

Università degli Studi di Camerino

SCUOLA DI SCIENZE E TECNOLOGIE

Corso di Laurea in Informatica (Classe L-31)



Qr-Code e Microsoft Tag: use case

Laureando
Chiara Cini
Matricola 77852

Relatore
Prof. Fausto Marcantoni

A.A. 2010/2011

Si volge ad attendere il futuro solo
chi non sa vivere il presente.
Seneca

Indice

Introduzione	5
1 Il progetto	7
1.1 Panoramica	7
1.2 Obiettivi	8
2 Ambienti di sviluppo	9
2.1 Ubuntu 10.04 e Web Server Apache 2	9
2.2 Sistema operativo smartphone: Android 2.2	10
2.3 Eclipse Indigo	14
3 Tecnologia utilizzata	16
3.1 Codici a barre: un po' di storia	16
3.2 Barcode bidimensionali	17
3.3 Qr-Code	20
3.4 Microsoft Tag	23
3.5 Differenze	24
3.5.1 Architettura	24
3.5.2 Licenza	25
3.5.3 Librerie	25
3.5.4 Utilizzo	25
3.5.5 Software	26
3.5.6 Personalizzazione	26
4 Fasi dello sviluppo	28
4.1 Prima fase: creazione delle applets	28
4.1.1 QR-Code applet	28
4.1.2 Tag Microsoft applet	30
4.1.3 Permessi e firma	32
4.2 Seconda fase: creazione sito internet	34
4.3 Terza fase: creazione di applicazioni per Android	36
4.3.1 QR-Code Reader	36
4.3.2 Microsoft Tag Reader	37
4.3.3 Adattabilità	38

5 Conclusioni	39
Bibliografia	43

Introduzione

Con la crescente popolarità della telefonia mobile, si tendono a sviluppare nuove soluzioni, nuovi accorgimenti per semplificare la comunicazione (o marketing) tra aziende, enti istituzionali e non e l'utente, affinché quest'ultimo possa ottenere nel miglior modo possibile, cioè veloce e chiaro, informazioni e contenuti direttamente sul proprio telefono.

Il cellulare, infatti, non è più solo un dispositivo utilizzato per le vecchie e care telefonate, ma ora è molto di più. E' diventato uno strumento elettronico complesso. Mobile Marketing è un nuovo termine che identifica proprio il marketing diretto ai telefonini e ai palmari.

La crescente diffusione di un collegamento internet negli smartphone e il cambiamento nel modo di utilizzarlo da parte degli utenti, hanno fatto sì che il mobile marketing diventasse uno strumento di notevole impatto sulla nostra vita.

Il marketing, infatti, è il primo strumento utilizzato dalle aziende per comunicare. All'inizio lo strumento principale per farlo era l'SMS. Le aziende o gli enti istituzionali, inviavano messaggi agli utenti per informarli su avvenimenti, prodotti, possibili offerte. Era ed è uno strumento che riesce a raggiungere molti utenti in poco tempo e a costo ridotto. Questa pratica però era facilmente accostata a problemi di privacy e allo spamming sul cellulare.

Ora si è fatto un passo avanti. Non è più necessario che sia l'azienda ad inviare messaggi per comunicare le proprie informazioni, è l'utente che, attraverso un gesto con il suo telefonino, può risalire a tutte le informazioni riguardanti un prodotto o un evento, messe a disposizione dall'ente stesso. Questo accade attraverso una nuova tecnologia, i barcode (codici a barre) bidimensionali, evoluzione dei precedenti codici a barre.

I barcode contengono tutte le informazioni che le aziende decidono di inserire, da URL, a testo, a numeri telefonici. Essi sono letti da un telefono cellulare dotato di fotocamera e tramite apposite applicazioni restituiscono immediatamente le informazioni contenute.

In questa tesi sarà affrontato, prima lo studio e il confronto tra due tipi di barcode, i QR-Code e i Tag Microsoft, poi lo sviluppo di un progetto che ne comprende l'utilizzo all'interno del dipartimento universitario.

Struttura della tesi

Il primo capitolo introduce il sistema generale, dando una visione completa delle caratteristiche e dei requisiti che il progetto deve raggiungere.

Il secondo capitolo parlerà degli ambienti di sviluppo che si andranno a utilizzare, spiegandoli prima in maniera generale e poi entrando maggiormente nel dettaglio.

Il terzo capitolo spiegherà in maniera approfondita al lettore la tecnologia dei barcode bidimensionali.

Il quarto capitolo è l'ultima parte della tesi ed elencherà le fasi dello sviluppo, entrando quindi dettagliatamente nella realizzazione implementativa del progetto.

1. Il progetto

In questo primo capitolo sarà presentata una panoramica generale del progetto e gli obiettivi concordati da raggiungere.

1.1 Panoramica

Il progetto consiste nell'analizzare la tecnologia dei nuovi barcode bidimensionali all'interno di zone in cui è necessaria una maggiore interazione dell'utente con il luogo. Un caso interessante sono i musei, dove a fianco di ogni quadro o scultura, potrebbe essere aggiunto un codice bidimensionale che contiene all'interno informazioni sull'opera, sull'autore o un link al sito di quest'ultimo. Ma sono anche altri i possibili campi di applicazione per questa tecnologia: ad esempio nelle mostre o, come accade già nelle zone orientali, nello shopping o nel marketing, o per pubblicizzare luoghi e film.

In questo progetto il luogo scelto è il dipartimento universitario. Il dipartimento è costituito dagli studi dei professori e dalle aule. L'idea è di associare un barcode ad ogni aula e ad ogni studio, attaccandolo a fianco di ciascuna porta.

Lo studente così, utilizzando l'applicazione adatta sul suo smartphone e inquadrando il barcode con la fotocamera, ne può leggere il contenuto come da figura 1.1. I contenuti possono essere diversi, come contatti, numeri telefonici e url. In questo caso i barcode conterranno solo url. Infatti, le url rimanderanno ad un sito creato appositamente, contenente le informazioni riguardanti i professori dello studio associato. Si può utilizzare lo stesso principio anche per le aule, così da poter ottenere informazioni su di esse o sugli orari delle lezioni.

I tipi di barcode presi in considerazione per questo progetto sono due: i Tag della Microsoft e i Qr-Code, così da poter fare un confronto e uno studio più approfondito sulle diverse tecnologie.

Per apprendere a pieno il loro funzionamento, l'idea è di crearli realmente, o meglio sviluppare applicazioni in Java che ne implementino la codifica: una per i QR-Code, i barcode a quadrati bianchi e neri e una per i Tag che sono rappresentati da triangoli di diversi colori.

Le applicazioni saranno poi inserite sul sito, così da permettere a tutti di poter creare il proprio barcode bidimensionale.

Per completare il progetto e lo studio dei barcode, affronterò l'implementazione di due applicazioni Android che permettano di decodificare il contenuto, in particolare url, e rimandare al sito di riferimento per leggere le informazioni dei professori o delle aule. Il sito è realizzato in html e php, ed è stato sviluppato con Apache 2 su Ubuntu.

Questa tecnologia permette una velocizzazione dei tempi di ricerca e minimizza le possibilità di errore, digitando url sbagliate, oppure troppo complicate da ricordare.



Figura 1.1: Rappresentazione grafica del progetto

Sicuramente il progetto può essere esteso anche ad altre situazioni che potrebbero ottenere vantaggi dall'utilizzo di questa tecnologia.

1.2 Obiettivi

Gli obiettivi da raggiungere al termine di questo progetto sono:

- uno studio iniziale della tecnologia dei codici a barre;
- un confronto tra due tipi di barcode QR-Code e Microsoft Tag;
- la generazione dei codici a barre;
- la realizzazione di due applicazioni che leggono i differenti barcode;
- adattare se possibile le applicazioni a dispositivi Android di diversa grandezza (smartphone e tablet).

2. Ambienti di sviluppo

Il capitolo approfondirà quali sono gli ambienti di sviluppo utilizzati nel progetto, ovvero:

- il sistema operativo Linux Ubuntu Desktop 10.04 e il web server Apache 2 per la configurazione del sito web;
- il sistema operativo dello smartphone, Android 2.2, sul quale testerò l'applicazione che legge i QR-Code e i Tag-Microsoft;
- l'ide Eclipse integrato con la SDK (Software Development Kit) di Android che permette la programmazione delle applicazioni su questa piattaforma, e inoltre la creazione di applet in Java standard da far girare online.

2.1 Ubuntu 10.04 e Web Server Apache 2

Uno dei passaggi fondamentali per lo sviluppo di questo progetto è l'utilizzo di un web server sul quale si possa ospitare il sito che conterrà tutte le informazioni riguardanti i professori, le aule ed anche le applet per la creazione dei rispettivi Qr-Code e Tag Microsoft.

La scelta è ricaduta sul web server Apache 2 installato sul sistema operativo Ubuntu Desktop 10.04. Apache 2 ha un insieme di caratteristiche davvero interessanti, come ad esempio:

- SSL e TSL, protocolli di comunicazione criptati tra client e server;
- un modulo proxy;
- interfacce che supportano i linguaggi più comuni;
- moduli di autenticazione;
- linguaggio di programmazione lato-server.

Essendo un web server, è utilizzato principalmente per l'hosting virtuale. Un solo web server può essere utilizzato per realizzare più siti differenti.

Oltre a essere in grado di servire contenuti web statici e dinamici su Internet, questo server può essere utilizzato per servire anche altre applicazioni come database Oracle o LAMP e per la trasmissione sicura dei dati sulla rete. Essendo un programma open source, è messo a disposizione dell'utente in maniera gratuita, permettendone modifiche e aggiornamenti a chiunque.

L'installazione di Apache su Ubuntu è molto semplice e può essere eseguita da terminale grazie a questo comando:

```
sudo apt-get install apache2
```

Sudo è l'abbreviazione inglese di Super User Do. E' un programma che permette di eseguire un comando come utente amministratore (root), quindi ogni volta che esegue un comando di questo tipo, il sistema ci chiederà la password.

Il file di configurazione predefinito di Apache è situato nel percorso `/etc/apache2/apache2.conf`. È possibile modificare questo file per configurare il server secondo le proprie esigenze. È possibile configurare il numero della porta, l'origine dei documenti, i moduli, i file di registro, gli host virtuali e altro.

