

Escola Universitària Politécnica de Mataró

Centre adscrit a:



**UNIVERSITAT POLITÈCNICA
DE CATALUNYA**

**Enginyeria Tècnica de Telecomunicacions
especialitat “Telemàtica”**

**Desenvolupament d’una aplicació mòbil
per OS Android
“CallMemory”**

Memòria

**Daniel Sánchez Ciurana
PONENT: Pere Barberán Agut**

PRIMAVERA ANY 2011



**TecnoCampus
Mataró-Maresme**

Dedicatòria

Per la meva àvia, que per molt que ens oblidi al seu cap, sempre ens recordarà al seu cor.

Resum

L'objectiu del projecte és la creació d'una aplicació, desenvolupada per a telèfons mòbils amb Sistema Operatiu Android, que ens permetrà la lectura de codis bidimensionals (Codis QR), un tipus de codi de dues dimensions que conté informació codificada, per exemple un telèfon mòbil, una URL, una imatge...

Aquesta aplicació facilitarà, la tasca habitual de trucar a una persona, mitjançant la captació de codis QR, amb la càmera integrada del dispositiu mòbil. Aquesta aplicació anirà destinada/orientada a persones amb disminució de memòria, facilitant el procés a un simple clic.

Resumen

El objetivo del proyecto es la creación de una aplicación, desarrollada para teléfonos móviles con Sistema Operativo Android, que nos permitirá la lectura de códigos bidimensionales (Códigos QR), un tipo de código de dos dimensiones que contiene la información codificada, por ejemplo un teléfono móvil, una URL, una imagen...

Esta aplicación facilitará, la acción habitual de llamar a una persona, mediante la captación de códigos QR, con la cámara integrada en el dispositivo móvil. Esta aplicación irá destinada/orientada a personas con disminución de memoria, facilitando el proceso a un simple clic.

Abstract

The project objective is to create an application developed for mobile phones with Android Operating System which allows the reading of two dimensional codes (QR codes), a type of two-dimensional code that contains the encoded information, such as a mobile phone, a URL, an image ...

This application will facilitate the common action of calling a person by the uptake of QR codes, with the integrated camera on your mobile device.

This application will be designed/oriented to people with decreased memory, making only a click.

Índex.

Índex de figures.....	III
Índex de taules.....	V
Glossari de termes.....	VII
1. Introducció.....	1
1.1. Objectiu del projecte.....	1
1.2. Objectiu Social.....	1
1.3. Objectiu personal.....	2
1.4. Estat de l'art.....	2
2. Una mica d'Història.....	3
2.1.- Breu història dels codis referencials.....	3
3. Què és un codi de barres.....	7
3.1.-Tipus de codis de barres.....	8
3.1.1.-Codis de barres lineals (1D).....	8
3.1.2.- Codis de barres bi-dimensionals (2D).....	15
3.2.- Comparatives 1D, 2D.....	20
4. Què són els codis QR.....	23
4.1.- Qui els desenvolupa.....	23
4.2.- Estructura codi QR.....	25
4.3.- Informació emmagatzemada.....	25
4.4.- Correcció d'errors.....	26
4.5.- Com crear i llegir un codi QR.....	26
4.6.- Exemples d'utilització al món real.....	28
5. Evolució de la telefonia mòbil i dels Sistemes Operatius.....	31
5.1.- Breu història de la telefonia.....	31
5.2.- Evolució de la tecnologia mòbil.....	31
5.3.- Sistemes Operatius Mòbils.....	34
5.4.- Comparativa Sistema Operatiu (IO's 4 Apple, Android, Windows 7, Symbian)	
.....	38
5.5.- Evolució del Sistema Operatiu Android.....	40
6. Eines utilitzades per programa Android.....	43
6.1.- Llenguatges de programació.....	43
6.1.1.- JAVA.....	43

6.1.2.- XML	44
6.2.- Entorn d'execució	45
6.3.- Software	45
6.3.1.- Eclipse	46
6.3.2.- ADT plug-in	46
6.3.3.- SDK Android + API.....	47
6.4.- Extres.....	48
6.4.1.- BARCODE Scanner.....	48
6.4.2.- ZXing	49
7. Desenvolupament de l'Aplicació.	51
7.1.- Requeriments de Desenvolupament.	51
7.1.1.- Requeriments del Dispositiu Mòbil.	52
7.2.- Instal·lació Eclipse.	53
7.3.- ADT Plug-in (Android Development Tools).....	56
7.3.1.- Configurar el plug-in ADT.....	58
7.4.- Creació KVM Android.....	61
8. Creació del projecte.	63
8.1.- Creació i estructura del Projecte.....	63
8.2.- Arquitectura de classes.....	65
8.3.- Casos us.....	66
8.4.- Fitxers Values.....	67
8.5.- Layouts	68
8.6.- Codi JAVA.....	69
8.6.1.- SplashScreen	69
8.6.2.- Crida BarCode Scanner.....	70
8.6.3.- Funció Call (String string)	71
9. Possibles ampliacions del projecte.	73
10. Conclusió.....	75
10.1.- Conclusió Personal.....	75
11. Bibliografia.....	77
12. Annexos.....	79
12.1.- Evolució històrica Sistemes Operatius Mòbils.	79
12.2.- Pressupost.....	80
12.3.- Codi final de l'aplicació.....	81

Índex de figures.

Figura 2.1. Fulla Patent 1985035	3
Figura 2.2. Fulla Patent 2020925	4
Figura 2.3. Fulla Patent 2612994	4
Figura 2.4. Primer Codi de Barres	5
Figura 3.1. Procés de descodificació	8
Figura 3.2. Codi UPC	8
Figura 3.3. Codi UPC-A	9
Figura 3.4. Codi UPC-E	10
Figura 3.5. Codi EAN-13	10
Figura 3.6. Codi EAN-8	11
Figura 3.7. Code 39	12
Figura 3.8. Simbologia Code 39	12
Figura 3.9. Exemple Mòdul Codi 128	13
Figura 3.10. Code 128	13
Figura 3.11. Code 93	14
Figura 3.12. Codabar	14
Figura 3.12. Codi Bookland	15
Figura 3.13. Codi PDF 417	16
Figura 3.14. Codi DATAMATRIX	17
Figura 3.15. Codi QR	19
Figura 3.15. Codi BEETAGG	20
Figura 3.16. Comparativa Codis 1D i 2D	20
Figura 4.1. Evolució utilització codis QR	24
Figura 4.2. Estructura Codi QR	25
Figura 4.3. Exemple de Codificació de text	26
Figura 4.4. Publicitat BUS Figura 4.5. Utilitat Google Maps	28
Figura 4.6. Anunci La Vanguardia Figura 4.7. Portada The Sun	28
Figura 4.8. Denominació Ampolla de vi Figura 4.9. Codi promocional Pepsi	29
Figura 4.10. Utilització paqueteria. Figura 4.11. Especificacions Electrónica	29
Figura 5.1. Telèfons amb els seus Sistemes Operatius	34
Figura 5.3. Gràfic de Ventas dels diferents Sistemes Operatius al Món	39

Figura 5.4. Gràfic de Ventas dels diferents Sistemes Operatius a Espanya	39
Figura 5.5. Interfície Gràfica Android Cupcake	40
Figura 6.1. Imatge Software Eclipse	46
Figura 6.2. Imatge Aplicació BarCode Scanner	48
Figura 6.3. Logotip ZXing.....	49
Figura 7.0. Requeriments desenvolupament Aplicació	51
Figura 7.1. Pàgina inici www.eclipse.org.....	53
Figura 7.2. Versió del Software a descarregar	54
Figura 7.3. Instal·lació Màquina Java.	54
Figura 7.4. Primera execució de Eclipse	55
Figura 7.5. Escollir el directori del Workspace	55
Figura 7.6. Càrrega de Eclipse.....	55
Figura 7.7. Entorn principal de Eclipse	56
Figura 7.8. Introducció del Plug-in ADT	57
Figura 7.9. Especificació URL.	57
Figura 7.10. Selecció dels complements desitjats.	58
Figura 7.11. Instal·lació complements ADT.	58
Figura 7.12. Pàgina principal Android Developers	59
Figura 7.13. Especificació del directori de l'instal·lació.....	59
Figura 7.14. Diferents SDK i API's Android	60
Figura 7.15. Accés a Preferències de Eclipse.....	60
Figura 7.16. Especificació del directori del SDK Android.....	61
Figura 7.17. SDK i API's compatibles.....	61
Figura 7.18. Pestanya Menú Windows.....	62
Figura 7.19. Finestra Android SDK	62
Figura 7.19. Creació AVD.....	62
Figura 8.1. New Android Project.....	63
Figura 8.2. Estructura Projecte Android	63
Figura 8.3. Estructura /res/ CallMemory	64
Figura 8.4. Estructura /gen/ Callmemory	64
Figura 8.4.b. Diagrama de classes	65
Figura 8.5. Layout Splash.....	68
Figura 8.6. Layout Main	68

Índex de taules.

Taula 3.1.Comparativa Codis Unidimensionals	15
Taula 3.2.Comparativa Codificació diferents Codis	21
Taula 5.1. Evolució Tecnològica telefonia mòbil	33
Taula 5.2. Comparativa Tecniques Sistemes Operatius	38

Glossari de termes.

API	“Applicaton Programming Interface”, Interfície de programació d’aplicacions.
ASCII	“American Standard Code for Information Interchange”, Codi Estadounidense Estàndard pel Intercanvi d’Informació.
CDMA	“Code Division multiple Acces”, Accés Multiple per divisió de codi.
GPRS	“General Packet Radio Services”, tècnica de commutació de paquets integrable amb l’estructura actual de les xarxes GSM.
GSM	“Global System for mobile Communication”
OS	“Operating System”, Sistema Operatiu.
KVM	“Kernel Virtual Machine”.
LED	“Light-Emitting Diode”, diode emissor de llum.
QR	“Quick Response”
SDK	“Software Development Kit”, Eines de Desenvolupament de Software.
Wifi	“Wide Fidelity”, Ampla fidelitat.
XML	“Extensible Markup Language”

1. Introducció.

1.1. Objectiu del projecte.

L'objectiu del projecte és el desenvolupament d'una aplicació, per a telèfons mòbils amb sistema operatiu Android, capaç de realitzar una trucada mitjançant la descodificació d'un codi QR que conté un número de telèfon codificat en una imatge de dimensions fixes.

Aquesta aplicació serà orientada a facilitar la realització d'una trucada a una persona amb un grau de disminució psíquica com és el cas de l'Alzheimer.

La decisió de desenvolupar i orientar el projecte cap aquest sistema, a estat condicionada pel creixement que a experimentat aquest sistema operatiu des del seu llançament al mercat per part de Google, realitzant una gran competència i guanyant mercat a altres sistemes operatius com, iOS X de Apple o al Windows Mobile de Microsoft.

També s'ha vist condicionat al gran augment de la demanda de treball, que es produeix actualment en el mercat de les telecomunicacions. Convertint-se així en unes de les sortides professionals més demandades actualment.

1.2. Objectiu Social.

Aquest projecte a banda de les finalitats tècniques té un objectiu social ja que va dirigit a les persones amb Alzheimer.

L'alzheimer és una enfermetat neurodegenerativa que es manifesta pel deteriorament cognitiu, es caracteritza sobretot per la pèrdua progressiva de la memòria y altres capacitats mentals.

Aquesta demència roba la memòria al pacient i amb el pas del temps la persona oblida coses que ha de fer, cites, noms, números de telèfon, etc..

L'aplicació posa a l'abast de persones amb Alzheimer i discapacitats intel·lectuals les noves tecnologies i és una eina molt important per aquestes ja que simplifica el menú per tal de que sigui més accessible.

En definitiva l'aplicació ha de servir per aportar una millora en la qualitat de vida del pacient amb Alzheimer i la seva família i fer que puguin viure amb més autonomia i independència.

1.3. Objectiu personal.

Finalment, aquest projecte és realitzat amb la finalitat d'introduir a l'estudiant en el desenvolupament d'aplicacions mòbils, en aquest cas per a sistemes operatius Android, degut al creixement de demanda que es pot trobar actualment en el mercat de les telecomunicacions, oferint una experiència per a possibles feines a desenvolupar en un futur.

Una altre finalitat es aprendre a utilitzar els programes per a la realització de l'aplicació, així com els llenguatges de programació

1.4. Estat de l'art.

En al actualitat, els codis QR són un fenomen en expansió, al trobar-se en diferents àmbits d'aplicació i per a diferents estructures socials, tant a nivell empresarial, com a nivell social.

Per aquest motiu podem trobar moltes aplicacions mòbils, desenvolupades per a treballar amb els codis QR i empreses que comencen a instaurar aquest codis en el àmbit de treball:

- Iberia, llança les targetes de embarcació mòbil, per que els propis usuaris puguin accedir, a l'àrea de embarcació, utilitzant una maquina de autocheckin.
- Facebook, llança un aplicació que permet crear un codi QR, per que la gent des de qualsevol punt, i mitjançant una fotografia del codi pugui agregar-te com a amic seu..
- EL google earth permet associar un destí, a una imatge d'un codi QR, per disposar de la localització, en tot moment.

2. Una mica d'Història

A la actualitat, els codis de barres, estan implantats a la societat en tots els àmbits: comercials, empresarials, socials...

Per aquesta raó es realitzarà una breu introducció al món dels codis referencials, amb una breu història dels inicis d'aquests.

2.1.- Breu història dels codis referencials

Al Desembre de 1934 es troba la primera patent (patent 1985035), d'un sistema relacionat amb els codis de barres. Aquesta va ser desenvolupada per John T. Kermode, Douglass A. Young i Harry P. Sparkes.

La seva funció era la de classificar les targetes mitjançant la lectura del seu codi de barres. (Figura 1).

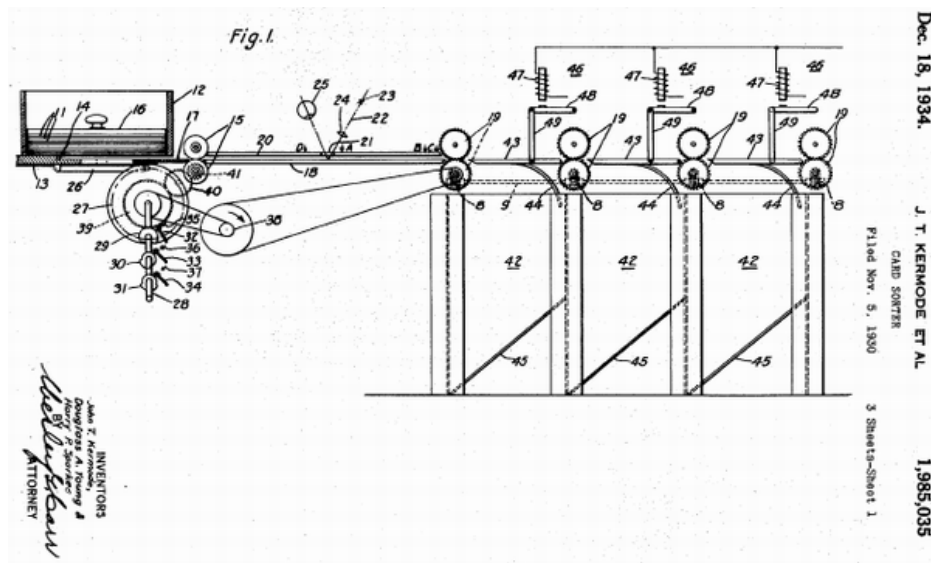


Figura 2.1. Fulla Patent 1985035

El 12 de Novembre de l'any següent, al 1935, es va patentar amb el número de patent 2020925, una "màquina organitzadora de targetes", realitzada per Douglass A. Young, i assignada a la empresa, Westinghouse Electric & Manufacturing Company, van realitzar algunes millores en el escàner òptic i en el codi, en referència a la desenvolupada per John T. Kermode, Douglas A. Young i Harry P. Sparkes. Introduint la primera simbologia multi-amplada pel lector de codi de barres òptic.

