fondo Europeo de Desarrollo Regional (FEDER).

Referencias

- [QRC] Referencia ISO/IEC 18004.
http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=43655

- [DaMa] Referencia ISO/IEC 16022.
http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=44230
- [PDF417] Referencia ISO/IEC 15438.
http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=43816
- [BeTa] Página principal. <http://www.beetagg.com/>
- [GTIN-13] Referencia ISO/IEC 15420.
http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=27663
- [ShCo] Página principal. <http://www.shotcode.com/>
- [TRIP] TRIP: a Low-Cost Vision-Based Location System for Ubiquitous Computing. Diego López de Ipiña, Paulo Mendonça and Andy Hopper, Personal and Ubiquitous Computing journal, Springer, vol. 6, no. 3, pp. 206-219, May 2002.
- [MaCo] Referencia ISO/IEC 16023.
http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=29835
- [Aztec] Barcode Studio 2007. <http://barcodingstudio.com/help/help.aspx?hld=aztec>
- [GS1-128] Referencia ISO/IEC 15420.
http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=27663
- [DaBa] Referencia ISO/IEC 24724.
http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=38834
- [HCCB] Página principal. <http://research.microsoft.com/en-us/projects/hccb/default.aspx>
- [EzCode] http://www.ethz.ch/index_EN
- [MoCo] Página principal. <http://www.moseycode.com/>
- [zint] Página principal. <http://www.zint.org.uk>
- [zint-cont] Contribución.
<http://zint.cvs.sourceforge.net/viewvc/zint/zint/backend/rss.c?view=log>
- [zxing] Página principal. <http://code.google.com/zxing/>
- [zxing-cont] Anuncio contribución.
http://groups.google.com/group/zxing/browse_thread/thread/1c08ab8dc03d09e6/32625a06a84999f8