All'interno del file di configurazione troviamo la direttiva `DocumentRoot` che stabilisce in che posizione, il server va a cercare i file che compongono il sito. Il valore predefinito è `/var/www`, ma può essere modificato se opportuno. In questa cartella si trova il file `index.html` che è la pagina di presentazione di Apache. Se s'inserisce in un browser l'indirizzo ip `127.0.0.1`, ovvero `localhost`, si vedrà il risultato di questa pagina se l'installazione è stata fatta correttamente.

Le pagine del sito dovranno essere inserite in questa cartella o in sottocartelle di essa, per essere visualizzati dal server. Questa cartella non ha i permessi di modifica ma inizialmente solo di lettura, quindi tramite il comando sottostante, otteniamo anche quello di scrittura:

```
sudo chmod -R 777 /var/www
```

A questo punto siamo pronti per creare il sito vero e proprio. La spiegazione è rimandata al paragrafo 4.2. Il sito può essere realizzato anche con un Content Management System, per ampliarne le caratteristiche e renderlo anche più accattivante dal punto di vista grafico.

Un CMS è uno strumento software installato su un server web studiato appositamente per poter gestire meglio i contenuti, rendendo più semplice la realizzazione di un sito ad un amministratore con poca conoscenza in ambito di programmazione web. Esso utilizza un database interno nel quale possono essere collocati i contenuti. Il CMS è suddiviso in due parti:

- Back-end: che permette all'amministratore di gestire e supervisionare i contenuti posti nel sito;
- Front-end: che permette all'utente di visionare il sito in maniera semplice, visualizzando contenuto e applicazioni disponibili.

Uno dei vantaggi nell'utilizzo dei CMS è la possibilità di creare siti dinamici, facili da aggiornare anche se di dimensioni elevate, senza dover modificare il codice. Anche la grafica può essere modificata cambiando solamente un foglio di stile. Ci sono moltissimi tipi di CMS, sia proprietari che non. Tra quelli open source, uno dei più utilizzati è Joomla.

2.2 Sistema operativo smartphone: Android 2.2

Android inc. è l'azienda creatrice del sistema operativo Android ed è stata fondata in California da Andy Rubin, Nick Sears e Chris White nel 2003. Nel 2005 viene acquisita da Google.

Appena due anni dopo è rilasciata la prima SDK (Software Development Kit) e un anno dopo sul mercato esce il primo dispositivo con Android 1.0, l'HTC Dream (G1).

Nel susseguirsi degli anni Android si è sempre migliorato, arrivando fino alla versione 4.0 dei moderni tablet.

Il sistema è basato completamente su Linux, che è il sistema operativo open source più diffuso. Ha un insieme di servizi e applicazioni affidabili, facili da integrare poiché è supportato dalla casa madre Google, la quale ha adattato tutte le sue tipiche applicazioni all'uso in mobilità.

Android ha sviluppato numerose versioni nel tempo. Android può supportare un insieme molto vasto di tecnologie per la connettività, come GSM/EDGE, CDMA, EV-DO, UMTS, Bluetooth, e Wi-Fi. L'interazione che l'utente ha con Android è quasi totalmente basata sul touchscreen.

L'architettura di questo sistema operativo è basata su diversi livelli, come si può vedere dalla figura 2.2. La parte principale del sistema Android è ovviamente il kernel Linux, dove all'interno sono stati inseriti direttamente i controlli dell'hardware del dispositivo: tastiera, schermo, bluetooth e così via. Appena sopra il kernel troviamo le librerie. Da segnalare ovviamente le principali, usate anche nel progetto, come SQLite per il database o OpenGL per la grafica.

Nell'architettura è compresa una macchina virtuale e una libreria fondamentale, che permettono lo sviluppo delle applicazioni. La macchina virtuale si chiama Dalvik e non è altro che una Java Virtual Machine (JVM) con una differenza: la Java Virtual Machine esegue il codice bytecode, mentre Dalvik esegue un codice chiamato DEX (Dalvik EXecutable), che permette una migliore resa sui dispositivi smartphone.

Al livello successivo vediamo che ci sono le application framework ovvero i gestori per le risorse, per le applicazioni installate, per le telefonate, per il file system e altro.

Nell'ultimo livello, infine, troviamo le applicazioni vere e proprie come il browser o file explorer, molte incluse all'acquisto del telefono con l'installazione di base.

Per sviluppare le applicazioni, è necessario avere il kit di sviluppo di Android (SDK) il quale contiene librerie, emulatore, esempi e anche molta documentazione. L'SDK è disponibile gratuitamente e la si può scaricare dal sito di Android Developers[3], scegliendo quella adatta al proprio sistema operativo. E' disponibile per Windows, Linux e MacOS.

L'installazione del kit è molto semplice, basta scompattare l'archivio scaricato. Una cosa importante da accertare prima di compiere questo passo è di avere sul proprio computer la JDK (Java Development Kit), indispensabile poiché Android si programma in Java.

La creazione delle applicazioni Android di questo progetto è rimandata al paragrafo 4.3.

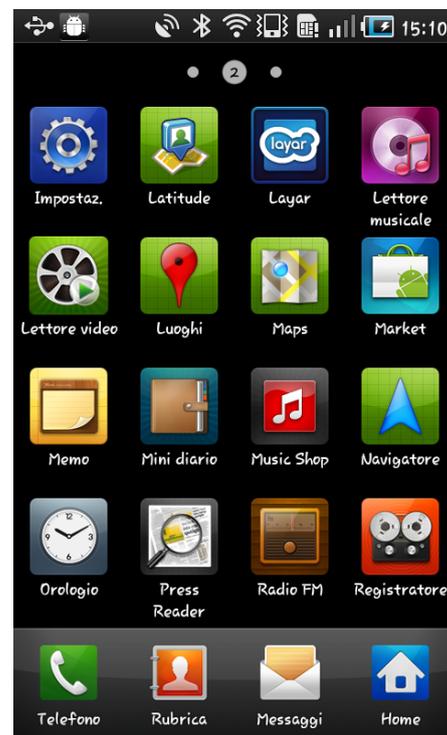


Figura 2.1: Schermata iniziale di Android



Figura 2.2: Architettura di Android

Per quanto riguarda la struttura delle applicazioni, Android fornisce quattro modelli:

- **Activity:** componente che è destinato ad una diretta interazione con l'utente. Un'activity può essere vista come la rappresentazione di una schermata (screen). La piattaforma mantiene in primo piano (foreground) un solo screen alla volta, portando in background tutti gli altri. L'activity ha un suo ciclo di vita (Figura 2.3), il quale è composto dai seguenti metodi:
 - onCreate() chiamato la prima volta quando l'activity è lanciata;
 - onStart() chiamato quando l'activity sta per essere visualizzata;
 - onResume() chiamato quando l'activity comincia ad interagire con l'utente;
 - onPause() chiamato quando l'activity sta per andare in background;
 - onStop() chiamato quando l'activity non è più visibile all'utente;
 - onRestart() chiamato quando l'activity sta per essere ridisegnata;
 - onDestroy() chiamato quando l'activity deve essere distrutta.
- **Service:** componente che gira in background e non fornisce un'interfaccia utente. Può essere lanciato sia dal boot del telefono oppure da un'altra activity o componente. Un esempio semplice è la riproduzione in sottofondo di un brano musicale mentre l'utente compie altre azioni;
- **Content Provider:** componente che gestisce un insieme condiviso di dati dell'applicazione. I dati possono essere salvati nel file system o in un database. Un

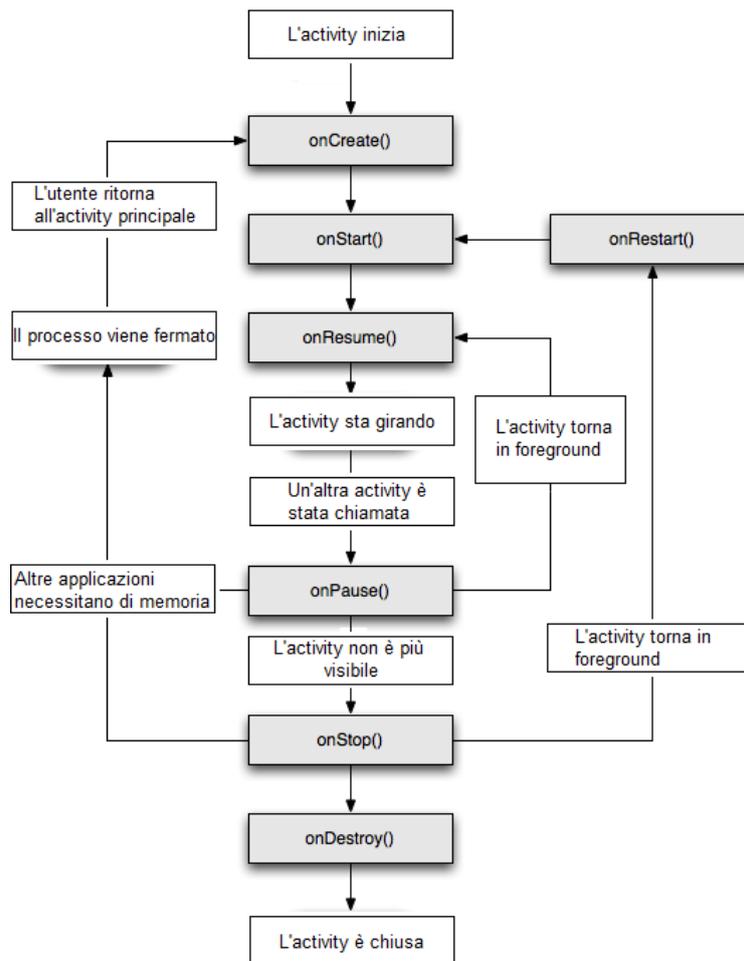


Figura 2.3: Activity lifecycle

esempio è il NotePad, che salva le note scritte dall'utente per renderle disponibili successivamente su richiesta;

- Broadcast Receiver: componente che fa da ricevitore, a livello di sistema, di annunci broadcast. Molti annunci provengono dall'interno, ma altri possono essere creati anche dalle applicazioni, per segnalare che un evento è avvenuto, ad esempio l'annuncio che lo schermo è spento o che la batteria è scarica.

Per quanto riguarda la grafica, essa può essere sviluppata da codice, utilizzando le varie librerie dedicate come SGL per la grafica bidimensionale oppure lo standard OpenGL ES 2.0 per la grafica tridimensionale. Il sistema operativo però, mette a disposizione anche un'altra modalità di costruzione delle interfacce utente (UI), la quale si basa su file XML.

Prima di poter avviare un elemento dell'applicazione, Android deve sapere che l'elemento esiste e questo avviene grazie alla lettura del file AndroidManifest.xml. L'applicazione deve dichiarare tutti i suoi componenti in questo file, il quale si trova alla radice della directory del progetto dell'applicazione[10].