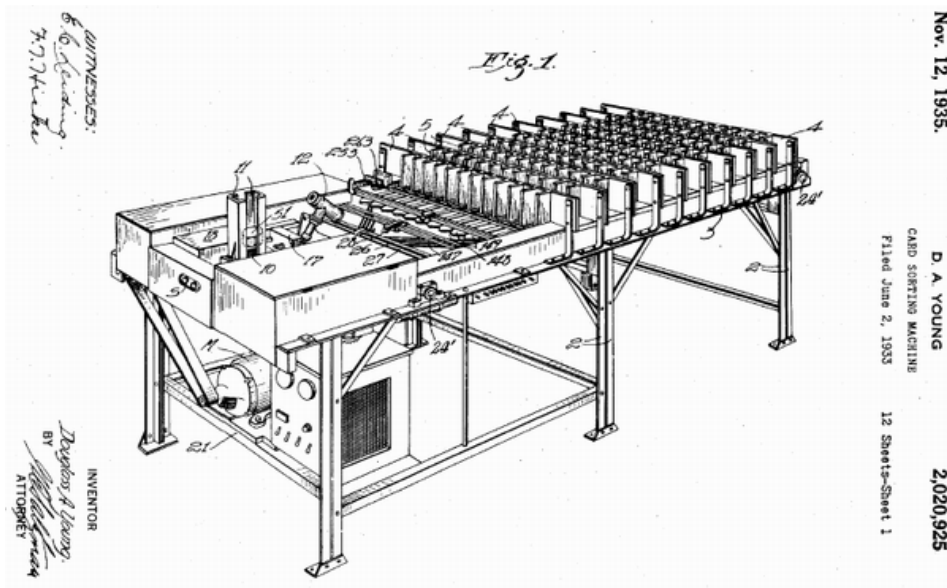


Figura 2.2. Fulla Patent 2020925

El 20 d'Octubre de 1949 es presenta el primer sistema de codificació de barres per Norman J. i Bernard Silver.

Era un tipus de codi anomenat “*código ojo de toro*” que es caracteritzava per que estava format per un conjunt de cercles concèntrics separats entre ells amb una distància relativa al seu significat. (Figura 3)

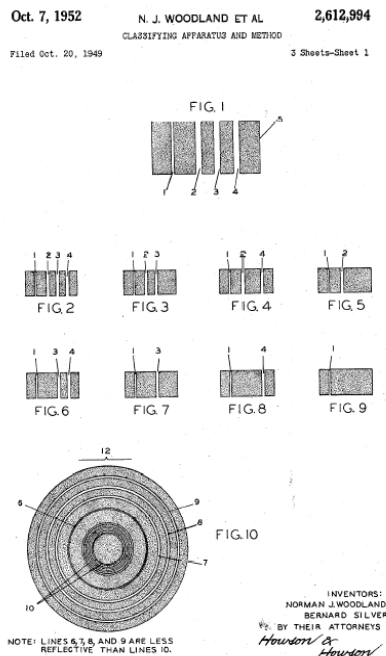


Figura 2.3. Fulla Patent 2612994

Al 1961, es desenvolupa el primer escàner fixe de codis de barres instal·lat per Sylvania General Telephone, i la seva utilitat era la d'escanejar uns codis de barres, que es caracteritzava pels seus colors (vermell, blau, blanc i negre), aconseguint identificar els diferents vagons de càrrega d'un ferrocarril.

Al 1970, el comitè AD-HOC de Supermercats de EEUU, va formar el “*Codi de Productes Comestibles Uniforme*”. Tres anys més tard van recomanar l'adopció del símbol de barres de UPC. Va ser presentat per IBM i desenvolupat per Georg Laurer, que es va inspirar al treball desenvolupat per Woodland i Silver.



Figura 2.4. Primer Codi de Barres

3. Què és un codi de barres

Un codi de barres és un codi basat en la representació d'informació o dades codificades en un conjunt de línies verticals de diferent amplada i espai, el conjunt de totes elles formen el codi de barres.

Aquest representa un mètode simple i fàcil per codificar la informació de text que pot ser llegida per dispositius òptics, els quals envien la informació a una computadora com si la informació hagués estat introduïda amb un teclat.

Però quines avantatges ens donen aquest codis? A continuació es tractaran tots els avantatges principals que ens faciliten els codi de barres.

Avantatges:

- S'imprimeix a un cost molt baix.
- Es una codificació amb una probabilitat d'error mínima.
- Captació ràpida de les dades.
- Equips de lectura e impressió, flexibles, econòmics i de fàcil instal·lació.
- Automatització de registre de productes i seguiment de vendes.

Com es processa la informació emmagatzemada?

El lector òptic de barres, descodifica la informació mitjançant la digitalització d'una font de llum que es refractar contra el codi de barres.

El procediment es realitza mitjançant la reflexió del codi de barres, aquest és il·luminat per una font de llum visible o d'infrarojos que actua de diferent manera segons si es un espai en blanc o una franja de color. En les propietats de la llum i els colors, els colors foscos absorbeixen la llum mentre que els espais en blanc la reflexen novament cap a l'escàner.

De manera que l'escàner transforma les fluctuacions de la llum en impulsos elèctrics, es a dir, un codi binari on 0 és no energia / franja negra i 1 és energia/franja blanca, envia el missatge descodificat a un terminal manual, PC o sistema centralitzat de computació.

El descodificador pot estar integrat o no dintre del lector. Actualment els escàners utilitzen díodes emissors de llum visible o infraroja (LED), làser de Neó o díodes làser en estat sòlid.

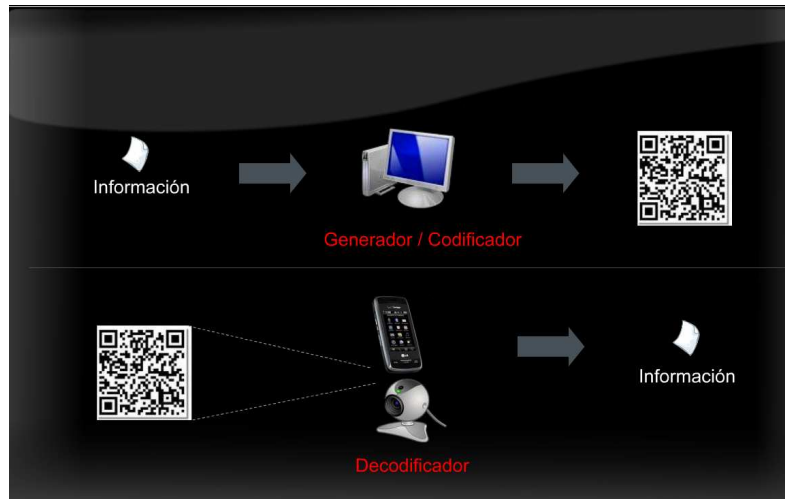


Figura 3.1. Procés de descodificació

3.1.-Tipus de codis de barres

Actualment dintre del mercat dels Codis de Barres, es poden trobar dues estructures diferents de codis: els 1D , codis lineals, i els 2D bidimensionals.

Dintre de cadascuna de les estructures trobarem diferents models que es caracteritzen pel contingut d'informació que contenen, com el número màxim de dígit, metacaracters, imatges o en els actuals codis els links de descàrrega o webs.

3.1.1.-Codis de barres lineals (1D)

UPC

Segurament és el codi més vist i més comú a la vida quotidiana ja que es tracta d'un sistema de codi utilitzat en la indústria dels supermercats des de l'any 1973.

També es utilitza en altres aplicacions com són: en botigues de conveniència, de secció i de llibres.



Figura 3.2. Codi UPC

En el codi UPC es poden diferenciar dos tipus de estructures de codi, els UPC-A i els UPC-E.

UPC-A

UPC-A és un codi de 12 dígits numèrics que estan separats en dues meitats:

Paritat senar.

Paritat igual.

Les dues meitats estan separades per un Guard Bar. Permeten així que un escàner òptic pugui llegir les dues meitats del codi, de manera independent.



Figura 3.3. Codi UPC-A

Els números en un sistema UPC s'estructuren de la següent manera, s'utilitzarà el sistema numèric del exemple de la Figura 3.3:

- El primer número és assignat a un Sistema del Número.
- Els següents 5 dígits (1 2 3 4 5) són assignats pel codi del fabricant.
- A continuació s'observa el caracter Guard Bar, per diferenciar la primera meitat de la segona.
- Els següents 5 dígits (6 7 8 9 0) són per establir el codi del producte.
- Finalment, el últim dígit és un número de verificació obtingut dels 11 dígits anteriors.

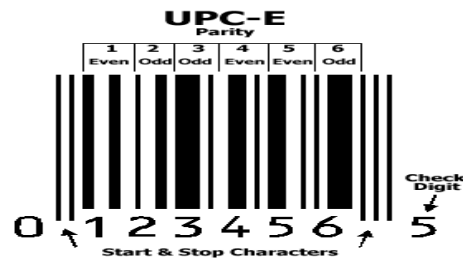
UPC-E

Figura 3.4.Codi UPC-E

Aquesta és una versió del Codi UPC, representada per 6 dígitos, només incorpora una meitat única a diferència del codi UPC-A.

Al igual que el codi UPC-A, aquest incorpora un caracter de Start i un altre de verificació.

Al ser un codi fixe de 6 dígitos, ens permet utilitzar-lo per identificar productes de dimensions reduïdes, gràcies a la seva dimensió.

EAN

European Article Number és un sistema de codis de barres adoptat per més de 100 països i gairebé un milió d'empreses a l'any 2003. A l'any 2005 la associació EAN es va fusionar amb la UCC (*Uniform Code Council*) per formar una nova i única organització mundial identificada com GS1 , amb la seva seu a Bèlgica.

EAN-13

El codi EAN-13 és el més usual, constituït per 13 dígitos i amb una estructura dividida en 4 parts:



Figura 3.5.Codi EAN-13

- La primera part es compon de dos dígits que identifiquen el país d'origen del producte.
- La segona part fa referència al codi de la empresa que es representa pels següents 5 dígits que identifiquen el propietari del producte.
- La tercera part es el codi de producte que completa la sèrie dels 12 primers dígits del codi.
- I per últim el dígit de control, que correspon a la quarta part, aquest és resol mitjançant una funció automàtica, que s'obté:
 1. Sumant els dígits de les posicions senars.
 2. El resultat obtingut és multiplicat per 3.
 3. Es sumen els dígits de les posicions parells.
 4. Es busca la desena immediatament superior.
 5. Es resta el resultat obtingut.

El resultat final és el dígit de control, si es produeix el cas de que el resultat sigui múltiple de 10 el dígit de control serà 0.

EAN-8

Com succeeix al codi UPC, en el cas del UPC-E, el codi EAN-8 és una versió del codi EAN-13, però en una versió reduïda que es compon per:

- Dos dígits de país
- Cinc dígits de dades.
- Un dígit de verificació



Figura 3.6.Codi EAN-8

Aquesta simbologia de codi, és utilitzada per aplicacions de dimensions reduïdes.

Code 39

És un codi de barres, que es caracteritza per la seva capacitat de representar majúscules, números i alguns caràcters especials, com és el espai.

Cadascun dels caràcters representats per aquest codi, és definit per un conjunt de 5 barres i 4 espais. Fent un total de 9 elements, 3 d'aquests elements són amples i 6 d'ells estrets, fent que sigui un codi d'una gran longitud.

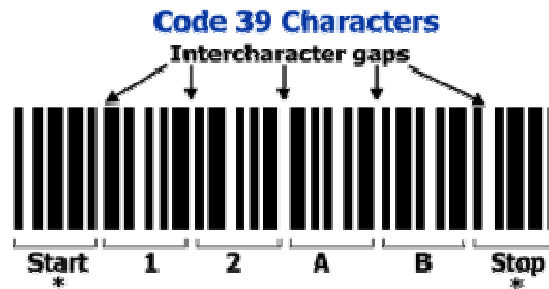


Figura 3.7.Code 39

Aquest codi és utilitzat en especificacions militars o governamentals.

El seu major desavantatge és la baixa densitat d'impressió, això significa que seria difícil etiquetar objectes massa petits, al tenir la característica de ser de dimensió variable. És un sistema de codi àmpliament utilitzat al ser interpretat per qualsevol lector de codi de barres.

A	B	C	D	E	F	G
H	I	J	K	L	M	N
O	P	Q	R	S		U
V	W	X	Y	Z		
C	1	2	3	4		*
5	6	7	8	9		
~	-	\$	%	.	/	+

Figura 3.8.Simbologia Code 39

Codi 128

És un codi de barres d'alta densitat que va sorgir l'any 1981 i és utilitzat sobretot per la indústria logística i de paqueteria. Aquest és un codi que es caracteritza perquè pot codificar caràcters alfanumèrics o només numèrics. Amb aquest codi és possible representar tots els caràcters de la taula ASCII.

Per poder entendre el seu funcionament, hem de tenir en compte que cadascun dels caràcters ASCII es codifica mitjançant 11 mòduls, de color negre o blanc.



Figura 3.9.Exemple Mòdul Codi 128

Per exemple, el caràcter ASCII <espai> està format per:

- | | |
|----------------------|----------------------|
| * Dos mòduls negres. | * Dos mòduls negres. |
| * Un mòdul blanc. | * Dos mòduls blancs |
| * Dos mòduls negres. | TOTAL= 11 MÒDULS |
| * Dos mòduls blancs. | |

El codi en realitat es divideix en 6 zones:



Figura 3.10.Code 128

- A la esquerra, una zona en blanc que té la longitud de dos caràcters.
- Un caràcter d'inici.
- Un número variable de caràcters ASCII, on es codifica la informació.
- Un dígit per fer un check de la integritat de les dades.
- Un caràcter de final o "Stop character"
- A la dreta, una zona en blanc equivalent a dos caràcters.

Code 93

Va ser dissenyat al 1982 per Intermecc, l'objectiu era aconseguir una major densitat de dades, en referència al CODE 39. És un codi alfanumèric, de longitud variable i que per cadascun dels símbols s'introdueixen dos caràcters de checksum. Els diferents caràcters, són representats amb 9 mòduls, de color negre o blanc, i sempre té 3 espais i 3 barres. Aquest codi es principalment utilitzat al codi Postal Canadenc.