Il passo successivo è la configurazione della piattaforma di sviluppo.

2.3 Eclipse Indigo

Eclipse è una piattaforma di sviluppo integrata, multi-linguaggio. Ufficialmente supporta oltre a JAVA anche C/C++, COBOL, PHP, e AspectJ. E' un framework basato su un sistema di API pubbliche e di plug-in, le cui specifiche di sviluppo sono di pubblico dominio. In questo modo chiunque può permettersi di sviluppare i propri plug-in e renderli disponibili alla comunità Open Source oppure, farli pagare. La versione utilizzata per questo progetto è Indigo, l'ultima release del giugno 2011.

Per quanto riguarda lo sviluppo di Android all'interno di questo IDE (Integrated Development Environment), è necessario installare un plug-in dal sito di Android. Il plug-in può essere installato in modo semplice direttamente da Eclipse. Dal menu Help si deve cliccare su Install new software. Giunti a questo punto si può aggiungere un nuovo sito[2] dal quale poter scaricare il plug-in. Selezionare infine Developer Tools, chiudere il form e riavviare Eclipse (Figura 2.4).

Ora bisogna aggiungere l'sdk di Android a Eclipse, quindi dal menu Windows, poi Preferences, si deve selezionare Android e inserire il giusto percorso della cartella dell'sdk (Figura 2.5). Ora l'ambiente di sviluppo è pronto per creare il primo progetto Android.

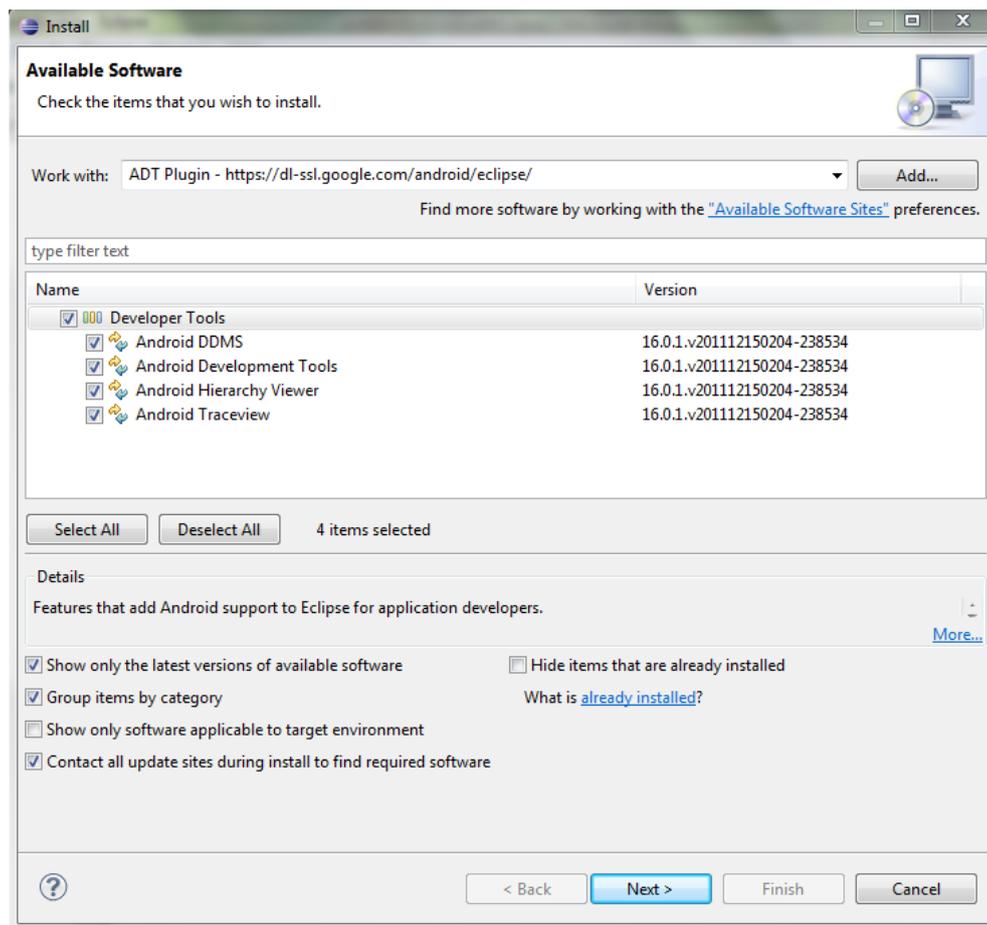


Figura 2.4: Configurazione Android in Eclipse

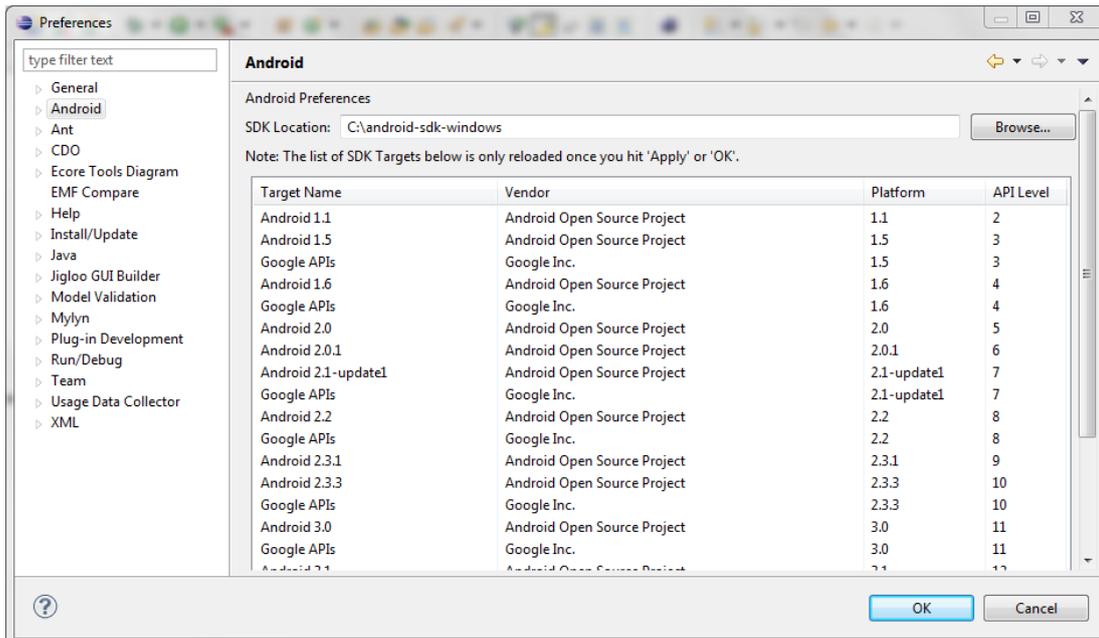


Figura 2.5: Configurazione SDK Android in Eclipse

3. Tecnologia utilizzata

Il capitolo presenterà la tecnologia relativa ai barcode bidimensionali, affrontandoli prima con un'introduzione generale e successivamente prendendone in considerazione solamente due tipi:

- QR-Code;
- Tag Microsoft.

3.1 Codici a barre: un po' di storia

I codici a barre furono sviluppati da Norman Joseph Woodland e Bernard Silver, all'epoca studenti di ingegneria dell'Università di Drexel. Nel 1948 ai due studenti nacque l'idea, dopo la richiesta di rendere maggiormente automatiche e veloci le operazioni di cassa, da parte del presidente di un'azienda del settore alimentare,.

I codici a barre unidimensionali sono un insieme di elementi grafici, disposti in un modo particolare che permette la loro lettura da strumenti dotati di sensori a scansione, così da poterne decodificare il contenuto.

All'inizio si era pensato di utilizzare il codice morse, in senso verticale e stampato per esteso, così da realizzare un insieme di linee verticale più strette o più larghe. Successivamente si iniziarono ad usare anche codici a barre ovali, ma caddero presto in disuso.

Grazie allo sviluppo della tecnologia laser, i lettori di codici a barre divennero accessibili a chiunque, in quanto il prezzo della tecnologia era minore di quello di altri strumenti utilizzati in precedenza[14].

Nell'ambito industriale la codifica maggiormente utilizzata è la 128, mentre in tutti gli altri ambiti la codifica più diffusa è quella EAN (European Article Number). Quest'ultimo tipo è utilizzato per identificare in maniera univoca i prodotti. E' internazionale ed è adottato da ben 99 paesi. E' compatibile anche con la codifica UPC (Universal Product Code) utilizzata negli Stati Uniti.

Nel codice sono presenti barre e spazi di diversi spessori, entrambi sono importanti al livello di decodifica. Ogni carattere è codificato in modo binario. E' leggibile nei due sensi, ovvero da destra verso sinistra e viceversa. La nazione è identificata dalle prime 3 cifre[15].

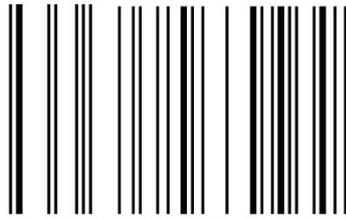


Figura 3.1: Esempio di barcode con codifica 128.



Figura 3.2: Esempio di barcode con codifica EAN.

3.2 Barcode bidimensionali

I barcode 2D (bidimensionali) sono codici a barre formati da un'immagine grafica che immagazzina delle informazioni in modo matriciale, quindi sia in senso orizzontale, come i più classici codici a barre, ma anche in senso verticale. Essi sono nati principalmente per un utilizzo in campo industriale e logistico, ma stanno avendo un enorme successo anche sul mercato di consumo: sono infatti stampabili su giornali, riviste e pubblicità. Si stanno diffondendo, oltre che su supporti fisici, anche su siti web, su piattaforme televisive, su supporti video, così da aumentare l'interazione diretta con l'utente.

Nei codici bidimensionali le informazioni sono rappresentate, negli esempi più diffusi, da un insieme di moduli quadrati bianchi e neri o triangoli colorati, posti su righe e colonne. La caratteristica matriciale permette un maggiore immagazzinamento dei dati, rispetto ad un normale barcode. Il codice a barre bidimensionale può essere visto come una serie di barcode monodimensionali molto piccoli, organizzati in fila. In questo modo l'area dedicata al codice può essere sfruttata meglio. All'interno, infatti, si possono veicolare una maggiore quantità di dati.