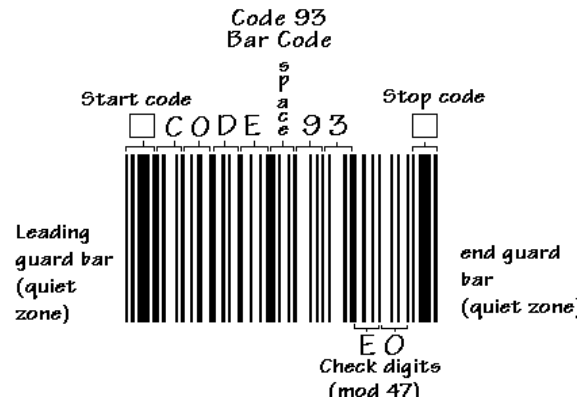


Figura 3.11.Code 93

Codabar

Desenvolupat als anys 70 per Pitney Bowes, va ser especialment dissenyat per poder ser interpretat encara que fos imprès per una impressora de matrius de punts, a més a més, permetia introduir molta més informació en una dimensió fixe d'etiqueta, gràcies a una nova simbologia de caràcters.

La seva implementació es va estendre en entorns bibliotecaris i medicinals.

Es caracteritzat per ser un codi numeral d'alta densitat.



Figura 3.12.Codabar

Codi de barres Bookland

És un codi de barres especial, utilitzat per representar els números ISBN i els preus en les tapes dels llibres. Utilitza el sistema de caracters EAN de 13 dígit, per representar el ISBN i un codi suplementari de 5 dígit, que indica el preu del llibre i la seva respectiva moneda.

Únicament ens permet codificar números, és de longitud fixe, 13 + 5 dígit.



Figura 3.12.Codi Bookland

Codi	Tipus de dades que codifica	Densitat	Cheksum	Longitut
UPC-A	0 – 9	Mitja	SI	Fixe, 12 dígit
UPC-E	0 – 9	Mitja	SI	Fixe, 8 dígit
EAN 13	0 – 9	Mitja	SI	Fixe, 13 dígit
EAN 8	0 – 9	Mitja	SI	Fixe, 8 dígit
Code 39	A – Z, 0 – 9, caracter especials.	Alta	Opcional	Variable
Code 128	A-Z, 0-9, caracters especials, codi de control ASCII	Alta	SI	Variable
Code 93	ASCII complet	Alta	Doble	Variable
Codebar	0-9-.:\$/+	Alta	NO	Variable
Bookland	0-9	Mitja	SI	Fixe, 13 + 5 dígit

Taula 3.1.Comparativa Codis Unidimensionals

3.1.2.- Codis de barres bi-dimensionals (2D)

El motiu de desenvolupament dels codi bidimensionals, és la necessitat de la societat i la industria d'introduir molta més informació en un codi de barres. Si intentem introduir uns 20 caracters, en un codi unidimensional, obtindríem un codi quilomètric que cap lector òptic podria desxifrar.

Aquesta nova tecnologia de codis de barres, ens permetran introduir o codificar milers de caràcters llegibles, en una etiqueta de longitud fixe i de dimensió reduïda. A més són uns codis molt més resistents a danys que els unidimensionals per que introdueixen fórmules de correcció d'errors (podem trobar el cas de codis que poden perdre un terç de la imatge que encara podrien ser descodificats).

Els codis més utilitzats en la actualitat són: PDF147, Datamatrix , MaxiCode i els QR. Aquests han estat molt ben adaptats en la societat ja que els drets de la propietat intel·lectual són de domini públic, eliminant el pagament per la seva utilització.

Són representats de manera matricial. Es a dir, estan fets mitjançant una estructura de cel·les establertes, que poden ser d'aparença quadrada, hexagonal o circular.

A continuació es tractarà els diferents codis 2D, desenvolupats en la actualitat:

PDF417

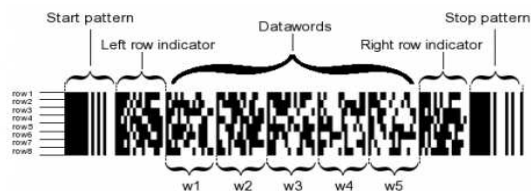


Figura 3.13.Codi PDF 417

És un codi de longitud variable, que permet codificar virtualment qualsevol lletra, número o caràcter ASCII, com a màxim 1800 caràcters alfanumèrics (ASCII) o 1100 codis binaris per cadascuna de les finestres de les que es compona el codi.

El seu nom ve del significat de PDF que significa "Portable Data File" i els números 417 es deriva de la seva estructura de 4 barres i espais per 17 mòduls.

Es considera una estructura de simbologia apilada, degut a que és composta de 3 a 90 cel·les apilades i envoltades per una zona estable en cadascun dels 4 cantons.

És un codi de longitud variable, amb una capacitat d'emmagatzematge molt gran.

I es caracteritza per seguir un patró de descodificació del codi escanejat dividit en dos passos, amb un nivell de detecció d'errors d'un 40%, realitzant una descodificació encara que el codi estigui en mal estat.

La impressió dels codis PDF147, es pot realitzar en qualsevol impressora làser o tèrmica.

S'aplica en:

- Indústria Electrónica.
- Sistemes de paqueteria, cartes.
- Validació de documents governamentals.
- Institucions financeres.
- Identificació personal i foto credencials.
- Testimonis notariais.
- Targetes de circulació.

DATAMATRIX

DataMatrix va ser desenvolupat a l'any 1989, per la companyia amb el mateix nom que el codi DataMatrix INC., encara que no va ser de domini públic fins l'any 1995.

DataMatrix és un codi de simbologia 2D de longitud variable, capaç de codificar els 128 caracters ASCII, a més d'una gran quantitat de jocs de caracters especials.

Cadascun dels símbols, consisteix en una zona perimetral quadrada, amb dos límits sòlits i dos límits puntejats. Dintre d'aquests límits es troben les cel·les on s'introdueix la informació, aquestes cel·les poden ser negre o blanques.

Els codis DataMatrix tenen una capacitat de fins a 500 MB per polzada quadrada amb una capacitat de dades de 1 a 2355 caracters.

Una dels seus grans avantatges és el seu alt grau de redundància i resistència a danys en la impressió.



Figura 3.14.Codi DATAMATRIX

S'aplica en:

- Identificació i control de components electrònics (identificació del fabricant, data de fabricació, número de lot...).
- Control i prevenció de productes en expiració.
- Codificació de direccions postals, per automatitzar la ordenació del correu postal.
- Etiquetatge de desfets perillosos, radioactius, o tòxics pel emmagatzematge.
- Indústria farmacèutica.
- Institucions financeres, transaccions segures, mitjançant la codificació de la informació.

MAXICODE

És un codi utilitzat principalment per UPS per classificar el correu a una gran velocitat degut a que es possible llegir amb gran velocitat i freqüència en qualsevol direcció.

Es considera un codi matricial 2D, de dimensió fixe (2,8 x 2,7 cm), compostat per 866 elements hexagonals repartits en 33 cel·les al voltant d'un patró central.

Un codi MaxiCode pot codificar fins a 93 caracters de dades i utilitza 5 jocs de codis diferents per poder incloure els 256 caracters ASCII.

AZTEC CODE

Es considera un codi 2D d'alta densitat, al poder codificar 3.750 caracters del joc complet de caracters ASCII de 256 bytes.

L'estructura del AZTECACODE, es caracteritza per formar-se en una quadricula amb un patró de "diana" al centre.

Aquest pot ser llegit independentment de la seva orientació i disposa d'uns mecanismes de correcció d'errors, que poden ser seleccionats pels usuaris.

RSS

Als últims anys la indústria ha demandat nous requisits pels codis comercials UPC i EAN, necessitant codificar molta més informació en un espai menor. Per aquest motiu es desenvolupa el codi RSS (“Reduced Space Symboli” o “Simbologia de Espai Reduit”).

Es poden identificar 3 versions del code RSS: RSS-14, RSS limitada i RSS Expandida, on la primera i la segona tenen un gran número de variacions optimitzades per diferents àmbits d’aplicació.

La seva aplicació en l’àmbit comercial en els pròxims anys experimentarà un creixement degut a la seva major capacitat d’emmagatzemar dades que els codis actuals (UPC i EAN).

Code QR (Quick Response)

És el codi per excel·lència actualment en la societat. Aquest va ser desenvolupat per l’empresa Denso-Wave al Japó.

És un codi 2D amb una matriu de propòsit general dissenyada per obtenir les dades amb un escaneig de la informació molt ràpid. Aquest emmagatzema les dades amb codificació de caracters Kanji (simbologia del llenguatge Japonès)

Els QR Code són fàcilment identificables degut a la seva estructura característica, aquest s’identifica per tenir uns quadrats a tres de les seves cantonades el quadrats estan definits per un interior i un exterior obscurs separats per una línia interior blanca. Aquests són utilitzats per establir el correcte posicionament del codi a l’hora de ser escanejat pel lector òptic.

Avui en dia els podem trobar en diferents àmbits d’aplicació, des de telefonia mòbil, targetes de visita, publicitat, informació...



Figura 3.15.Codi QR

BEETAGG

El codi Beetag és un codi 2D no estandarditzat, desenvolupat i optimitzat per l'empresa Mobile Tagging.

El seu nom es podria traduir com “Bee = Abella, Tagg=Etiqueta” o “Etiqueta Abella” i no es d'estranyar degut a que el seu codi, es caracteritza per la seva estructura en forma de panell d'abelles.

Una altra característica del codi Beetag és la seva capacitat d'introduir en la seva estructura, el logo de l'empresa o una imatge en concret, sense interferir en la descodificació d'aquest.

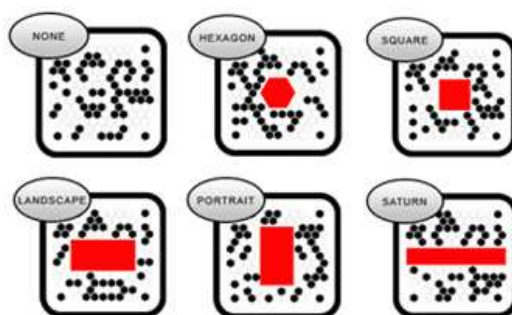


Figura 3.15. Codi BEETAGG

3.2.- Comparatives 1D, 2D

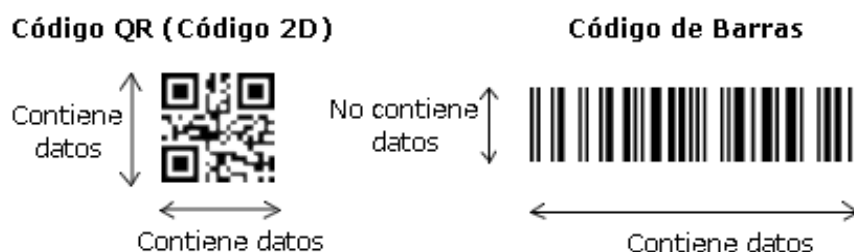


Figura 3.16. Comparativa Codis 1D i 2D.

Comparació de simbologia entre un codi 1D i 2D.

A continuació s'observarà la transformació del codi “12345678901234567890” representat per un codi 1D i dos codis 2D. Es pot observar com l'estalvi de l'espai entre el codi 1D i 2D és molt considerable.

Codi	Informació	Imatge
Code 39	12345678901234567890	 12345678901234567890
PDF 417	12345678901234567890	
DataMatrix	12345678901234567890	

Taula 3.2.Comparativa Codificació diferents Codis

4. Què són els codis QR.

Com s'ha tractat al punt anterior, “3.1 Tipus de Codis de barres”, els codis QR, són les sigles de Quick Response (*Resposta Ràpida*), és un tipus de codi 2D, compost per una matriu de punts que conté informació.

Però que se sap realment d'aquest codi?

En aquest apartat es tractarà amb profunditat la composició o estructura d'aquest codi, així com la informació que ens facilita o la seva codificació.

Un aspecte a destacar és que aquest codi està disponible per a qualsevol persona/organització, i a estat aprovat mitjançant un Standard AIM, JIS i ISO.

4.1.- Qui els desenvolupa.

La corporació Denso-Wave (www.denso-wave.com), és la responsable del desenvolupament dels Codis QR. Va ser a l'any 1994, quan es va treure al mercat públic, principalment destinat a ser una solució fàcil d'interpretar per equips òptics. Poc després va passar a trobar-se en altres aplicacions, més quotidianes, com la publicitat, la informació, la telefonia mòbil...

Inicialment aquesta tecnologia de Codi va ser molt acceptada al Japó on va ser dels codis més utilitzats per la societat, encara que a Europa i els EEUU també està rebent una gran acceptació.

Aquesta acceptació es pot observar a través de l'estudi realitzat per la companyia, “3G Visión”, creadora del lector-òptic per telefonia mòbil “i-nigma”, que actualment es troba en el mercat dels Sistemes Operatius Android i iPhone.

Top 10 users of mobile barcodes during Q1/2011

- (1)1.United States
- (2)2.Italy
- (3)3.Germany
- (7)4.United Kingdom

(8)5.Netherlands

(6)6.Canada

(5)7.France

(4)8.Hong Kong

(17)9.Spain

(15)10.Switzerland

() – Posició al quart trimestre de l'any 2010

Top 5 growing countries for Q1/2011

1.United States(181.1%Growth over Q4/2011).

2.United Kingdom(166.5%).

3.Netherlands(146.3%)

4.Spain(94.4%)

5. Canada (94.0%)

Un cop més Espanya es troba entre el Top 10 dels països amb més número de captures de QR's i en el Top5 de major creixement respecte al quadrimestre anterior un (+94%).

També profitarem l'estudi realitzat per l'empresa 3G Vision per observar el creixement que han obtingut els Codis Qr, en la societat des de l'any 2009. Es podrà observar un creixement important en el transcurs dels dos últims anys.

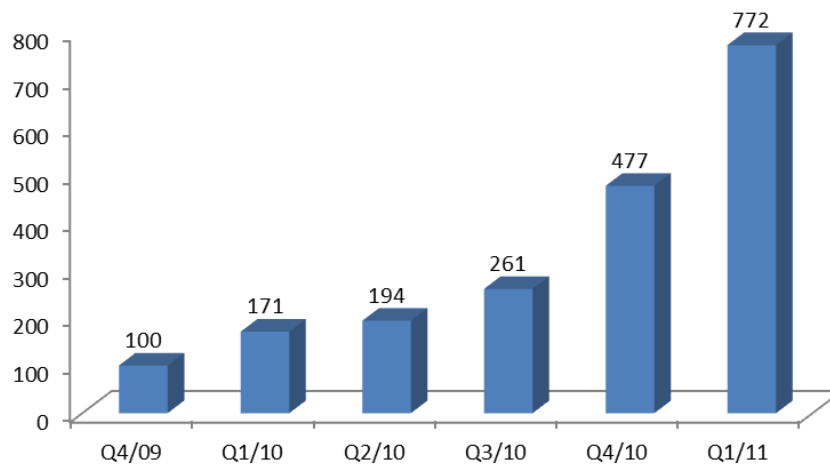


Figura 4.1. Evolució utilització codis QR.