L'utilizzo dei barcode 2D coinvolge diversi operatori:

- i content provider, ovvero chi fornisce il contenuto immagazzinato;
- i generatori, coloro che li creano;
- gli operatori mobili;
- le agenzie pubblicitarie.

Nei codici a barre si possono inserire informazioni, come contatti, url, e-mail. All'interno dei codici però, non vi è salvato solo il contenuto richiesto dall'utente ma anche una serie di altre informazioni aggiuntive come: titolo, data di scadenza e la zona relativa alla correzione degli errori.

La correzione degli errori è molto importante perché permette una resa più duratura. Se il barcode si è leggermente deteriorato, danneggiato o coperto in parte, esso è ancora funzionante e si può decodificare, mentre per i vecchi codici a barre questo non è possibile. I codici possono essere realizzati di grandezza diversa secondo il supporto sul quale devono essere posizionati. Sono generalmente realizzati tramite una piattaforma di encoding on-line. Le tecniche di codifica sono spiegate nei paragrafi relativi ai QR-Code e ai Tag Microsoft (3.3 e 3.4). Esistono diversi tipi di barcode. I principali sono elencati nella tabella 3.1.



MaxiCode

MaxiCode è un codice 2D utilizzato molto per letture veloci. E' adatto ad esempio per ambienti industriali, come lo stoccaggio di merci, o in logistica. E' stato introdotto sul mercato nel 1992 dalla UPS Inc. Ha una media capacità di dati, contiene fino a 93 caratteri e ha inoltre una ottima densità. E' indicato se si vogliono codificare i dati binari, quindi è in grado di codificare le foto. E' funzionale anche per i normali caratteri alfanumerici.



DataMatrix

Il datamatrix si presenta con delle celle o moduli bianchi e neri a forma quadrata o rettangolare. Possono essere codificati una quantità di dati che si aggira intorno ai 2335 bytes per i caratteri alfanumerici, ai 3116 per quelli numerici. La dimensione può variare anche in base alla presenza o meno dell'algoritmo della correzione dell'errore. Il barcode è di pubblico dominio, ciò vuol dire che può essere utilizzato in applicazioni senza pagare la licenza.



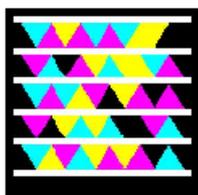
Aztec-Code

Aztec Code è stato inventato da Andrew Longacre dell'allora Welch Allyn Inc. nel 1995. Il codice è costruito su una griglia quadrata. Contiene al centro un pattern concentrico, necessario per la localizzare il contenuto. Le informazioni sono codificate aumentando la serie di quadrati intorno al pattern centrale. La sua capacità di codifica ha un range compreso tra 13 cifre o 12 caratteri alfanumerici e 3832 cifre o 3067 lettere o ancora 1914 bytes di dati.



QR-Code

Il QR-Code contiene le informazioni sia verticalmente sia orizzontalmente. Contenendole in questa maniera permette di ampliarne la possibilità di codifica arrivando a raccogliere fino a centinaia di volte la capacità di un barcode lineare. Questo codice a barre è stato sviluppato dalla Denso Wave nel 1994. E' il barcode più diffuso in Giappone e altre zone orientali, dove lo utilizza quotidianamente il 90 per cento della popolazione.



Microsoft-Tag

I Tag della Microsoft sono i barcode 2D più particolari perché sono formati da triangoli colorati. Si basano sulla tecnologia HCCB. La Microsoft offre oltre al servizio di decodifica, anche uno di reportistica e gestione delle statistiche. Sono particolari non solo per la diversa tecnologia di codifica utilizzata, ma anche per la possibilità di personalizzazione a livello grafico.

Tabella 3.1: Tipi di barcode bidimensionali

Per quanto riguarda la decodifica, sono stati creati numerosi client in grado di sfruttare le API della fotocamera degli smartphone e le API per lettori appositi, così da acquisire il codice a barre e le sue relative informazioni.

Per leggerlo, l'utente deve solamente inquadrarlo con la fotocamera, il client mostrerà il contenuto e se ad esempio è un'url, l'utente può realizzare una ricerca, evitando errori nella digitazione dell'indirizzo. Aiuta anche nel caso di un link particolarmente lungo e complesso.

Il sistema permette una maggiore interazione dell'utente con i contenuti messi in vendita dai content provider, così da favorirne anche l'acquisto d'impulso. E' davvero un forte strumento pubblicitario.

A livello di architettura generale, per decodificare i barcode ci sono due possibili strade: una definita metodo diretto e l'altra metodo indiretto.

Nel metodo diretto abbiamo un client (smartphone) che acquisisce l'immagine del barcode, la analizza in maniera stand-alone, ovvero all'interno del client stesso ed infine ottenuta l'URI, che è la stringa che identifica in maniera univoca una risorsa, e la visualizza (Figura 3.3).



Figura 3.3: Architettura con metodo diretto

Il metodo indiretto richiede un passaggio in più. In altre parole quando il codice a barre è letto dal client, esso riceve un identificativo associato al barcode. L'identificativo è risolto lato-server da un service platform. Esso restituisce il contenuto del barcode al client che può utilizzarlo a piacimento. Questo metodo è molto utile, se si ha bisogno di aggiornare il contenuto dei codici nel tempo. L'identificativo rimane lo stesso cambia solamente il contenuto a cui è associato, così da poter riutilizzare lo stesso barcode più volte[8] (Figura 3.4).

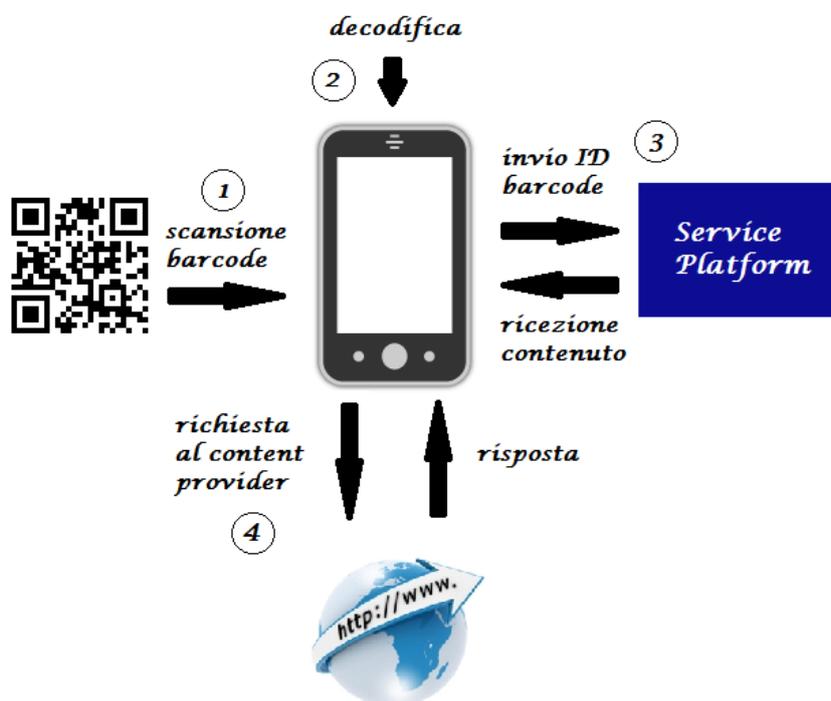


Figura 3.4: Architettura con metodo indiretto

3.3 Qr-Code

Un QR-Code è un codice a barre bidimensionale. QR sta per Quick Response. Il creatore pensava, infatti, alla possibilità di decodificarne ad alta velocità il contenuto. Un QR-Code è un codice a matrice, sviluppato e rilasciato principalmente per essere un simbolo facilmente interpretabile da attrezzature scanner apposite. Esso contiene, come accennato in precedenza, informazioni sia in senso verticale, sia in senso orizzontale.

Rispetto a un codice a barre unidimensionale, questo barcode bidimensionale può contenere un volume notevolmente maggiore d'informazioni: 7.089 caratteri numerici o 4296 caratteri alfanumerici o 2.953 byte binari.



Con 8 simboli è
codificato un byte (8 bit)

Figura 3.5: Codifica di un byte

Il QR-Code è anche in grado di utilizzare la correzione degli errori. I dati possono essere ripristinati anche quando parti significative del codice sono distorte o danneg-

giate. Essi sviluppano il codice Reed-Solomon che permette di ricostruire i dati persi, ripristinando, durante la decodifica, fino al 30 per cento delle informazioni.

I codici QR possono contenere: indirizzi internet, testi, numeri di telefono, sms e e-mail.

La struttura interna dei QR-Code è la seguente: una regione da codificare e una serie di pattern che permettono di stabilire tutti i parametri per il riconoscimento del barcode (Figura 3.6).

I pattern principali sono:

- Finder Pattern, che permette di orientare e localizzare il barcode. Il QR-Code può essere rilevato anche se ruotato di 360° .
- Timing Pattern, che permette di identificare la coordinata centrale di ogni cella del QR-Code e aiuta a rendere lineare il simbolo.
- Alignment Pattern, che permette anch'esso di rendere lineari e quindi leggibili anche immagini che sono state lette in maniera inclinata dal reader.

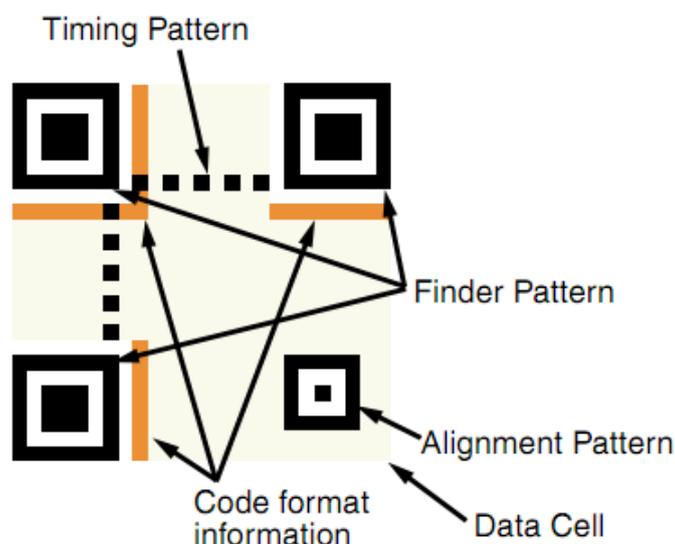


Figura 3.6: Struttura di un QR-Code

Il processo di codifica dei dati avviene in due macro fasi:

- la high level encoding in cui i dati sono convertiti in piccole unità da 8 bits;
- la low level encoding in cui le unità sono convertite in quadratini bianchi o neri.

Per quanto riguarda la decodifica, il processo è facilitato perché ogni insieme di celle forma una regione, organizzata in righe e colonne, con bordi ben delimitati per facilitarne al client l'acquisizione.

Il riconoscimento del contenuto di questi barcode può essere svolto in 5 fasi (Figura 3.7):

- scansione iniziale in scala di grigi;
- un primo riconoscimento approssimativo dei tre angoli grazie all'utilizzo del finder pattern;
- riconoscimento definitivo dei tre angoli principali;
- riconoscimento del quarto angolo non segnato;
- decodifica del contenuto.

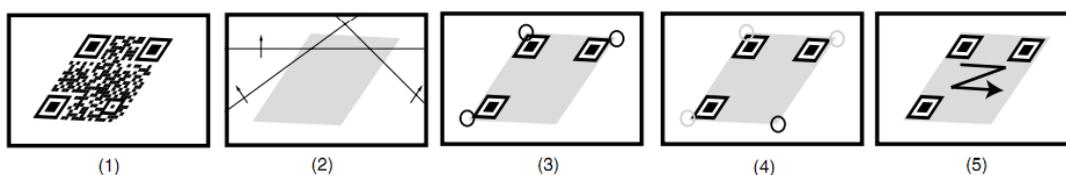


Figura 3.7: Fasi della decodifica di un Qr-Code

Senza una macchina, è impossibile per un essere umano decifrare i codici QR, ma quest'ultimi sono facilmente trattabili da apparecchiature come gli smartphone.

In Giappone, molti telefoni cellulari sono ora dotati in modo nativo di uno scanner adatto.

A seconda del tipo di dati contenuti, le applicazioni possono eseguire azioni differenti: un numero di telefono può essere composto automaticamente, un breve messaggio di testo può essere inviato, un link decodificato può essere visualizzato direttamente in un browser[9].

Riassumendo, possiamo elencare i numerosi vantaggi di questa tecnologia:

- è semplice da generare, ci sono moltissimi generatori on-line;
- ha una risposta veloce alla lettura;
- la licenza ne permette un uso libero e la possibilità di integrare questa tecnologia in applicazioni personali;
- non è richiesta alcuna connessione internet.

Ci sono anche alcuni svantaggi:

- gli utenti devono essere maggiormente informati su questo tipo di tecnologia, perché al momento sono davvero poche le persone che la conoscono e sanno come utilizzarla;
- richiede l'installazione di un'applicazione, ma nei successivi dispositivi potrebbe essere già integrata;
- ovviamente è richiesto uno smartphone e non un telefono di vecchia generazione.

3.4 Microsoft Tag

I tag sono i codici a barre bidimensionali di proprietà della Microsoft. Hanno una loro particolare tecnologia di codifica chiamata HCCB. La tecnologia è stata creata da Gavin Jancke, direttore del settore d'ingegneria alla Microsoft Research.

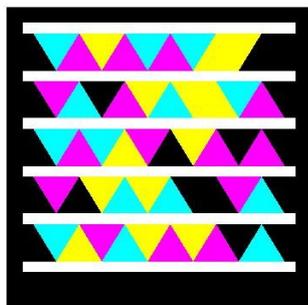
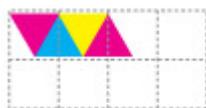


Figura 3.8: Microsoft Tag

L'idea che potrebbe scaturire è che, se si aumentano i colori, si aumentano anche il numero di informazioni codificabili. Per esempio, usando 16 colori invece di 4 (ovvero i barcode più diffusi) si potrebbero codificare il doppio dei bit nella stessa area. Ma questo non è sempre possibile perché l'utilizzo di molti colori diventerebbe un elemento di disturbo per la codifica che si basa non solo sul modo in cui la superficie riflette i colori, ma anche sul modo in cui essa stessa è illuminata. La strategia adottata dall'algoritmo di rilevamento HCCB, infatti, è adatta soprattutto ad uno schema con quattro colori. Esso ha come scopo quello di individuare quattro simboli alla volta e di assegnare ad ognuno di essi un colore secondo quelli di base che si trovano sempre in fondo ad ogni barcode (Figura 3.9 e 3.10).



Con 4 simboli (4 colori)
è codificato 1 byte.

Figura 3.9: Rappresentazione della codifica di un byte

Il tag è costituito da righe di simboli (triangoli) di colori diversi: ogni riga è separata da una linea bianca. Mentre il numero di righe in un codice HCCB può variare, il numero di moduli in ogni riga è sempre un multiplo del numero di righe. Un modulo rappresenta la base entità per la memorizzazione delle informazioni nel codice 2D.

La tecnologia HCCB posiziona intorno al tag un confine nero, circondato da un bordo bianco spesso. Il contorno nero è stato progettato per agire come punto di riferimento visivo per individuare il codice a barre in un'immagine. Il confine nella parte inferiore è più spesso dei confini sugli altri tre lati, così da far capire da che punto di vista deve essere inquadrato il codice[16].

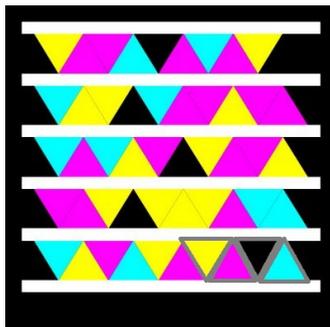


Figura 3.10: Simboli di base nei tag

I tag sono collegati a dati contenuti sul server Microsoft, garantendo un migliore rendimento on-line, con la possibilità di aggiornamento.

Come per i QR-Code, anche i tag hanno pro e contro. I vantaggi sono:

- possibilità di essere personalizzati;
- possibilità di modificare il contenuto in un qualsiasi momento;
- quantità maggiore di dati contenibili;
- maggiore velocità di codifica rispetto ai QR-Code.

Gli svantaggi di questi barcode sono invece di più:

- la è licenza proprietaria. Il brevetto appartiene alla Microsoft;
- la connessione internet deve essere sempre attiva;
- è richiesto anche in questo caso uno smartphone e non un telefono di vecchia generazione;
- è necessario scaricare l'applicazione principale della Microsoft per poterli riconoscere;
- gli utenti non sono informati sul modo di utilizzare questa tecnologia.

3.5 Differenze

Oltre alla differenza del tipo di codifica ci sono anche altre differenze di cui tener conto per conoscere a pieno questi due barcode.

3.5.1 Architettura

I QR-Code, come accennato in precedenza, sono un esempio di architettura a metodo diretto, senza necessitare di internet per ottenere il risultato della lettura.

Il Microsoft Tag Reader necessita di una connessione ad internet: il motivo è che il programma non legge direttamente il contenuto del tag ma codice identificativo. Esso è inviato al sito tag.microsoft.com ovvero il server della Microsoft dal quale riceverà la risposta. I tag sono l'esempio di un'architettura con metodo indiretto.

Il passaggio addizionale, che in certe situazioni rende i tag poco comodi o inutilizzabili, se non si ha una connessione internet sempre disponibile nello smartphone, rappresenta l'unico modo per fornire ai clienti paganti delle statistiche o dei report, come il numero di volte che un tag è stato letto o la distribuzione geografica degli accessi. A tal proposito va detto che, al primo utilizzo del software, l'utente può scegliere se condividere o no la propria posizione: una scelta che in seguito può essere cambiata dalle opzioni del programma. In ogni caso Microsoft garantisce che questo dato non è utilizzato per identificare il singolo utente.

3.5.2 Licenza

Nel 1999 Denso Wave ha rilasciato la licenza libera per l'uso dei barcode QR. Quest'ultimo è stato definito come standard ISO[13]. Negli anni si sono susseguiti diversi avvenimenti riguardanti i QR:

- nel gennaio 1999 è stato rilasciato lo standard JIS (X 0510).
- nel giugno 2000 è stato approvato lo Standard Internazionale ISO (ISO/IEC 18004).
- nel novembre 2004 Micro QR Code è stato approvato come JIS (X 0510:2004.)
- il primo settembre 2006 è stato approvato un nuovo standard ISO (ISO/IEC 18004:2006.)

I tag hanno invece la licenza proprietaria, come quasi tutti i prodotti messi in commercio dalla Microsoft.

3.5.3 Librerie

Per quanto riguarda lo sviluppo di applicazioni che ruotano intorno ai barcode, il progetto open source Zxing ha pubblicato una libreria free per i QR-Code. E' stato molto semplice trovare la libreria sul web. E' , infatti, facilmente scaricabile dal sito di GoogleCode[17]. Zxing è utilizzabile sia per la codifica che la decodifica.

Per i tag è necessario avere una libreria diversa per ogni fase: per la decodifica la libreria adatta può essere scaricata dal sito <http://mstagjavajlib.codeplex.com>[7], ed è la libreria `net.suddenelfilio.ws.tag.jar`, per la decodifica si utilizza la libreria della Microsoft, scaricabile dal sito ufficiale[1].

3.5.4 Utilizzo

I campi di applicazione legati alla connettività mobile sono davvero innumerevoli per entrambi i codici: ad esempio un monumento o un'opera d'arte possono essere associate ad un barcode che rimanda a descrizioni e informazioni quasi fosse un'audio-guida perennemente accessibile a chiunque. Si possono utilizzare per fare shopping o possono essere utilizzati per la pubblicità applicandoli a volantini, cartelloni o video.

La differenza principale in questo caso non sono i campi di applicazione, che sono all'incirca gli stessi, ma i supporti su cui possono essere realizzati. I tag hanno un vantaggio in più perchè possono vivere in ambienti diversi dalla carta, come ad esempio la stoffa.

3.5.5 Software

Per generare i codici QR-Code ci sono numerosi software sia gratuiti sia a pagamento. Molto utili sono i generatori di QR-Code on-line[11].

In un sito, ad esempio, si può scegliere che tipo di QR-Code realizzare: un'url, un numero telefonico, un contatto. Si può scegliere la grandezza, il tipo di codifica e altre opzioni.

Anche nel caso dei tag si trovano numerosi siti internet adatti allo scopo, ma il migliore è ovviamente quello interno della Microsoft[12].

Per quanto concerne la decodifica, nel mondo Android esiste uno strumento molto utile per la ricerca di applicazioni per gli smartphone e si chiama Android Market[4]. In questa piattaforma gli utenti possono trovare tantissime applicazioni di tutti i generi, sia gratis, sia a pagamento. Di applicazioni che scansionano i QR-Code, se ne possono trovare diverse. Sull'Android Market si può trovare anche l'applicazione ufficiale della Microsoft per scansionare i tag, l'unica disponibile. Se ne possono creare di personali, ma sempre sotto autorizzazione della Microsoft e senza la possibilità di metterle sul mercato, quindi solo per uso personale.