4.2.- Estructura codi QR.

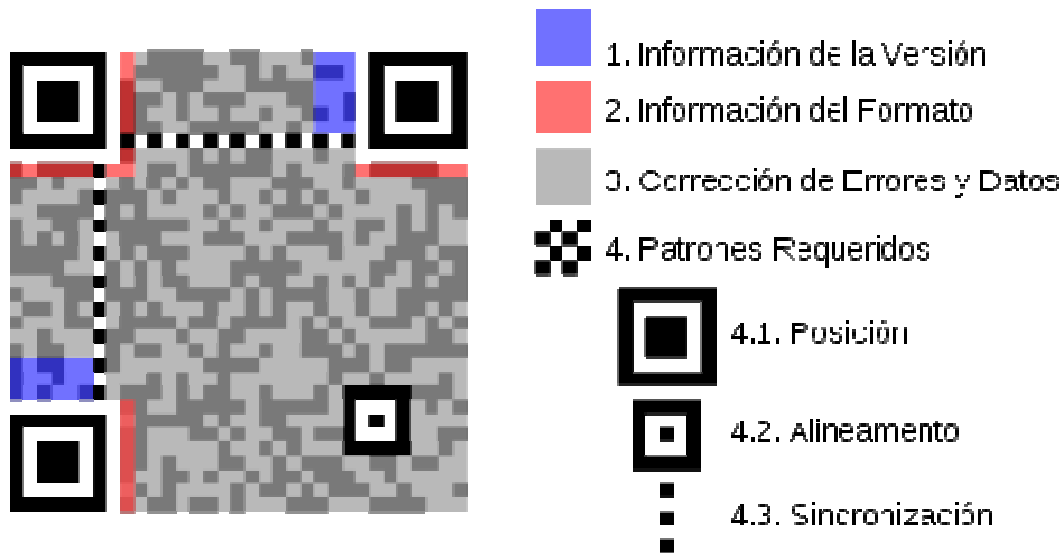


Figura 4.2.Estructura Codi QR.

Els quadrats de les cantonades, com hem comentat en apartats anteriors, són els punts de referència general per establir la correcta posició del codi, pel lector.

El quadrat més petit (alineació), indica a l'escàner la posició correcta de tota la informació.

Observarem que els quadrats de posició estan enllaçats per puntets negres-blancs denominats de sincronització que ajuden encara més a una correcta ordenació de les dades i la correcta disposició de tots els elements.

Altres elements com són els de color vermell i blau, indiquen al lector les característiques del codi com són el seu nivell de correcció d'errors o la versió d'aquest.

4.3.- Informació emmagatzemada.

Els codis QR, ens permeten emmagatzemar una gran quantitat de dades, en una imatge de dimensions fixes.

Capacitat d'emmagatzematge de dades:

Numèric	màxim de 7.089 caracters
Alfanumèric	màxim de 4.296 caracters
Binari	màxim de 2953 bytes
Kanji/Kana	màxim de 1.817 caracters

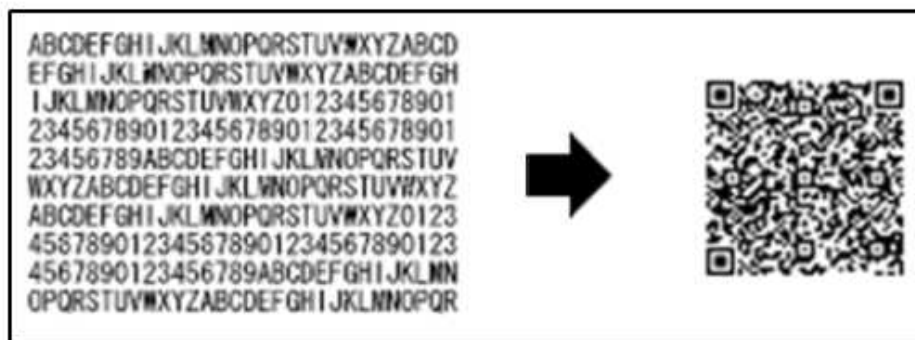


Figura 4.3. Exemple de Codificació de text.

4.4.- Correcció d'errors

Un altre avantatge dels codis QR, és la seva capacitat de correcció d'errors, facilitant la seva descodificació encara que aquest hagi estat malmès, per motius d'inclemències del temps, pols, pintades...

Podem trobar fins a 4 nivells de correcció d'errors, augmentant la seva capacitat correctora del codi però també augmentant la seva mida.

Capacitat de correcció d'errors:

Nivell L	7% de l'estructura es pot restaurar
Nivell M	15% de l'estructura es pot restaurar
Nivell Q	25% de l'estructura es pot restaurar
Nivell H	30% de l'estructura es pot restaurar

L'algoritme utilitzat es coneix com el corrector d'errors Reed-Solomon. Es caracteritza per ser un codi cíclic no binari, catalogat entre els codis BCH (*Bose Chaudhuri Hocquenqhem*), que tenen la capacitat de detectar i corregir errors en un canal de comunicació, utilitzant uns bits addicionals que permeten la recuperació posterior.

Depenent sempre de la quantitat de dades que s'han de corregir.

4.5.- Com crear i llegir un codi QR

Com hem comentat en apartats anteriors, els codis QR són un sistema de codi que és obert per a qualsevol persona o organització, utilitzant-ho per emmagatzemar informació, però: com aconseguim emmagatzemar aquesta informació en un codi QR?

En l'actualitat podem trobar molts sistemes, via web, telèfon, software, per codificar les dades desitjades en un codi QR, per aquest motiu es tractaran algunes d'elles:

Kaywa (Web): Es tracta d'un generador amigable, amb un formulari que ens permet configurar quasi tots els aspectes, versió de codi, nivell de correcció d'errors, dimensió del codi desitjat...Ens permet codificar diferents tipus d'informació, com una URL, text, número de telèfon o un SMS. La limitació que ens ofereix el codificador, és la capacitat de codificar fins a 250 caràcters.

Llibreria QRCode per Java: És una llibreria que ens permet incloure un suport per codificar o descodificar codis QR. Inclou el suport per la correcció d'errors, així com unes aplicacions per tenir una sortida molt més ràpida. (*Open Source*).

PyQRcode (soft): mòdul per de/codificar imatges amb codis QR en Python. Esta disponible per Windows i Linux (*Open Source*).

QRdraw Pro (soft): Creació de codis QR únicament, desenvolupada per la companyia creadora dels codis QR, Denso-Wave (*Pagament*).

QRmaker Pro: al igual que l'anterior, és una aplicació que esta creada per Denso Wave, que ens permet crear i controlar aquest tipus de codis mitjançant un control de ActiveX. (*pagament*).

Invx (Web): Es tracta d'un altre web que ens permet crear directament un codi des de qualsevol navegador, amb opcions molt més limitades.

Un cop codificada la informació en un codi QR, arriba l'hora de seleccionar l'eina de descodificació que s'utilitzarà per descodificar la informació i recuperar-la. Com succeeix amb els codificadors, podem trobar diferents aplicacions per fer-ho (web, mòbil o PC), disponibles gratuïtament o de pagament.

Algunes d'elles són les següents:

Kaywa Reader (soft): Esta pensat per a telefonia mòbil, i necessita una càmera per poder utilitzar-lo.

i-Nigma (soft): És un altre aplicació per a telefonia mòbil desenvolupada per la companyia 3G Vision.

Intelcom QR Code Reader (Web): Es tracta d'una llibreria Java que pot ser utilitzada via web, o descarregada i executada de forma local.

ZXing: És una aplicació de Google, desenvolupada per a Sistema Operatiu Android.

ZXing Decoder Online (Web): aquesta és la mateixa aplicació desenvolupada per Google per aplicacions mòbils, però en aquest cas per ser utilitzada via web, amb l'únic requisit que hem de posar la imatge a descodificar-la en un servidor web.

4.6.- Exemples d'utilització al món real

Transport públic.

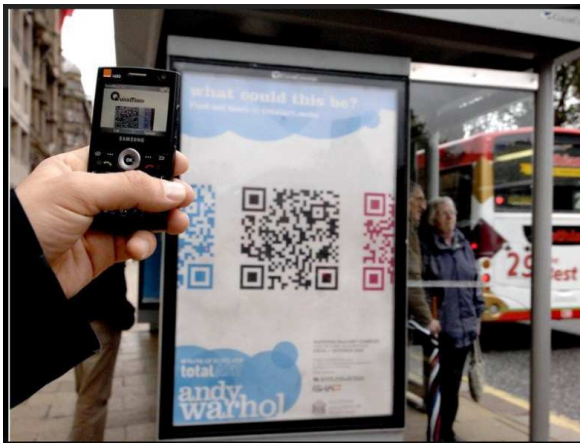


Figura 4.4. Publicitat BUS



Figura 4.5. Utilitat Google Maps.

Figura 4.4. Utilització per promocionar una marca de roba, en una parada de bus.

Figura 4.5. Imatge en el Google Earth, per disposar en tot moment de la ubicació.

Revistes i diaris



Figura 4.6. Anunci La Vanguardia



Figura 4.7. Portada The Sun.

Figura 4.6. Anunci de oferta de treball, oferida a La Vanguardia, obtenint la informació amb un codi QR.

Figura 4.7. Portada de la Revista The Sun, preguntant que te la noia a les mans, anunciant un article de la revista.

Alimentació



Figura 4.8. Denominació Ampolla de vi



Figura 4.9. Codi promocional Pepsi

Figura 4.8. QR utilitzat per identificar la denominació de la ampolla de vi que s'observa a l'imatge.

4.9. Codi promocional de la marca Pepsi, per una promoció oferida per la pròpia marca.

Indústria



Figura 4.10. Utilització paqueteria.



Figura 4.11. Especificacions Electrónica.

Figura 4.10. Codi utilitzat per la identificació de paqueteria en un entorn de producció real.

Figura 4.11. Codi per identificar, i obtenir les especificacions tècniques d'un component electrònic.

5. Evolució de la telefonia mòbil i dels Sistemes Operatius.

5.1.- Breu història de la telefonia.

Martin Cooper va ser el pioner en la tecnologia mòbil, es considera com el “pare de la telefonia cel·lular” al introduir el primer radiotelèfon al 1973 als EEUU, quan estava treballant per Motorola. Però no va ser fins el 1979 quan va aparèixer el primer sistema comercial sense fils, va ser a Tokio per la companyia NTT (*Nippon Telegraph & Telephone Corporation*).

En 1981 als països Nòrdics es va introduir un sistema cel·lular similar a AMPS (*Advance Mobil Phone System*). Per una altre banda, als EEUU gràcies a que l'entitat reguladora d'aquest país va prendre mesures per la creació d'un servei comercial de telefonia cel·lular, al octubre de 1983 es va posar en funcionament el primer sistema comercial a la ciutat de Chicago.

A partir d'aquell moment, la resta de països van veure en la telefonia cel·lular, l'alternativa a la telefonia convencional amb fils. Al tenir una gran acceptació en la societat, es van trobar amb la necessitat de desenvolupar i implementar altres maneres d'accés multicanal i transformar el sistemes analògics en digitals, per donar-li servei a molts més usuaris.

5.2.- Evolució de la tecnologia mòbil

Per separar una etapa d'una altre, la telefonia cel·lular s'ha caracteritzat pel desenvolupament de les generacions mòbils. Per aquest motiu es descriurà cadascuna d'elles en el següent apartat.

La primera generació 1G

La 1G de la telefonia mòbil va fer la seva aparició al 1979, i es va caracteritzar per ser analògica i estrictament per veu. La qualitat dels enllaços era molt dolenta, degut a una

baixa velocitat [2.400 bauds], la transferència de cel·les era imprecisa, una qualitat basada en FDMA (*Frequency División Multiple Access*) i la seguretat no existia.

La tecnologia que va predominar aquella època, s'anomenava AMPS (*Advances Mobile Phone System*).

La segona generació 2G

Als anys 90 va aparèixer la segona generació que a diferència de la primera, aquesta es va caracteritzar pel canvi analògic-digital.

Utilitzava uns protocols de codificació molt més sofisticats, que són els utilitzats actualment al mercat.

Les tecnologies predominants són: GSM (*Global System for Mobile Communications*); IS-136 (conegut també com TIA/EIA-136 o ANSI-136) , CDMA (*Code División Multiple Access*) i PDC (*Personal Digital Communications*), aquest últim utilitzat al Japó.

Aquests protocols suportaven unes velocitats d'informació molt més altes per la veu, però limitades en comunicacions de dades. Encara que es van començar a introduir alguns serveis com fax, dades i SMS (*Short Message Service*).

Aquesta generació als EEUU, és coneguda com PCS (*Personal Communications Services*).

La generació 2.5G

Va ser una generació intermèdia entre la 2G i la 3G, degut a que des del punt de vista dels proveïdors de serveis de telecomunicacions, aquesta petita evolució, va facilitar econòmicament arribar al desenvolupament de la tecnologia 3G. Doncs era una tecnologia molt més ràpida i més econòmica.

Va oferir característiques i serveis addicionals als sistemes 2G, tals com GPRS (*General Packet Radio System*), HSCSD (*High Speed Circuit Switched Data*), EDGE (*Enhanced Data Rates for Global Evolution*), IS-136B, IS-95B, entre d'altres.

L'únic que no realitzarà aquesta petita migració serà el país del Japó, que va evolucionar directament de la generació 2G a la 3G.

Tecnologia	Generació	Tipus de transmissió	Velocitat màxima
TDMA IS-136	2G	Commutació de circuits	9,6 Kbps
GSM	2G	Commutació de circuits	9.6 Kbps
PDC	2G	Commutació de circuits	9.6 Kbps
CDMA IS-95 A	2G	Commutació de circuits	14.4 Kbps
CDMA IS-95B	2G	Commutació de paquets	64 Kbps
GPRS	2.5G	Commutació de paquets	115Kbps
EDGE	2.5G	Commutació de paquets	384 Kbps
CDMA 2000 1X	2.5G	Commutació de paquets	144 Kbps
CDMA 2000	3G	Commutació de paquets	2.0 Mbps
WCDMA	3G	Commutació de paquets	2.0 Mbps

Taula 5.1. Evolució Tecnològica telefonia mòbil

La tercera generació 3G

És el punt més àlgid de la telefonia, degut a que es produeix la convergència entre la veu i les dades, introduint accés sense fils a Internet, aplicacions multimèdia i altes velocitats de transmissió de dades. Els protocols emprats als sistemes 3G suporten altes velocitats d'informació enfocades per aplicacions molt més enllà de la veu, com són el àudio (MP3), vídeo en moviment, vídeo conferències i accés ràpid a Internet.

Aquesta generació es va començar a implementar al 2001 al Japó per NTT DoCoMo, i a Europa i part d'Àsia al 2002. Posteriorment als EEUU i altres països.

Els sistemes 3G arriben a velocitats de 384 Kbps fins els 2Mbps, depenent del lloc, (interior o exterior), velocitat del cos, condicions meteorològiques... Entre les tecnologies d'aquesta tercera generació es troben la UMTS (*Universal Mobile telephone Service*), CDMA2000, IMT-2000, 3GPP, UWC-136 entre d'altres.

La quarta generació 4G

Està basada completament en el protocol IP, aconseguint uns sistemes homogenis que poden arribar a la convergència entre les xarxes amb fils i sense fils. Aquesta tecnologia utilitza mòdems sense fils, terminals intel·ligents i altres dispositius mòbils. Arribarà a garantir velocitats de 100 Mbps en moviment i fins a 1Gbps en estat de repòs, oferint

una qualitat de servei increïble, a més a més d'una seguretat que oferirà serveis de qualsevol classe.

L'objectiu de la WWRF (*Wireless World Research Forum*), és obtenir la fusió de les diferents tecnologies i protocols, similar a 3G, que actualment treballen amb les tecnologies com GSM i CDMA.

L'empresa NTT DoCoMo al Japó, va ser la primera en realitzar experiments amb la tecnologia de quarta generació, arribant a oferir 100 Mbps a un vehicle a una velocitat de 200 Km/h i oferint a l'any 2010 cel·lulars amb tecnologia 4G basats en LTE (*Long Term Evolution*), treballant amb tècniques avançades de radio com MIMO i OFDM.

Aquesta tecnologia de generació s'espera que estigui implantada a la resta de països al any 2020.

5.3.- Sistemes Operatius Mòbils.



Figura 5.1. Telèfons amb els seus Sistemes Operatius

Un sistema operatiu és una capa complexa entre el hardware i el usuari, que facilita a l'usuari i als programadors les eines i interfícies adequades per realitzar les tasques informàtiques, de manera ràpida i assequible, evitant els nivells de processament complexos per realitzar-les.

Es pot deduir que l'ús d'un o un altre sistema operatiu determinarà les capacitats multimèdia dels dispositius, i la forma en que aquests sistemes interactuaran amb l'usuari final.

En l'actualitat, existeixen multitud d'opcions dintre del mercat de la telefonia mòbil, com són Symbian, BlackBerry OS, Windows Mobile, iPhone OS i Android (Google) o petits sistemes operatius en entorns Linux.

Per aquest motiu es farà una breu descripció de tots ells, tractant les seves característiques més rellevants en diferència amb els altres sistemes operatius.

Symbian

Aquest és el sistema operatiu més estès entre "SmartPhones", i per aquest motiu és el sistema operatiu amb més aplicacions desenvolupades pel seu entorn.

Actualment el sistema operatiu Symbian, es troba en el 65% dels mercats de la telefonia mòbil.

La seva principal virtut és la capacitat que té el sistema d'adaptar-se i d'integrar qualsevol tipus d'aplicació: oferint les rutines, els protocols de comunicació, un correcte control d'arxius; i tot això per garantir el correcte funcionament de l'aplicació.

La tecnologia del sistema operatiu Symbian, ha estat dissenyada tenint en compte punts claus com són els de proporcionar energia, memòria i gestió dels inputs/outputs dels recursos específics d'un telèfon mòbil.

Una altre de les seves virtuts és que es considerada una plataforma oberta degut a que l'empresa AUNA Telecomunicaciones, a realitzat una estandardització global en Internet del seu sistema operatiu. Destacant una compatibilitat amb els estàndards de connectivitat i xarxa, com Bluetooth, WIFI, GSM, GPRS, CDMA i WCDMA.

Windows Mobile

Com no, Microsoft, va treure al mercat el seu propi Windows per a dispositius mòbils, anteriorment conegut amb el nom de Windows CE o Pocket PC, desenvolupat per a dispositius PDA i ordinadors de butxaca, amb una llarga història a l'ombra del líder Palm OS. En la actualitat ha superat a Palm OS.

Windows va desenvolupar Windows Mobile, un sistema operatiu escrit des de 0 i que utilitza algunes de les convencionals interfícies d'usuaris del Windows de sempre.

Una de les avantatges de Windows Mobile sobre els seus competidors, es que els programadors poden desenvolupar aplicacions mòbils utilitzant, els llenguatges i

entorns que ofereix Windows per PC. Altres sistemes operatius en comparació amb Windows Mobile, necessiten molt més esforç de desenvolupament, encara que per una altra banda ofereixen una millor optimització per a cadascun dels diferents telèfons.

Android

Google, el gegant de la informàtica en el món actual, que és capaç d'agafar una pedra i convertir-la en or, va desenvolupar el seu propi sistema operatiu anomenat "Android".

Android és un Sistema Operatiu mòbil basat en Linux i Java que ha estat alliberat sota llicència Apache v.2.

El sistema busca, novament, un model d'estàndarització de programació que simplifiqui les labors de creació d'aplicacions mòbils i normalitzi les eines en el camp de la telefonia mòbil.

Al igual que com Symbian, el que es busca és que els programadors només tinguin que desenvolupar la seva aplicació, sense tenir en compte per a quin dispositiu mòbil va orientat ja que el Sistema Operatiu s'encarrega de donar suport pels diferents terminals mòbils.

A l'actualitat és el Sistema Operatiu que està experimentant un creixement exponencial més gran als dispositius mòbils.

Aquest sistema serà tractat amb molta més profunditat en pròxims apartats.

iPhone OS

iPhone OS, de la companyia Apple, és una versió reduïda de Mac OS X per PC, optimitzada pels processadors ARM.

Per poder desenvolupar les aplicacions la via oficial, forma part del iPhone Developer Program de pagament i que utilitza el SDK de iPhone que és gratuït.

Una de les seves virtuts, és que iPhone disposa d'una interfície d'usuari realment interessant, i un únic desavantatge és la quantitat de restriccions que té.

El futur d'aquest sistema i el seu temps de vida és molt qüestionable, al ser un sistema propietari de la seva companyia, Apple, només funciona amb dispositius desenvolupats per ella, es diu que té un temps de vida curt, a no ser que alliberin el seu sistema.

En la actualitat es pot trobar en un 7% del mercat mòbil.

Blackberry OS

BlackBerry és un sistema operatiu multi-tasca que està arrasant en l'escena empresarial, en especial pels seus serveis de correu electrònic, chat (BlackBerry Chat) i teclat QWERTY.

Actualment BlackBerry OS disposa d'un 11% del mercat mòbil.

El Sistema Operatiu BlackBerry surt al mercat al mateix temps que en el món de la telefonia mòbil es comença a demandar un sistema operatiu que ofereixi la utilització de forma ràpida i fàcil dels serveis de correu electrònic.

Avui en dia, també es proveïdor de serveis de correu electrònic a dispositius que no són BlackBerry, gràcies al programa BlackBerry Connect.

En línies generals el que ens permet un dispositiu BlackBerry i pel que es caracteritza és per poder redactar, enviar i rebre tot tipus de missatges de correu electrònic, al igual que en el programa que utilitzem al PC. A més a més, és possible realitzar i contestar a les trucades que són transmises a través de la xarxa (VoIP).

Una altre de les evolucions lògiques, és la capacitat dels dispositius d'oferir una navegació web en pàgines HTML o WAP.

És un sistema que es completa amb una quantitat d'aplicacions i programes d'organització empresarial, com són calendaris, llibreta de direccions, bloc de notes, llistat de tasques...

Palm OS

Durant la història ha estat el líder del mercat des de la seva aparició al 1996, va començar a perdre protagonisme amb l'arribada del nou segle, fins que al any 2003 es va veure amb la necessitat de dividir-se i formar part de una nova empresa anomenada PalmSource, que al any 2005 va ser adquirida per la companyia japonesa Access.

Els motius d'aquesta inestabilitat van ser l'arribada dels nous Sistemes Operatius, com el Windows Mobile, telèfons mòbils intel·ligents, i l'aparició de les plataformes i dispositius BlackBerry que van començar a deixar en segon plànol a les PDA que no incorporaven telefonia mòbil.

Aquesta plataforma a evolucionat, incorporant tot tipus de programes i funcionalitats, així com telèfon, correu electrònic, missatgeria, vídeo i imatge, compatibilitat amb arxius Office, etc.

En la actualitat es desenvolupen dispositius Palm Treo que incorporen el Sistema Operatiu Windows Mobile.

Un altre de les evolucions que garanteixen un futur prometedor és que actualment el sistema operatiu Palm OS està adquirint la operativitat de Linux.

5.4.- Comparativa Sistema Operatiu (IO's 4 Apple, Android, Windows 7, Symbian)

Per començar amb la comparativa, el primer que s'ha de saber es que el Kernel d'un sistema operatiu, és el nucli d'ell mateix, per tant és el que facilita als programes l'accés, i és l'encarregat de gestionar els recursos. Per això és important conèixer quin Kernel utilitzen els diferents Sistemes Operatius de la telefonia mòbil.

	Android	BlackBerry OS	iPhone OS	S60 5th Edition	Palm WebOS	Windows Mobile 6.5
Kernel	Linux amb màquina virtual Dalvik	Propietari	OS X	Symbian	Linux	Windows CE
Connectivitat	3G, WIFI, GSM i GPRS	3G, WIFI, GSM i CDMA	3G, WIFI, GSM	3G, WIFI, GSM	3G, WIFI, GSM i CDMA	3G, WIFI, GSM i CDMA

Taula 5.2. Comparativa Tècniques Sistemes Operatius

La principal diferència entre un kernel de lliure distribució i un propietari es troba en que els de lliure distribució com Linux, gaudeixen d'una ampla i experimentada comunitat de desenvolupadors, que gràcies a ells es detecten ràpidament problemes de seguretat, errors, etc. I es realitzen millores per solucionar els problemes i actualitzar els sistemes.

Cosa que en els sistemes propietaris o tancats, és molt més difícil trobar els errors i millorar els sistemes, degut a que han de ser els propis desenvolupadors del sistema els que detectin i realitzin les millores, incrementant el cost econòmic del sistema.

Altres comparatives poden ser l'adaptabilitat dels sistemes, que es tracta de la capacitat o facilitat de poder adaptar el Sistema Operatiu a diferents terminals mòbils. En aquest sentit Android és el que millor adaptabilitat presenta, degut a que cada vegada s'estan utilitzant en més dispositius, no només en telèfons mòbils, sinó també en Netbooks. En canvi la resta de sistemes operatius tenen una adaptabilitat menor i més complexa.

Per últim la connectivitat dels sistemes operatius avui en dia és primordial per poder treure el màxim partit a totes les funcionalitats que ofereixen cadascun d'ells. Una d'elles i la més important és la connectivitat a Internet.

Avui en dia aquesta comparativa seria absurda realitzar-la degut a que tots els sistemes operatius actuals garanteixen una connectivitat a Internet, encara que no tots garanteixen les últimes tecnologies de captació de la informació.

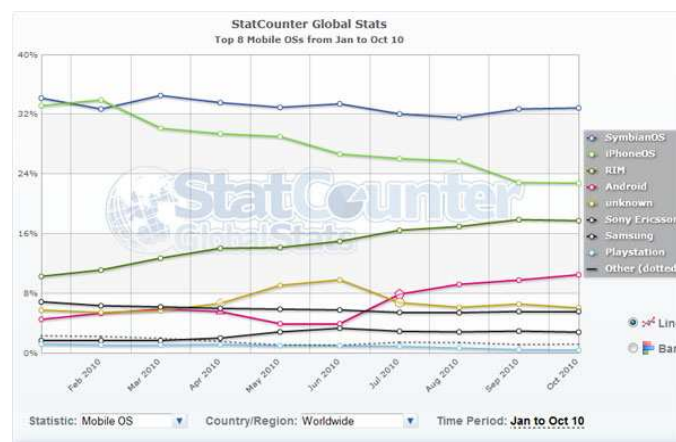


Figura 5.3. Gràfic de Ventas dels diferents Sistemes Operatius al Món

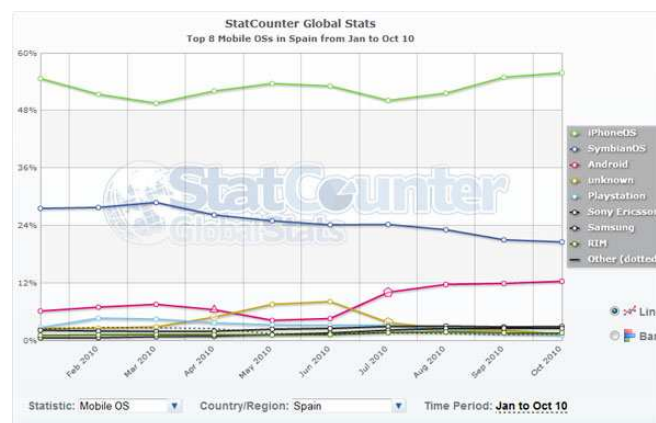


Figura 5.4. Gràfic de Ventas dels diferents Sistemes Operatius a Espanya

5.5.- Evolució del Sistema Operatiu Android.

Des de que va aparèixer al mercat el primer Sistema Operatiu de Google, al any 2008 al terminal T-Mobile HTC G1, anomenat Android, aquest a evolucionat a un ritme frenètic, deixant a un costat els grans sistemes operatius de la telefonia mòbil com Windows Mobil i Symbian per lluitar de tú a tú amb el gran líder del mercat actual, Apple amb el seu Sistema Operatiu IO's X.

Però com ha sigut aquesta evolució frenètica d'aquest sistema operatiu? A continuació es tractarà aquesta evolució veient les diferents versions que han aparegut, des de la 1.5 Cupcake fins a l'actual 3.0 Gingerbread, que no té res a envejar al sistema operatiu d'Apple.

Android 1.5 "Cupcake"

La primera versió d'Android, que ja era una evolució de les primeres versions betes que corrien per la xarxa, es va presentar amb uns acabats molt polits, i amb unes característiques que el diferenciarien de la resta com eren, els widgets en pantalla principal i la botiga online Android Market. Cosa que no era una novetat en el mercat, el tema dels Widgets, però oferien una millor optimització que la resta dels seus competidors. Mentre que la botiga on-line Android market podia mirar de tu a tu, a la gran AppleStore, de la companyia de Apple.

En aquesta versió ja es va poder observar una altre de les virtuts que caracteritzaria aquest Sistema Operatiu, i era que disposava de la flexibilitat suficient per que qualsevol fabricant de telefonia mòbil pugues introduir la seva pròpia interfície gràfica.



Figura 5.5. Interfície Gràfica Android Cupcake

Android 1.6: Donut

Va ser una actualització que va enriquir de noves funcionalitats al sistema. Introduint el buscador, realitzant les cerques per Internet o dintre del propi dispositiu, com podien ser contactes, aplicacions, etc. Una altre aspecte que va millorar va ser la funcionalitat de la càmera, introduint un botó per poder canviar a gravació de vídeo o càmera fotogràfica, i com no una nova millora en la seva aplicació Android Market.

Aquesta va ser una actualització, de curt termini per que es va veure eclipsada per les versions (2.0/2.1), que van sortir pocs mesos després.