3.5.6 Personalizzazione

Una caratteristica particolare dei tag è la possibilità di crearne di personalizzati. Dal punto di vista della pubblicità e del marketing è una caratteristica accattivante (Figura 3.11).

Le aziende possono crearne diversi con il proprio logo. Se ne possono creare anche in base ad avvenimenti, a prodotti pubblicizzati e così via. I Qr-Code non supportano questa possibilità.

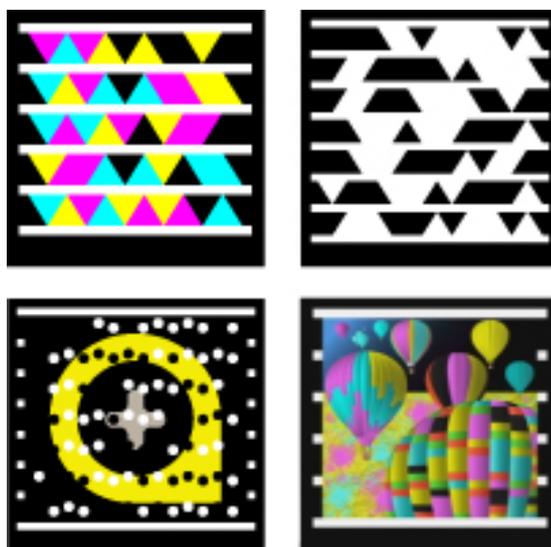


Figura 3.11: Microsoft Tag personalizzati

	Qr-Code	Microsoft Tag
Tecnologia	Quadrati bianchi e neri	HCCB (triangoli multicolore)
Quantità dei caratteri contenuti	4296 caratteri alfanumerici o 7.089 caratteri numerici	3500 caratteri per pollice quadrato
Licenza	Free	Proprietaria
Librerie	Free	Free (senza codice sorgente)
Personalizzazione	Non possibile	Possibile
Supporti	Carta e contenuti video	Carta, stoffa e contenuti video
Connessione internet	Non necessaria	Necessaria
Contenuto	Statico	Dinamico
Architettura	Metodo diretto	Metodo indiretto

Tabella 3.2: Schema riassuntivo delle differenze tra Qr-Code e Tag

4. Fasi dello sviluppo

Il capitolo spiegherà la realizzazione vera e propria del progetto, ovvero tutta la parte implementativa del progetto.

4.1 Prima fase: creazione delle applets

La prima fase consiste nello sviluppo di applets che permettano di realizzare i QR-Code e i Tag Microsoft. Le applet sono programmi progettati per essere eseguiti all'interno di applicazioni più ampie. A differenza di un'applicazione normale, le applet non possono essere eseguite direttamente dal sistema operativo.

Un'applet ben progettata può essere invocata da molte applicazioni diverse. I browser Web, sono quelle più comuni che riescono ad interpretarla. Essi sono dotati, infatti, di macchine virtuali Java. Le applet hanno, molto spesso, una piccola interfaccia utente, che le differenzia dai programmi scritti nei linguaggi di scripting (come Javascript). Un esempio molto famoso di applet è Adobe Flash.

4.1.1 QR-Code applet

L'oggetto principale di cui sia ha bisogno per la creazione dei QR-Code è una libreria contenente le classi necessarie alla codifica dei codici a barre. Quella scelta è Zxing. La libreria non è subito utilizzabile, è necessario infatti, eseguire un ulteriore passaggio per ricavarne i file .jar, ovvero i file con l'estensione delle librerie in Java. Per fare ciò si può utilizzare Apache Ant.

Apache Ant è un applicativo a riga di comando che consente l'automazione del processo di build. Ant fa parte di un progetto open source di Apache, che è comunemente impiegato per il building di applicazioni Java. Può essere scaricato dal sito di Apache[5] e può essere installato da terminale tramite il comando:

```
sudo apt -get install ant
```

Poi ci si deve posizionare all'interno delle root folder di nostro interesse, ovvero prima la cartella Core e poi la cartella Javase, che fanno parte del pacchetto Zxing. Da qui si compilano i relativi file build.xml, in questo modo:

```
ant build
```

Con questo comando saranno creati i file core.jar e javase.jar. Le due librerie devono essere incluse nel progetto in Eclipse così da poterne poi richiamare le classi necessarie per lo sviluppo dell'applicazione. La stessa procedura può essere eseguita da Windows scaricando il file .zip di Ant e scoppattandolo opportunamente.

Successivamente si deve inserire una variabile d'ambiente chiamata ANT_HOME, che ha come valore il percorso in cui Apache Ant è stato scompattato. Infine si deve

aggiungere alla variabile d'ambiente Path ;%ANT_HOME%\bin. Ora la configurazione in Windows è pronta, portandoci all'interno della cartella core e digitando da terminale:

```
ant -f core/build.xml
```

sarà creato il file core.jar. La stessa cosa è da ripetere per il file build.xml della cartella Javase.

A questo punto le librerie sono pronte e possono essere importate nel progetto in Eclipse. Per fare ciò ci si posiziona nella cartella del progetto all'interno del workspace di Eclipse e si aggiungono le due librerie. Successivamente, all'interno della piattaforma di sviluppo, si può poi cliccare con il tasto destro sulle librerie, che saranno ora visibili anche da Eclipse e scegliere Aggiungi al Build Path.

La prima parte dell'applicazione riguarda la GUI (Graphical User Interface). E' una grafica molto semplice composta da:

- una textview dove inserire l'url da codificare;
- una combo box per scegliere la grandezza;
- un'immagine che mostra il risultato dopo aver premuto il bottone relativo;
- un bottone che permette di salvare l'immagine in una directory del proprio computer.



Figura 4.1: Home Qr-Code Applet

Per la parte relativa alla creazione del QR-Code, che è la parte fondamentale, il primo oggetto da ottenere è un array di byte con la codifica UTF-8 dall'url scritta dall'utente. Ora entrano in gioco le librerie:

- si crea un oggetto di tipo Bitmatrix che rappresenterà la matrice del QR-Code;
- la matrice è creata tramite un oggetto MultiFormatWriter(). Quest'oggetto permette la codifica in formati diversi. Ovviamente noi scegliamo il formato QR-Code;
- la classe QRCodeWriter, che ora chiameremo solo writer per comodità, è la classe che inizializza l'algoritmo della codifica. Questa codifica utilizza diversi

parametri: l'url, l'altezza e la larghezza della matrice da creare. All'url è aggiunto un vettore di bit che rappresentano l'header e la data. Poi sono aggiunti i bit per la correzione dell'errore Reed-Solomon;

- infine il writer restituirà un array bidimensionale di valori come in figura 4.2, dove ogni simbolo corrisponde ad 1, e gli spazi bianchi a 0, come nel più classico codice binario;
- a questo punto utilizzando la libreria `javase.jar` si può passare alla creazione dell'immagine vera e propria, come in figura 4.3.

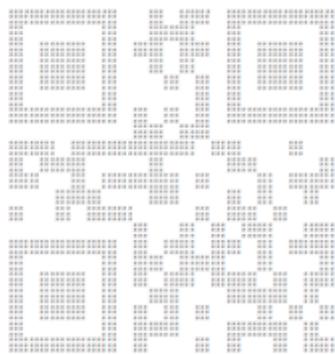


Figura 4.2: Rappresentazione Qr-Code in bit



Figura 4.3: Qr-Code disegnato

4.1.2 Tag Microsoft applet

La seconda applet permette la creazione di tag. La Microsoft, com'era probabile aspettarsi, anche se lascia qualcosa a disposizione dell'utente in maniera gratuita, vuole comunque assicurarsi un controllo su di esso. In questo caso, infatti, la libreria per la codifica può essere scaricata gratuitamente, ma durante il suo utilizzo Microsoft richiede un codice d'identificazione per permetterne il funzionamento.

L'identificativo o key, è necessario per svolgere per conto delle aziende, controlli statistici sui tag creati, ad esempio per calcolare il numero di scansioni da parte degli utenti, accumulando così indizi di mercato per se stessa, ma anche per le aziende, sue clienti, che con i tag si pubblicizzano.

Durante un primo sguardo alla libreria, ci accorgiamo subito che è necessaria una connessione internet sempre attiva per utilizzarla e soprattutto è necessario il codice identificativo come spiegato in precedenza, che Microsoft rilascia solo su richiesta. Il codice può essere ottenuto registrandosi al sito della Microsoft[1]. Per farlo bisogna avere un account Windows.

Una volta loggati, nella sezione Developers si può eseguire la richiesta della key, compilando il form di figura 4.4, specificando vari parametri: chi siamo, azienda o privato, qual è lo scopo dell'applicazione, ecc.

Una volta inviata la richiesta, il giorno successivo Microsoft invia un'e-mail di risposta con il codice da usare, il quale è composto da 32 cifre.

L'iscrizione al sito della Microsoft permette di avere una sezione di amministratore dei propri tag, questo garantisce una visione generale di tutti quelli che sono stati

Microsoft+ tag WHAT IS TAG? TAG IN ACTION COMMUNITY RESOURCES MY TAGS Chiara | SIGN OUT search

Tag Manager TAGS CATEGORIES REPORTS

Microsoft Tag Creation API Access Request

The Microsoft Tag Creation API allows you to programmatically access many of the same operations used on the Microsoft Tag Web site. Using Visual Studio and the .NET Framework, you can write programs to create, modify, and otherwise manage Tags for your business - including bulk Tag creation and updates - without having to use the Microsoft Tag Web site interface. The information requested below will be used to create your Microsoft Tag Web Service account. Once your request is approved, your access token will be sent to the Email address you provide. We may also contact you periodically to inform you about updates to the service.

Tell us about yourself...

Last Name *

First Name *

Email (Contact Email) *

Verify your Email *

Company *

Company Web Site *

Tell us about your application...

Application Name *

How many Tags you anticipate creating? *

Will your application create Tags on behalf of 3rd parties? * Yes No

Does your application runs advertising? * Yes No

Description (Describe how you will be using Tag and why you need access to the Web Service.) *

I agree to the [Microsoft Tag Terms Of Use.](#) *

Figura 4.4: Richiesta dell'identificativo alla Microsoft

creati, con la possibilità di poterne poi in futuro, modificare il contenuto, senza dover ristampare il tag, vantaggio non di poco conto.