Android 2.0: Eclair / Android 2.1:Eclair Updated

Amb l'arribada de la versió 2.0, al 2009, va ser el dia en que Android va començar a refermar la seva imatge al mercat de la telefonia mòbil.

Amb aquesta actualització una de les millores més importants, que va fer que Android millores el seu aspecte social, va ser la capacitat de sincronitzar tots els contactes de qualsevol xarxa social: twitter, facebook, twenti, etc a un sol dispositiu, permetent agafar les fotografies dels contactes del seus perfils i enllaçar-les amb els contactes de l'agenda telefònica.

En l'aspecte estètic de l'interfície es van millorar les animacions, les transicions, etc.

Però va ser amb l'aparició de dues aplicacions on es va observar les grans novetats que oferia aquesta nova versió.

La primera va ser l'explorador d'Internet, que introduïa el zoom amb un doble click, i la segona aplicació va ser la del Google maps, oferint navegació gratuïta.

La versió 2.1, va sorgir amb la sortida del Nexus One, ampliant un parell de funcionalitats noves al sistema, com la possibilitat de disposar de 5 escriptoris, la realització de cerca per veu i l'aplicació de Google que permetia realitzar la cerca d'un objecte enfocat per la càmera.

Android 2.2:Froyo

Després de moltes actualitzacions per les versions 2.0 i 2.1, va aparèixer la versió Android 2.2: Froyo, una versió que millorava notablement la velocitat de tot el sistema, tant a nivell d'aplicació com a l'hora de navegar per Internet.

Permetia la connexió 3G amb altres dispositius via WIFI, anomenada “tethering”. I amb l'esperada compatibilitat amb Adobe Flash.

Android 3.0: Gingerbread

I arribem a la versió Android 3.0, la versió que permet treballar sobre dispositius amb doble nucli, memòries de 1Gb i càmeres de 12Mpx.

Aquesta versió, introdueix en el mercat la versió de Google Music, per poder fer front a l'aplicació iTunes del sistema d'Apple i la capacitat de sincronitzar el terminal amb qualsevol PC, mitjançant l'aplicació Google Sync.

Aquest sistema encara no esta disponible per segons quins terminals mòbils, mentre que en sistemes Tablet Pc i PC's amb Sistema Operatiu implantat es pot trobar en el mercat.

6. Eines utilitzades per programa Android

6.1.- Llenguatges de programació

Dintre dels llenguatges de programació es poden trobar de molts tipus diferents al llarg de l'història, des de el llenguatge COBOL, C/C++, VisualBasic, JAVA...

Com tota l'història de l'informàtica, ha anat evolucionant, segons la demanda del mercat de l'època, inicialment eren les targetes perforades, per a grans màquines de treball, i eines de càlcul, després va arribar el COBOL, el C/, d'un nivell de programació màquina i fins a l'actualitat on trobem el codi JAVA, que és el codi de programació per excel·lència, amb totes les seves evolucions diferents J2EE i J2ME, ja que esta orientat a màquines amb interfícies gràfiques "orientat a objectes", amb eines web, eines de I/O...

6.1.1.- JAVA

Java és una tecnologia orientada al desenvolupament de software amb el que podem realitzar qualsevol tipus de programa. Avui en dia, la tecnologia JAVA té molta importància en l'àmbit d'Internet gràcies a la seva plataforma J2EE i la plataforma J2ME per a dispositius Mòbils.

Una de les raons per les que el codi JAVA va experimentar un gran creixement, va ser la seva capacitat de que el codi pugui funcionar en qualsevol plataforma de software i hardware. Això significa que pot funcionar tant en un Linux o Windows, sense cap problema.

Quines són les característiques del JAVA, que el fan tant característic:

1.- Llenguatge Simple: és un llenguatge amb un aprenentatge ràpid, a comparació amb altres llenguatges com el C, C++.

2.- Orientat a Objectes: desenvolupada per objectes, agrupa l'estructura dels objectes i els seus mètodes "funcions" que manipulen les seves dades, encapsulats en el seu codi.

3.- Distribuït: proporciona milers de col·leccions de classes per al seu ús en aplicacions de xarxa, que ens permeten obrir sockets, establir connexions amb servidors clients i remots, facilitant així la creació de les aplicacions.

4.- Robust: va ser desenvolupat per crear software altament fiable. Per aquest motiu proporciona nombroses comprovacions en compilació i en temps d'execució.

5.- Indiferent a l'arquitectura: Java desenvolupa aplicacions que han de suportar els entorns de xarxa més variats, des de Unix fins a Windows NT, passant per MAC. Per aquest motiu el compilador Java genera bytcodes, que es tracta d'un format intermedi indiferent a l'arquitectura dissenyat per transportar el codi a múltiples plataformes hardware i software.

6.- Multitasca: avui en dia es veuen molt limitades les aplicacions que només poden executar una acció alhora, per aquest motiu Java és capaç de treballar amb múltiples fils en execució (multithreading).

7.- Productes APPLETS: petits programes que apareixen a les Webs, amb la capacitat d'executar accions molt complexes, com animar imatges, establir connexions de xarxa, activar menús i quadres de diàlegs per després realitzar accions...

8.- APP Mobils.: és l'evolució final del codi Java, Java Micro Edition, una especificació de la plataforma Java orientada a proveir una col·lecció certificada de API's de desenvolupament de software per a dispositius com PDAs, mòbils o electrodomèstics. Ens permet un entorn de simulació al PC, durant la fase de desenvolupament, i després pujar-lo fàcilment al telèfon.

6.1.2.- XML

XML, són les sigles de l'expressió "*Extensible Markup Language*", el que podríem denominar com llenguatge de marques generalitzat. Es tracta d'un llenguatge utilitzat per estructurar la informació en qualsevol document que contingui text, imatges, botons, gràfiques...

XML va ser desenvolupat pel World Wide Web Consortium, amb l'objectiu de simplificar el SGML per adaptar-lo a un camp molt precís: els documents d'Internet.

Però per a que és utilitzat el llenguatge XML, en el desenvolupament de aplicacions mòbils?

El llenguatge XML, és utilitzat per a definir l'estructura de l'aplicació, com l'orientació de l'objecte, del text i les característiques dels diferents objectes que componen l'aplicació: botons, imatges, text, layers...

Estructura etiqueta XML:

```
<?xml versió="1.0"?>
```

```
<Nombre objecte atributs etiqueta...> Contingut de l'etiqueta </Nombre objecte>
```

6.2.- Entorn d'execució

Tota aplicació desenvolupada, ha de ser programada sobre un entorn d'execució que ens permeti simular el sistema pel qual s'utilitzarà la nostre aplicació. Com pot ser un telèfon mòbil, simulant el sistema operatiu on s'executarà l'aplicació, com un sistema d'electrodomèstics, com és una petita pantalla amb un hardware mínim.

És el que anomenem Màquina Virtual o KVM (*kernel-based Virtual Machine*), que ens permet totes aquestes simulacions abans d'utilitzar la nostra aplicació a l'entorn real.

El KVM (*Kernel-based Virtual Machine*), forma part de la versió més reduïda de l'entorn d'execució i ve inclosa al software de la plataforma Java2 Micro Edition.

És utilitzat, per dispositius amb memòria i potencia de processament limitada, com per exemple els telèfons mòbils, PDAs... per oferir algunes funcions informàtiques habituals.

La KVM es molt similar a la JVM (*Màquina Virtual Java*), ja que totes dues són motors que executen les aplicacions i els applets que utilitzen la tecnologia Java.

6.3.- Software

A continuació es tractarà el software utilitzat pel desenvolupament de l'aplicació Android, juntament amb els complement necessaris per desenvolupar-la, com són el ADT Plug-in, el KVM o SDK de Android i les APIs de Google.

6.3.1.- Eclipse

Eclipse Open Software, es tracta d'una companyia líder de projectes basats en Software Lliure, que fomenta la comunitat de codi obert i facilita productes complementaris pel desenvolupament d'aquests, un d'ells és el que s'utilitzarà per desenvolupar la nostra aplicació, el Software Eclipse.

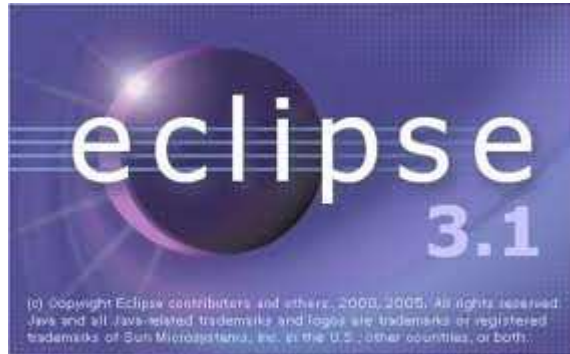


Figura 6.1. Imatge Software Eclipse

Eclipse, és un entorn de desenvolupament integrat de codi obert multiplataforma per desenvolupar “aplicacions de client”, basades en codi Java i XML per l'estructuració de l'aplicació.

Aquest entorn de desenvolupament, utilitza mòduls (plug-in), per proporcionar tota la seva funcionalitat a la plataforma de client, a diferència d'altres entorns monolítics, on totes les funcionalitats estan implementades, es necessitin o no.

En el cas del desenvolupament d'aplicacions per sistemes Android, aquests components es tracten del ADT plug-in i el SDK Android, que incorpora les API de Google necessàries per a cadascun dels SDK existents en el mercat.

Aquest software és definit pel projecte Eclipse com:

“una especie de herramienta universal – un IDE abierto y extensible para todo y nada en particular”.

6.3.2.- ADT plug-in

ADT (*Android Development Tool*), és una eina de desenvolupament per Android, dissenyada per la IDE de Eclipse. Oferint una gran capacitat de recursos per la construcció de aplicacions Android.

Aquest plug-in, augmenta les capacitats per poder configurar ràpidament nous projectes Android, creant una interfície d'usuari molt amigable, oferint components basats en les API de Android i oferint la possibilitat de depurar les aplicacions utilitzant les eines SDK de Android.

La instal·lació d'aquest paquet i l'eina Eclipse és la manera més ràpida de començar a desenvolupar aplicacions. Degut a que ofereix una estructura inicial de projecte, eines de integració, editors XML personalitzats i la depuració de l'aplicació amb el panel output.

Aquest plug-in dona un impuls increïble a Eclipse pel desenvolupament d'aplicacions Android.

6.3.3.- SDK Android + API

El SDK Android, és un paquet de desenvolupament necessari per programar i implementar tot tipus de aplicacions per Android. Aquest paquet de desenvolupament, inclou les APIs i les eines necessàries per desenvolupar les aplicacions utilitzant JAVA com a llenguatge de programació i testeig del codi, respectivament.

Aquesta llibreria és compatible amb molts altres entorns de programació, com són Eclipse, JDK5 o JDK6, Apache Ant...

Les APIs de Google, són una ampliació del entorn de desenvolupament del SDK de Android que proporciona a les aplicacions un accés senzill als serveis i a les dades de Google. La seva funció principal és la biblioteca externa de Google Maps, que permet incloure funcions d'assignació potents a les aplicacions.

Per poder utilitzar aquest complement es necessari tenir-lo instal·lat al SDK de Android. On ens permetrà accedir a les classes de la biblioteca i compilar les aplicacions.

Aquest complement també inclou una imatge del sistema Android, que serà executat al Emulador de Android i permetrà depurar, provar i ajustar el rendiment de l'aplicació abans de ser publicada al Market.

El complement API inclou:

- La biblioteca externa de Google Maps per Android 1.5.
- Una imatge del sistema Android.
- Una aplicació de Android de exemple, denominada "MapsDemo".
- Documentació de les classes.

6.4.- Extres

Pel desenvolupament de l'aplicació, CallMemory, s'ha disposat a utilitzar com a base, una aplicació ja creada anomenada BarCode Scanner, que permet la descodificació del CodiQR, a través d'una trucada a l'aplicació.

6.4.1.- BARCODE Scanner

BarCode Scanner, és una de les aplicacions amb millor acollida en el Android Market, aquesta és una aplicació que com el seu nom indica, es tracta d'un lector de diferents codis de barres (1D i 2D).

Aquesta utilitza la càmera del dispositiu mòbil i les funcionalitats d'Android per realitzar la captura del Codi.

Però com funciona aquesta aplicació?

El seu funcionament es bastant senzill a nivell extern, es tracta d'activar la càmera i alinear el codi desitjat al centre del requadre que surt i automàticament es disposa a capturar la imatge.



Figura 6.2. Imatge Aplicació BarCode Scanner

A nivell intern disposa d'una complexitat de codi molt elevada ja que treballa amb la llibreria ZXing, que disposa de les funcions per realitzar la descodificació dels diferents codis amb els que es capaç de treballar juntament amb els seus sistemes de correcció de dades, etc.

6.4.2.- ZXing

ZXing, anomenat “*zebra crossing*”, es tracta d’una llibreria Open-Source, de lectura de codis multi-format 1D/2D, implementada en Java. Disposa de clients per J2ME, J2SE i Android, que permet un cop escanejats i descodificats els codis, utilitzar la informació codificada ,les direccions URL, números de telèfon, correus electrònics.



Figura 6.3. Logotip ZXing

7. Desenvolupament de l'Aplicació.

Per començar a desenvolupar l'aplicació, el primer que s'ha de realitzar es valorar els requeriments que necessitem per desenvolupar l'aplicació.

Al tractar-se d'una programació per a telefonia mòbil, s'ha decidit utilitzar l'eina de desenvolupament "Eclipse".

Aquesta eina és la més utilitzada pels entorns de programació de telefonia mòbil i pel que es demana al projecte que és desenvolupar una aplicació per Android.

A continuació es tractaran, els requeriments que es necessiten per poder començar a desenvolupar l'aplicació per entorns Android, i es mostrarà la instal·lació i configuració dels diferents plug-in's i components necessaris, ADT PLUG-IN + KVM Android (Simuladors Sistema Operatiu d'Android+API), pel desenvolupament d'aquesta.

7.1.- Requeriments de Desenvolupament.

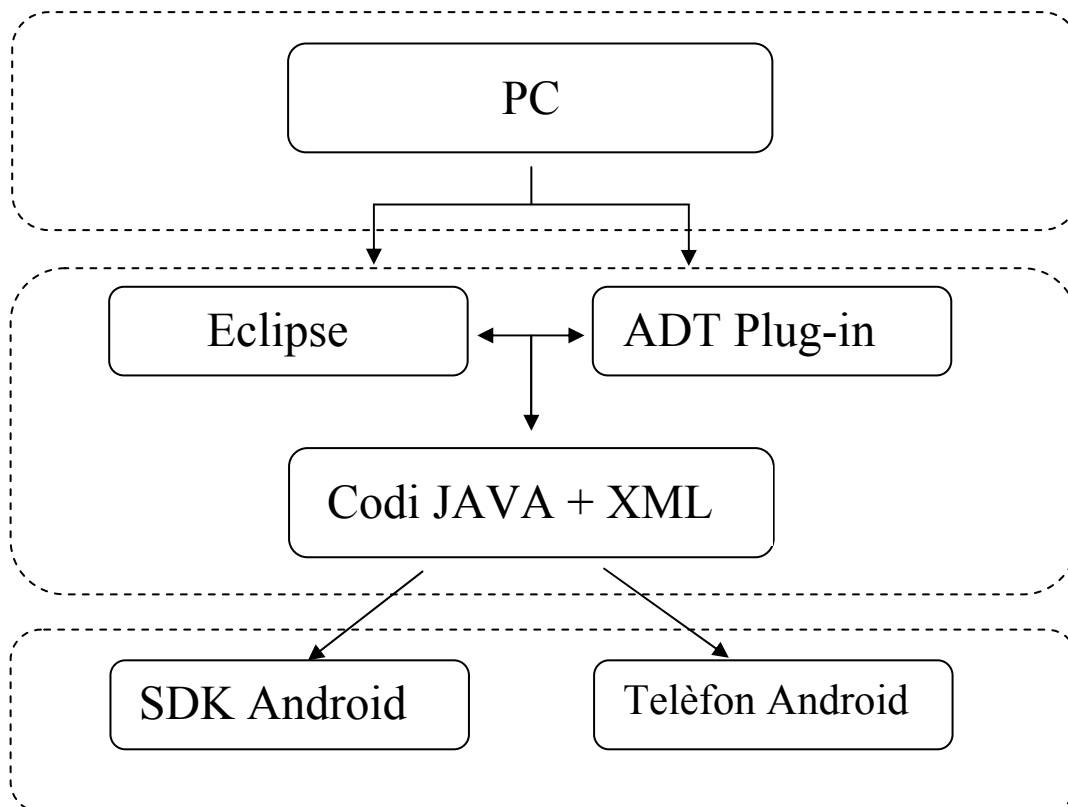


Figura 7.0. Requeriments desenvolupament Aplicació

Per començar el primer que s'ha de tenir clar, són els requeriments que es necessiten per començar a desenvolupar una aplicació per Sistemes Operatius Android.

Inicialment, el requeriment principal serà disposar d'un ordinador personal, en el que es treballarà, i on s'instal·larà tot el programari necessari .

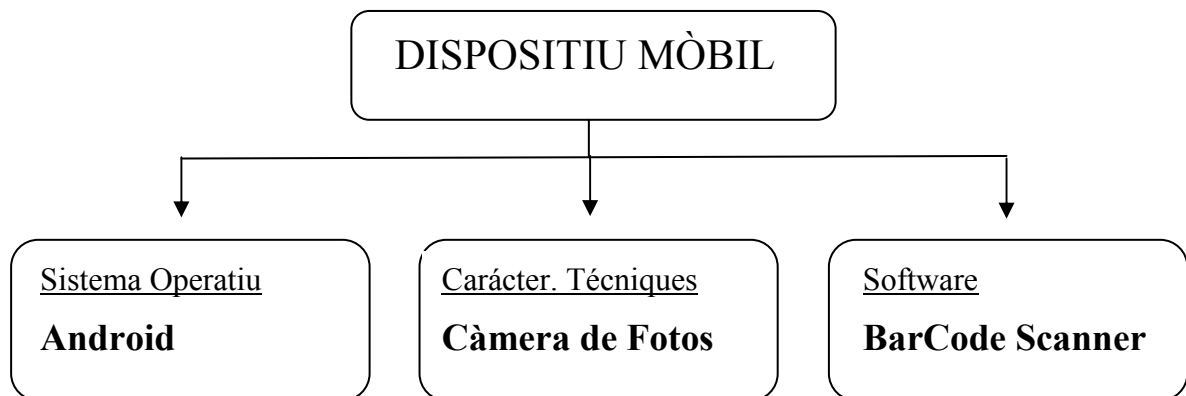
Aquest programari estarà compost per l'eina de desenvolupament JAVA i XML, Eclipse comentada anteriorment en el punt, 6.3.1. Eclipse. Que permetrà treballar en un entorn dinàmic, tant pel treball en codi JAVA com en XML. Juntament amb un plug-in, anomenat ADT Plug-in, que incorporarà totes les eines necessàries pel desenvolupament d'una aplicació Android.

I finalment, es necessitarà un entorn d'execució/proves, per l'aplicació desenvolupada, que es pot tractar de l'eina SDK Android, que permet simular una màquina virtual Android, especificant les característiques tècniques. O la utilització d'un telèfon mòbil amb OS Android.

Es aconsellable disposar d'un telèfon mòbil, amb OS Android, degut a que Eclipse més la eina KVM, acostuma a quedar-se bloquejat.

7.1.1.- Requeriments del Dispositiu Mòbil.

Un cop realitzat els requeriments, per poder realitzar el desenvolupament de l'aplicació, s'observa que un d'ells és un dispositiu mòbil o una KVM de Android utilitzant el SDK Android. Pero quins requeriments a de disposar el dispositiu mòbil?



El dispositiu mòbil, com es pot observar en el gràfic anterior, requereix disposar del sistema operatiu Android, que es per el qual s'està realitzant el estudi. Disposar com a característica tècnica, d'una càmera fotogràfica incorporada al dispositiu per poder realitzar la captació de la imatge del codi QR, i finalment disposar de l'aplicació Barcode Scanner, per aconseguir la descodificació de la imatge capturada mitjançant la càmera del dispositiu.

7.2.- Instal·lació Eclipse.

El primer pas que serà anar a la pàgina oficial de Eclipse: <http://www.eclipse.org/> , i procedir a la descàrrega del software. Al tractar-se d'un projecte Open-Source, és una aplicació gratuïta.

S'observarà que només accedir a la pàgina oficial, es trobarà a la dreta de la pantalla, un link de descàrrega de l'aplicació Eclipse:

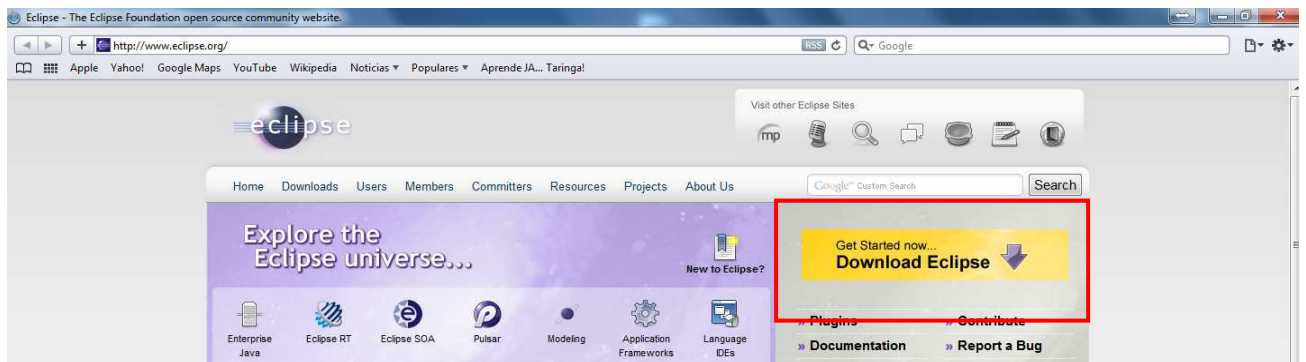


Figura 7.1. Pàgina inici www.eclipse.org

Al tractar-se d'una eina de desenvolupament JAVA, es trobaran diferents versions per descarregar, des de les versions més específiques, com versions ja preparades per començar a desenvolupar amb tot els complements.

Pel treball que s'ha de realitzar, descarregant la versió : Eclipse IDE for Java Developers (99 MB) hi haurà suficient.

Un concepte important abans de inicialitzar la descàrrega, és tenir en compte la versió de sistema Operatiu amb el que es treballarà, a l'observar que tenim dues versions diferents del programa segons el nivell de bus del nostre Sistema Operatiu, 32 o 64 bits.

The screenshot shows the Eclipse Downloads page. The main content area lists several software packages for download. The 'Eclipse IDE for Java EE Developers' package is highlighted with a red box. The package details include the name, size (206 MB), and download count (868,609 Times). The download links are provided for Windows 32 Bit and Windows 64 Bit. The page also includes a navigation menu at the top, a search bar, and a sidebar with additional resources like 'Master BIRT at BIRT Exchange' and 'Installing Eclipse'.

Figura 7.2. Versió del Software a descarregar

NOTA: S'ha d'assegurar tenir instal·lat la consola JAVA a la màquina, sino Eclipse no funcionarà.

Per descarregar la màquina Java: <http://java.sun.com/javase/downloads/index.jsp>

The screenshot shows the Oracle Java SE Development Kit 6u25 with JavaFX 1.3.1 download page. The page features the Oracle logo at the top, a navigation menu, and a search bar. The main content area displays the download link for the JDK 6u25, JavaFX 1.3.1, Windows version. The page also includes a license agreement and a list of Java SDKs and Tools.

Figura 7.3. Instal·lació Màquina Java.

Un cop descarregat, s'accedirà a l'arxiu i es farà doble click sobre: eclipse.exe.



Figura 7.4. Primera execució de Eclipse

El primer que apareixerà, serà l'opció d'escollir on es vol que estigui instaurat, el "Workspace", es a dir el directori de treball, on volem que es guardin els projectes desenvolupats.

Sempre es podrà modificar segons el desig de la persona, o per si volem carregar un Workspace, on es trobin els projectes ja desenvolupats.

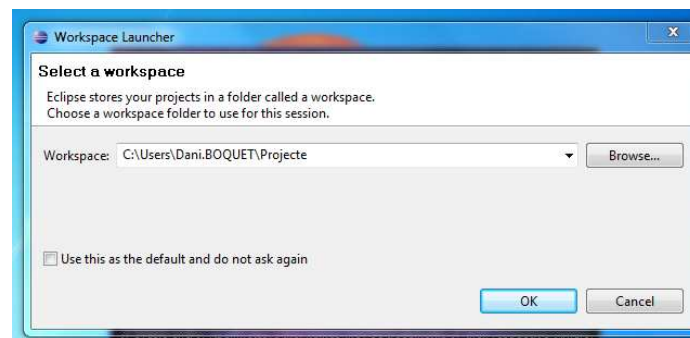


Figura 7.5. Escollir el directori del Workspace



Figura 7.6. Càrrega de Eclipse

Amb aquest tres passos, es disposarà d'una eina de desenvolupament de projectes JAVA, a la computadora. En aquest cas Eclipse:

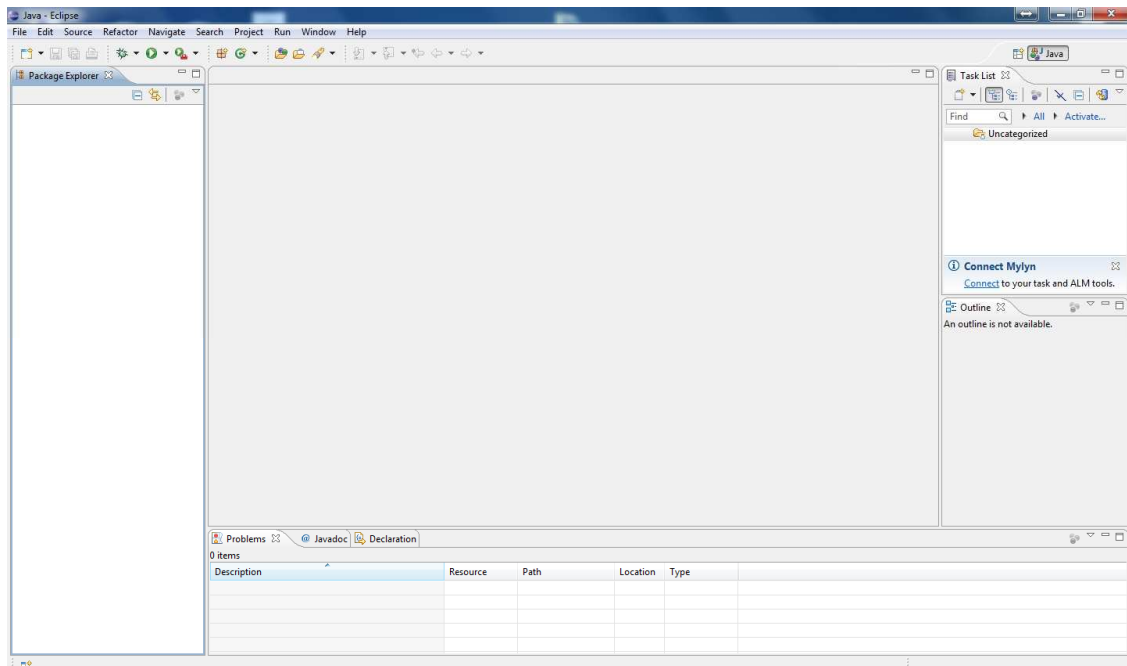


Figura 7.7. Entorn principal de Eclipse

7.3.- ADT Plug-in (Android Development Tools).

Per poder desenvolupar una Aplicació Android, no només es necessita l'eina de desenvolupament Java Eclipse, sinó que aquesta necessita uns plug-in's (plug-in: annexa un programa a un altre per augmentar les seves funcionalitats. No es tracta d'un "parche" ni una actualització, sinó un mòdul opcional).

Per a realitzar el projecte Android, es necessitarà, instal·lar el mòdul dels plug-ins's Android (eines de Android), anomenada ADT Plug-in.

Per procedir a l'aplicació d'aquest plug-in ADT es seguirà el següent procediment:

- Executem Eclipse, y seleccionem **Ajuda -> Instal·lar nou Software.**

Es trobarà una finestra per introduir un text, amb l'opció incloure "Add" al quadre de diàleg Software disponible, es polsarà Incloure:

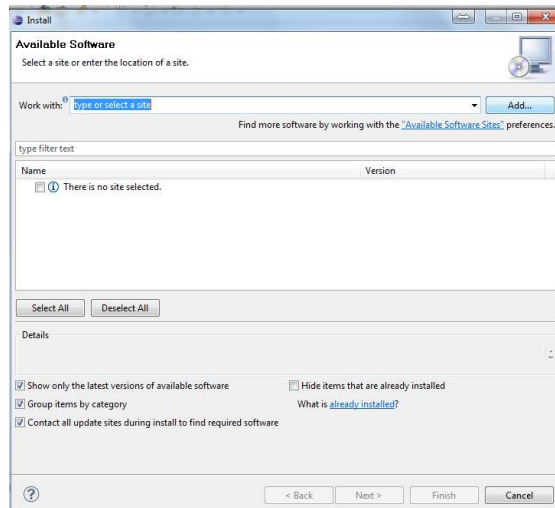


Figura 7.8. Introducció del Plug-in ADT

Al diàleg Add site, que apareix, s'introduirà un nom pel nostre Site remot, per exemple "Android Plug-in" al camp nom.