Ottenuta la key, si può iniziare l'implementazione.

La grafica è la stessa utilizzata per l'applets dei QR-Code, l'unica differenza è che non c'è la possibilità di scegliere a piacimento la grandezza da dare ai tag.

La prima fase è di autenticazione al server della Microsoft tramite l'Api Key fornitaci in precedenza. Senza questo passaggio sarebbe impossibile compiere qualsiasi altra operazione.

Il tag per essere creato deve avere determinati parametri obbligatori:

- il nome della categoria, in cui il tag può essere raggruppato, sempre per statistiche e report;
- il titolo ovvero la parte che sarà rappresentata nel tag;
- la data di creazione;
- il contenuto.

Come spiegato al paragrafo 3.5.1, i tag Microsoft si basano sul metodo indiretto, grazie al quale possono essere aggiornati in un secondo momento. Questo accade perché è il titolo che viene codificato nel tag e non direttamente il contenuto. Quest'ultimo sarà associato al titolo all'interno del server Microsoft. Per evitare disguidi, nell'applet è stato fatto sì che il titolo del tag equivalga al suo contenuto. Il tag inoltre deve essere attivato, è importante perché, avendo una scadenza, potrebbe avere la necessità di essere riattivato. Per creare il tag, come per i QR-Code, dall'array di byte risultante dalla codifica del titolo, è creata un'immagine, grazie alla tecnologia HCCB. L'immagine sarà visualizzata prima del salvataggio, così da constatare se il tag è stato realizzato nel modo corretto.

4.1.3 Permessi e firma

Le applet come sappiamo sono adatte all'utilizzo in internet. A volte ci sono situazioni in cui sono richiesti dei permessi per eseguire determinate azioni. Per risolvere questo problema si può utilizzare una classe java chiamata `AccessController` che con il suo metodo `doPrivileged()` permette di ottenere tutte le autorizzazioni necessarie a eseguire i metodi delle applet su computer remoti. In particolare è necessario posizionare il codice che compie l'azione che necessita del permesso, proprio all'interno del metodo `doPrivileged()`. Nell'applet dei tag, ad esempio, è necessario aprire una socket verso il server Microsoft, per permettere l'utilizzo del web service che risponde alle chiamate dei client. Ottenuti i permessi, affinché le applet possano essere pubblicate sul server, è necessario un ultimo passaggio, ovvero devono essere firmate. Se le applet non sono firmate, si può incappare nella generazione di molte eccezioni, poiché non si ha la possibilità di effettuare accessi a particolari zone dei computer remoti: ad esempio al disco rigido dell'utente per creare e salvare un file. La firma risolve questo tipo di problemi. I passaggi per firmare un'applet sono i seguenti:

- aprire il terminale e creare un keystore, in altre parole un certificato contenente tutte le informazioni riguardanti l'autore, con questo comando :
`keytool -genkey -keystore mykeystore -alias me`
dove *mykeystore* rappresenta il nome del certificato che si andrà a creare e *me* un alias da dover dare alla firma. Poi è necessario compilare le domande che vengono richieste (Figura 4.5);
- associare il keystore con il file .jar, che contiene l'intera applet, tramite il comando `jarsigner`. L'importante è che entrambi si trovino nella stessa cartella. Sempre da terminale ponendosi nella cartella in questione si deve digitare:
`jarsigner -keystore mykeystore applet.jar me`
come in figura 4.6.

La procedura può essere eseguita allo stesso modo su Windows.

```
chiara@chiara-laptop:~$ keytool -genkey -keystore mykeystore -alias me
Enter keystore password:
Re-enter new password:
What is your first and last name?
  [Unknown]: Chiara Cini
What is the name of your organizational unit?
  [Unknown]: Unicam
What is the name of your organization?
  [Unknown]: Unicam
What is the name of your City or Locality?
  [Unknown]: Camerino
What is the name of your State or Province?
  [Unknown]: Italy
What is the two-letter country code for this unit?
  [Unknown]: IT
Is CN=Chiara Cini, OU=Unicam, O=Unicam, L=Camerino, ST=Italy, C=IT correct?
  [no]: y

Enter key password for <me>
  (RETURN if same as keystore password):
Re-enter new password:
```

Figura 4.5: Screenshot per la creazione del keystore

```
chiara@chiara-laptop:~$ cd Desktop
chiara@chiara-laptop:~/Desktop$ jarsigner -keystore mykeystore core.jar me
Enter Passphrase for keystore:

Warning:
The signer certificate will expire within six months.
chiara@chiara-laptop:~/Desktop$
```

Figura 4.6: Screenshot per la firma dell'applet

4.2 Seconda fase: creazione sito internet

Il sito può essere suddiviso concettualmente in tre sezioni:

- una sezione informativa ovvero quella contenente le informazioni degli studi dei professori e delle aule;
- una sezione dedicata alle applets;
- una sezione amministratore.

Il sito è interamente formato da file .php. Php è un modulo che è utilizzato dai web server per permettere la creazione di pagine dinamiche. Un file .php può essere composto sia da codice html sia da istruzioni php. Quest'ultime sono eseguite prima che la pagina sia inviata all'utente, generando un codice in completo html. Come abbiamo detto, le informazioni riguardano sia i professori sia le aule. I professori sono stati raggruppati in base allo studio e hanno una pagina dedicata con tutte le informazioni necessarie ovvero: ruolo, sito internet, orario di ricevimento ed altro (Figura 4.7).

Per le aule invece, si ha un'immagine e una piccola descrizione, ma la pagina può essere sempre ampliata con altre informazioni, come gli orari delle lezioni o altro (Figura 4.8).

Studio Flavio Corradini

Prof. Flavio Corradini

Ruolo: Rettore dell'Universita' degli studi di Camerino

Telefono:

e-Mail: flavio.corradini@unicam.it

Ubicazione: Polo informatico "Carla Ludovici" Secondo Piano

Sito Web: <http://www.cs.unicam.it/corradini/>

Ricevimento:

Figura 4.7: Rappresentazione degli studi nel sito

L'url di ognuna di queste pagine sarà il contenuto dei barcode. All'interno del sito c'è una piccola parte alla quale si può accedere solamente mediante password ed è la parte riservata all'amministratore. Esso può modificare sezioni del sito in maniera dinamica.

Per aggiungere le applets all'interno del sito, bisogna prima creare un archivio con estensione .jar completo. Con completo si intende comprensivo di:

- classi già compilate dell'applet, quindi con estensione .class;
- relativa libreria scompattata, quindi non come archivio.

Aula Magna: Sauro Tulipani



Polo informatico "Carla Ludovici" Primo Piano
L'aula Magna e' l'aula principale del dipartimento.
Vi si tengono sia le lezioni sia importanti riunioni e convegni.

Figura 4.8: Rappresentazione delle aule nel sito

Successivamente è necessario firmarlo, seguendo l'iter spiegato al paragrafo 4.1.3. Ora si deve porre questo nuovo archivio all'interno della cartella del web server che contiene il file .php che lo richiama tramite queste poche righe di codice html:

```
<applet code=NomeClasseMain.class ARCHIVE=ArchivioFirmato.jar  
height=500 width = 800> </applet>
```

dove *NomeClasseMain.class* è la classe principale dell'applet e *ArchivioFirmato.jar* è l'archivio finale firmato.

A questo punto si può passare alla realizzazione dell'ultima fase ovvero la decodifica dei barcode tramite smartphone Android.

4.3 Terza fase: creazione di applicazioni per Android

La terza fase consiste nell'implementazione di due applicazioni per gli smartphone con sistema operativo Android che permettano la decodifica veloce dei barcode.

4.3.1 QR-Code Reader

L'applicazione ha come scopo quello di decodificare l'immagine di un QR-Code, dopo che è stato inquadrato e immortalato con la fotocamera dello smartphone. A seconda del contenuto letto, l'applicazione lancerà il browser per visualizzare l'indirizzo internet richiesto oppure mostrerà semplicemente l'informazione. Tutte le scansioni verranno salvate in una bacheca, così da poterle avere sempre disponibili.

Quando si vanno a scansionare i codice a barre, la prima cosa da impostare è una SurfaceView ovvero la view che sfrutta l'inquadratura della fotocamera. Al premere di un bottone viene scattata una foto al barcode. La foto è decodificata usando ancora la libreria Zxing.

L'immagine presa in considerazione è utilizzata sottoforma di un array di byte. Dall'array di byte è ricreata una bitmap, in altre parole un'immagine nel formato RGB.

RGB è la tecnica di composizione di un immagine in base alla presenza di soli tre colori rosso, verde, blu. L'immagine è quindi riconvertita in un array 2D in scala di grigi. La matrice ottenuta è scansionata in ordine di 8x8 pixel in una griglia 5x5, quindi in tutto una matrice di 40 x 40 pixel.

Il risultato di questa scansione ritorna una matrice di punti neri, che segnala un punto con il valore "true". La matrice è trasformata in una matrice di bit, così da poter essere decodificata in base alla codifica ASCII.



Figura 4.9: Screenshot dell'applicazione QR-Code Reader

L'algoritmo prende un blocco di dati e aggiunge i bit di parità. Il reader processa ogni blocco e cerca di correggere gli errori se ce ne sono. Il numero di errori, che può essere riscontrato, è diverso a seconda delle caratteristiche del codice. Tutte le volte

che si esegue una scansione, il dato ottenuto è salvato in un database SQLite, con il quale è popolata la bacheca, ovvero una ListView.

C'è inoltre una piccola parte dedicata alle impostazioni, dove si può decidere, in caso il dato letto dal reader fosse un url, di saltare il passaggio di visualizzazione dell'indirizzo, ma di passare direttamente al browser. In quest'algoritmo è compresa la correzione degli errori Reed-Solomon. Reed-Solomon è un codice a correzione d'errore basato sui codici a blocco.

4.3.2 Microsoft Tag Reader

Per sviluppare il reader dei tag della Microsoft, è necessaria una libreria diversa da quella usata fino ad ora per la loro codifica. Questa libreria può essere ottenuta dal sito della Microsoft. Se è stata inviata in precedenza la mail per il primo codice identificativo, si è ottenuto anche il secondo, quello che si deve utilizzare per la scansione. Quest'ultimo non è scritto nella mail ma bisogna ritrovarlo in una sezione particolare del sito[6].

Dopo il log-in, nella sezione Key manager, troveremo la Tag Scanning SDK Key. Come per l'applet, anche per il reader è richiesto un codice identificativo. A questo punto, sempre nella stessa sezione, possiamo scaricare l'Android SDK.

La libreria è scaricabile gratuitamente, ma non è possibile usufruire del codice sorgente. La Microsoft mette a disposizione due possibili esempi, insieme alla SDK, che aiutano a comprendere i metodi da utilizzare per effettuare la scansione dei tag e ricevere le informazioni dal server Microsoft.

Le funzionalità generali sono ovviamente simili all'applicazione dei QR-Code. La parte riguardante la scansione del tag è identica, solo che in questo caso basta inquadrarlo senza scattare la foto. La procedura della ricezione del contenuto è differente, infatti, viene inviata una richiesta http GET al server della Microsoft con il titolo del tag appena scansionato. Come risposta si otterrà un contenuto in formato XML. Il dato ricevuto sarà interpretato e visualizzato nel browser.

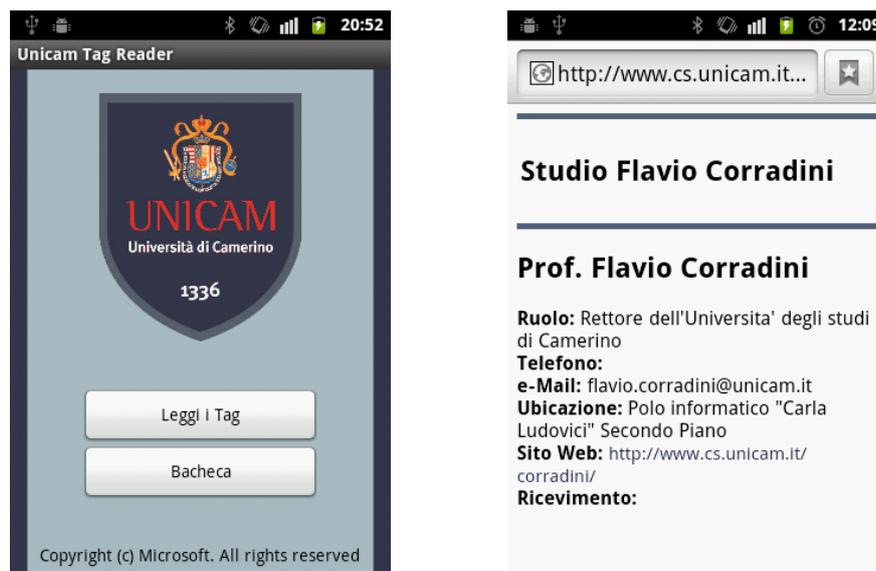


Figura 4.10: Screenshot dell'applicazione Tag Reader

Grazie al fatto che quando si legge un tag, si esegue un GET ad un web service, Microsoft può facilmente tenere conto di quante volte quel tag è stato letto, e può redigere report e statistiche da fornire ai propri clienti.

Anche in quest'applicazione abbiamo la presenza di una bacheca che visualizza tutti i dati salvati dalle precedenti scansioni.

4.3.3 Adattabilità

L'applicazione è stata testata su due diversi dispositivi Android, con diversa risoluzione:

- smartphone Huawei U8650 con display di 3.5" e fotocamera da 3.2 megapixel;
- tablet HTC con display 7" e fotocamera da 5 megapixel.

Nel primo dispositivo l'applicazione è perfettamente funzionante. Infatti, su un dispositivo non eccessivamente grande non sono stati riscontrati problemi. Su un dispositivo con uno schermo più ampio, come ad es. il tablet, l'applicazione non è funzionante.

Il problema è stato riscontrato nelle impostazioni riguardanti la fotocamera e l'immagine scattata. Questo richiede quindi, una modifica del codice proprio su questi particolari. Il problema è valido per entrambe le applicazioni.

Una soluzione possibile è diminuire lo spazio destinato all'inquadratura del barcode anche sui tablet, così da fornire un'immagine di dimensioni identiche ad una scattata con uno smartphone, così da essere accettabile per la decodifica.

Non avendo a disposizione un dispositivo intermedio, non è possibile stabilire quale sia il limite di grandezza e risoluzione del display, fino a cui le applicazioni possono funzionare.

5. Conclusioni

In questa tesi è stato analizzato il funzionamento della tecnologia dei codici a barre bidimensionali, dalla loro struttura interna, all'implementazione di applicazioni che possano interpretarli. Sono state elencate le principali differenze tra i barcode 2D in generale e tra la tecnologia utilizzata dalla Microsoft, ovvero l'HCCB e la tecnologia a quadrati neri dei QR-Code.

L'adattabilità delle applicazioni sui diversi dispositivi è da migliorare, ma è un qualcosa che, in seguito, può essere risolto.

Per quanto riguarda i QR-Code, una nuova prospettiva sono i contextual QR-Codes. La particolarità di questi barcode è la possibilità di modificare il contenuto statico in base al contesto ovvero alle informazioni private dell'utente, in particolare ciò può essere fatto tramite dei reader appositi, che riescono a decifrare il codice xml contenuto all'interno del barcode ed utilizzarlo nel modo appropriato.

Per le industrie, la Denso Wave, ha continuato ad investire sulla ricerca, ed ha realizzato l' IQR-Code. Quest'ultimo è veramente adatto alle esigenze delle industrie: ha dimensioni ridotte, può essere anche di forma rettangolare, perfetto per le superfici strette e lunghe di molti prodotti in circolazione e può codificare ancora più informazioni dei normali QR.

Per i tag Microsoft, il loro più grande vantaggio, che ne aumenterà l'utilizzo, è la personalizzazione. Le aziende, infatti, possono creare un tag con il logo personalizzato. Dal punto di vista della pubblicità e del marketing è un notevole punto in più, perchè come si dice anche l'occhio vuole la sua parte.

Un'esigenza di mercato riguardo i barcode richiede che essi siano sempre più piccoli per occupare meno spazio possibile e per poter essere collocati in maniera più semplice sui supporti materiali e non. Questo può essere fattibile fino ad un certo limite, in quanto, da un certo punto in poi i barcode non saranno più leggibili dalle fotocamere degli smartphone che supportano fino ad una certa risoluzione, soprattutto se ciò che è memorizzato prevede un'informazione molto lunga.

Come ben sappiamo, i codici a barre facilitano di gran lunga l'accesso alle informazioni. Le azioni possibili sono ormai tantissime e aumenteranno in futuro.

Ora come ora, i barcode sono utilizzati per pubblicizzare qualsiasi tipo di oggetto, dai prodotti, ai film, alle riviste, ecc.

I barcode bidimensionali potranno essere utilizzati, come già accade nel Giappone e zone orientali, in altri campi:

- per effettuare la spesa, ovvero aggiungendo l'oggetto ad un carrello virtuale;
- all'interno di ristoranti, per permettere una maggiore interazione del cliente con i piatti cucinati;

-
- all'interno di musei per avere più informazioni riguardo agli oggetti esposti;
 - sui monumenti, così da fornire ai visitatori molte più informazioni di quante se ne possano scrivere su una classica didascalia.

La tecnologia mobile è entrata prepotentemente nella vita quotidiana ed è ormai alla portata di tutti. E' molto semplice quindi, prevedere una grande diffusione dell'utilizzo di questi barcode nel futuro.

Elenco delle figure

1.1	Rappresentazione grafica del progetto	8
2.1	Schermata iniziale di Android	11
2.2	Architettura di Android	12
2.3	Activity lifecycle	13
2.4	Configurazione Android in Eclipse	14
2.5	Configurazione SDK Android in Eclipse	15
3.1	Esempio di barcode con codifica 128.	17
3.2	Esempio di barcode con codifica EAN.	17
3.3	Architettura con metodo diretto	19
3.4	Architettura con metodo indiretto	20
3.5	Codifica di un byte	20
3.6	Struttura di un QR-Code	21
3.7	Fasi della decodifica di un Qr-Code	22
3.8	Microsoft Tag	23
3.9	Rappresentazione della codifica di un byte	23
3.10	Simboli di base nei tag	24
3.11	Microsoft Tag personalizzati	26
4.1	Home Qr-Code Applet	29
4.2	Rappresentazione Qr-Code in bit	30
4.3	Qr-Code disegnato	30
4.4	Richiesta dell'identificativo alla Microsoft	31
4.5	Screenshot per la creazione del keystore	33
4.6	Screenshot per la firma dell'applet	33
4.7	Rappresentazione degli studi nel sito	34
4.8	Rappresentazione delle aule nel sito	35
4.9	Screenshot dell'applicazione QR-Code Reader	36
4.10	Screenshot dell'applicazione Tag Reader	37

Elenco delle tabelle

3.1	Tipi di barcode bidimensionali	18
3.2	Schema riassuntivo delle differenze tra Qr-Code e Tag	27

Bibliografia

- [1] 2d barcode and mobile tag. <http://tag.microsoft.com/home.aspx>.
- [2] Adt plug-in per eclipse. <https://dl-ssl.google.com/android/eclipse>.
- [3] Android developers. <http://developer.android.com/sdk>.
- [4] Android market. <https://market.android.com/>.
- [5] Apache web site. <http://ant.apache.org/>.
- [6] Microsoft developer page. <http://tm.tag.microsoft.com/developer>.
- [7] Microsoft tag java library. <http://mstagjavalib.codeplex.com>.
- [8] G. P. Cecilia Corbi, Stefania Lisa. Codici a barre bidimensionali: tecnologia e campi applicativi. *Notiziario tecnico- Telecom Italia*, 2009.
- [9] L. A. H. Eisaku Ohbuchi, Hiroshi Hanaizumi. Barcode readers using the camera device in mobile phones. *Proceedings of the 2004 International Conference on Cyberworlds*, 2004.
- [10] P. Informatico. *Android Programming*. Edizioni Master.
- [11] Kaiwa. Generatore di qr-code. <http://qrcode.kaywa.com/>.
- [12] Microsoft. Generatore di microsoft tag. <http://www.webtivitydesigns.com/tools/microsoft-tag-generator.aspx>.
- [13] Wikipedia. Codice qr. http://it.wikipedia.org/wiki/Codice_QR.
- [14] Wikipedia. Codici a barre. http://it.wikipedia.org/wiki/Codice_a_barre.
- [15] Wikipedia. European article number. http://it.wikipedia.org/wiki/European_Article_Number.
- [16] Wikipedia. High capacity color barcode. http://en.wikipedia.org/wiki/High_Capacity_Color_Barcode.
- [17] Zxing. Multi-format 1d/2d barcode image processing library with clients for android, java. <http://code.google.com/p/zxing/>.