Al camp "Ubicació" introduïrem la següent direcció:

<https://dl-ssl.google.com/android/eclipse/>

Nota: Si existeix algun problema alhora d'adquirir el plug-in, s'introduirà la direcció següent:

<http://dl-ssl.google.com/android/eclipse/>

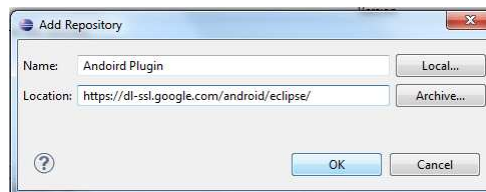


Figura 7.9. Especificació URL.

Al pulsar el botó "OK", es retornarà a la pàgina de software disponible, on s'observarà una nova nota afegida a la llegenda: "Developer Tools".

El següent pas serà seleccionar la casella per activar la instal·lació dels complements, es seleccionaran els complements desitjats per l'aplicació. I es pulsarà Següent...

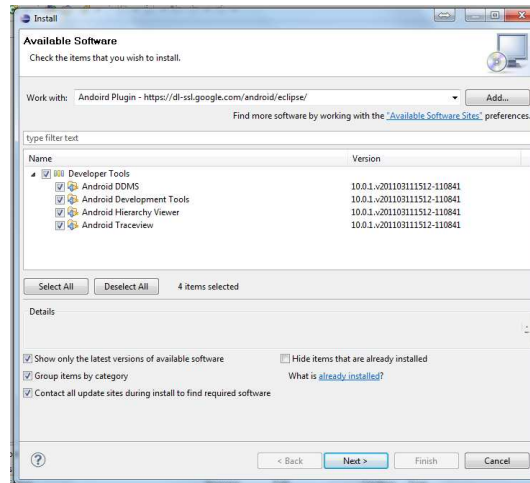


Figura 7.10. Selecció dels complementos desitjats.

Finalment, es mostraran els permisos i les llicències dels complementos que es volen instal·lar, un cop llegits es pulsarà Ok si s'està d'acord i es començarà la instal·lació.

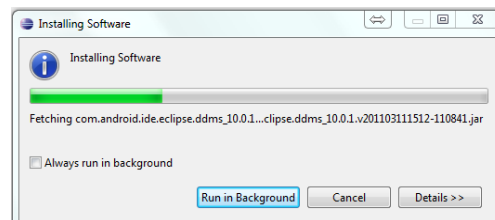


Figura 7.11. Instal·lació complementos ADT.

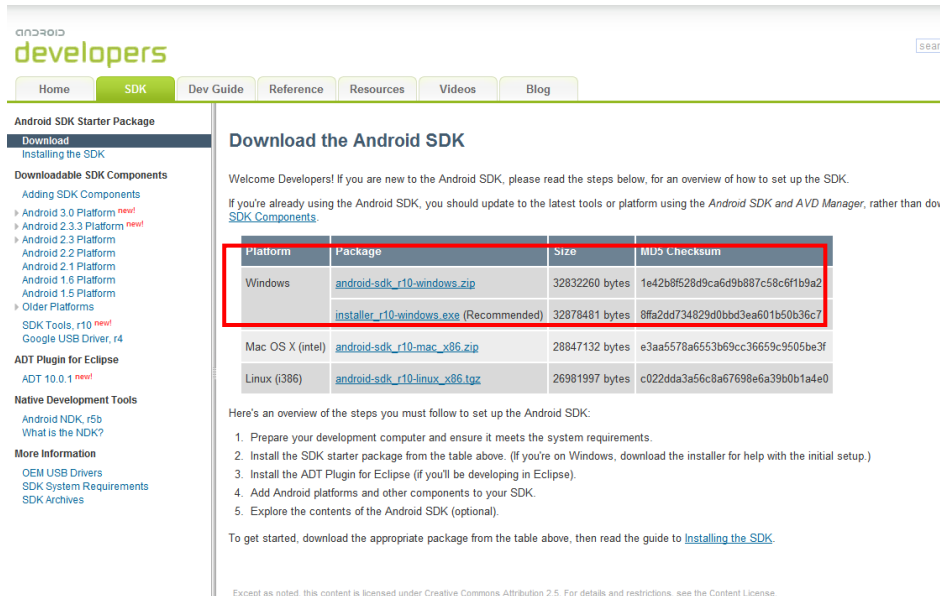
Un cop finalitzada la instal·lació, es procedirà a reiniciar el software Eclipse.

7.3.1.- Configurar el plug-in ADT

Una vegada executat novament Eclipse, es procedirà a realitzar la correcta configuració dels complementos del plug-in ADT, per que aquests apuntin al directori on es troba el Android SDK.

Però per poder realitzar la següent tasca s'haurà de disposar del Android SDK. Per poder disposar d'ell es tindrà que descarregar des de la següent URL:

http://developer.android.com/sdk/1.5_r2/index.html



Download the Android SDK

Welcome Developer! If you are new to the Android SDK, please read the steps below, for an overview of how to set up the SDK.

If you're already using the Android SDK, you should update to the latest tools or platform using the *Android SDK and AVD Manager*, rather than download *SDK Components*.

Platform	Package	Size	MD5 Checksum
Windows	android-sdk_r10-windows.zip	32832260 bytes	1e42b8f528d9ca6d9b887c58c6f1b9a2
	installer_r10-windows.exe (Recommended)	32878481 bytes	8fa2dd734829d0bbd3ea601b50b36c7
Mac OS X (intel)	android-sdk_r10-mac_x86.zip	28847132 bytes	e3aa5578a6553b69cc36659c9505be3f
Linux (386)	android-sdk_r10-linux_x86.tgz	26981997 bytes	c022dda3a56c8a67698e6a39b0b1a4e0

Here's an overview of the steps you must follow to set up the Android SDK:

1. Prepare your development computer and ensure it meets the system requirements.
2. Install the SDK starter package from the table above. (If you're on Windows, download the installer for help with the initial setup.)
3. Install the ADT Plugin for Eclipse (if you'll be developing in Eclipse).
4. Add Android platforms and other components to your SDK.
5. Explore the contents of the Android SDK (optional).

To get started, download the appropriate package from the table above, then read the guide to [Installing the SDK](#).

Except as noted, this content is licensed under [Creative Commons Attribution 2.5](#). For details and restrictions, see the [Content License](#).

Figura 7.12. Pàgina principal Android Developers

Un cop finalitzada la descàrrega i executada la instal·lació d'aquesta, un dels punts més importants és el que atorgarà el directori arrel, on realitzarem l'instal·lació, ja que aquest directori serà el que introduïrem a la configuració del ADT, del software Eclipse.

Per defecte es pot observar que el directori arrel serà el següent:

C:\Program Files\Android\android-sdk



Figura 7.13. Especificació del directori de l'instal·lació.

Un cop finalitzada l'instal·lació, s'executarà el SDK Setup, on s'observaran tots els complements Android, que permet la instal·lació

El següent a realitzar serà seleccionar i acceptar tots els mòduls SDK i API's que interessin per realitzar l'aplicació. Es trobaran tots els SDK diferents que es poden trobar en Android, 1.5, 1.6, 2.1, 2.2 ... i altres complements com els de google earth.

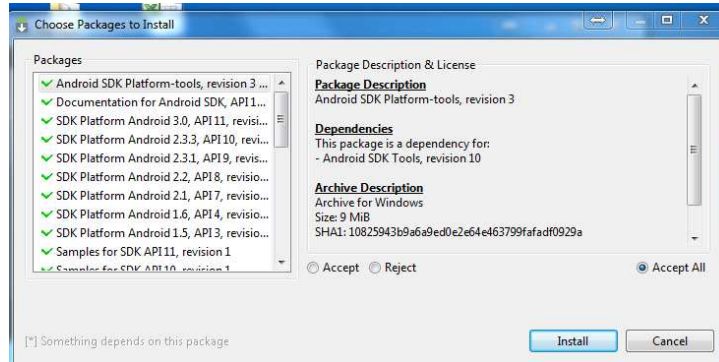


Figura 7.14. Diferents SDK i API's Android

Un cop finalitzada la instal·lació i configuració del SDK Android, el següent serà continuar amb la configuració del ADT plug-in introduint el directori, el qual a d'apuntar per trobar els diferents complements del SDK Android.

Per realitzar aquesta configuració els passos a seguir seran:

Es selecciona: Ventana -> Preferencias...

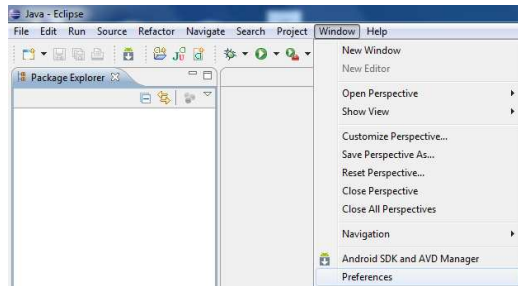


Figura 7.15. Accés a Preferencies de Eclipse.

Es selecciona l'opció Android del panell esquerre, i es procedeix a introduir el directori al que apuntarà el ADT per trobar els diferents SDK de Android instal·lats. El directori a introduir serà l'establert a la instal·lació del SDK Android.

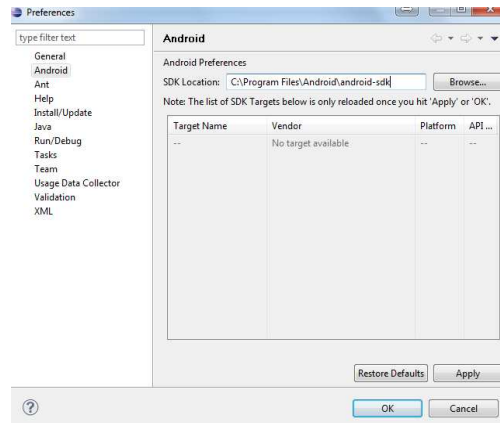


Figura 7.16. Especificació del directori del SDK Android.

Un cop introduït el directori, s'actualitzarà la finestra, introduint tots el SDK i API's instal·lats, que podran ser utilitzats per realitzar les aplicacions.

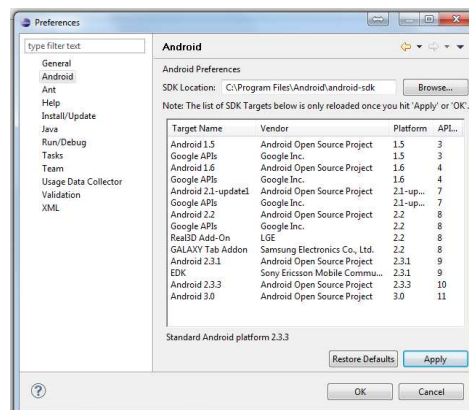


Figura 7.17. SDK i API's compatibles.

7.4.- Creació KVM Android.

Un cop realitzats els passos anteriors, es podrà inicialitzar la programació del projecte, encara que trobarem el problema de no disposar d'una màquina de simulació, per executar l'aplicació, en un entorn de proves.

Per poder disposar d'un entorn de proves, es pot realitzar de dues formes diferents. La primera disposant d'un dispositiu mòbil que ens permeti treballar en format debug, avui en dia tots els dispositius amb entorn Android disposen d'aquesta opció. O la segona serà la creació d'una màquina virtual mitjançant el KVM Android.

A continuació es tractarà la realització d'una màquina virtual on poder simular l'aplicació per comprovar el seu correcte funcionament.

Per començar la configuració de la Màquina Virtual, accedirem a la finestra:

Windows → Android SDK and AVD Manager:

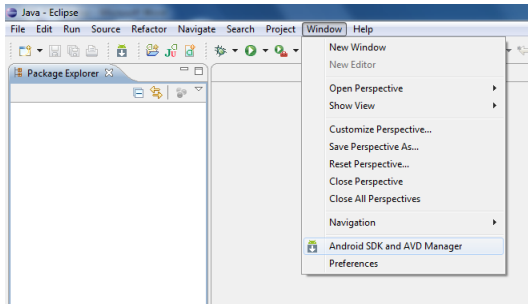


Figura 7.18. Pestanya Menú Windows

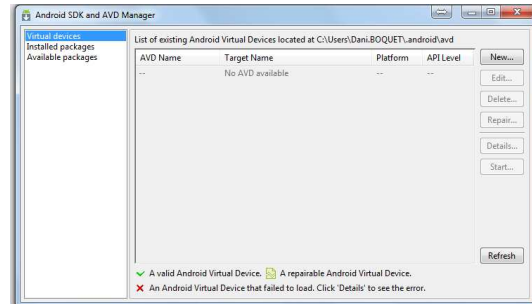


Figura 7.19. Finestra Android SDK ...

Pulsar el botó “New” per començar a definir les característiques tècniques que tingui el dispositiu, a continuació es mostraran les diferents opcions:

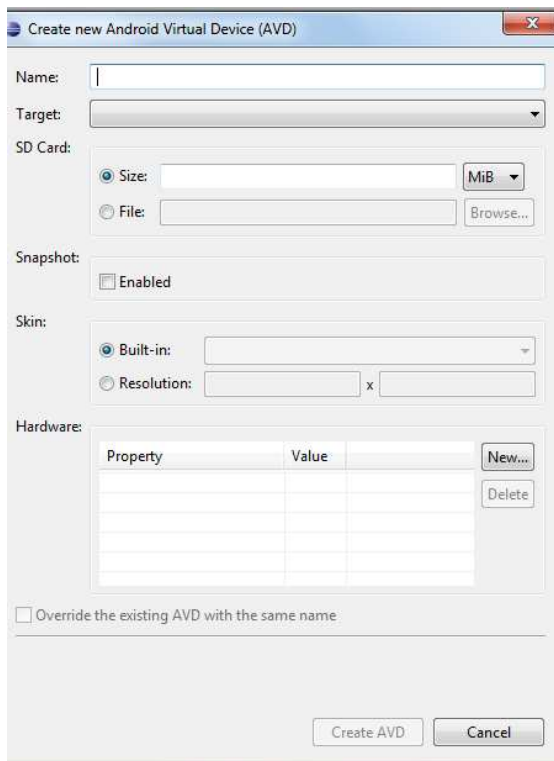


Figura 7.19. Creació AVD

- **Name**_ Nom de la màquina Virtual Android (AVD).
- **Target**_ Sistema Operatiu (SDK) amb les seves eïnes, que contindrà la AVD. Apareixeran tots el instal·lats, amb els software SDK Android.
- **SD Card**_ Memòria virtual que disposarà la màquina, simulant una targeta SD.
- **Snapshot**_ Activació de càmera.
- **Skin**_ Definició de la resolució de la pantalla del dispositiu Android.

- **Hardware**_ per últim la definició del hardware del que disposarà la màquina, oferint una gran quantitat de component, aconseguint així la configuració necessària per les necessitats de l'aplicació.

8. Creació del projecte.

Un cop disposem dels coneixements necessaris, tant a nivell teòric com a nivell informàtic amb els llenguatges de programació, i les eines de desenvolupament, es procedirà al desenvolupament de l'aplicació CallMemory. Definint un nou projecte a l'eina de desenvolupament Eclipse.

8.1.- Creació i estructura del Projecte

El primer pas serà la definició del projecte, definint el nom desitjat pel projecte, la Target pel que desenvoluparem l'aplicació, el nom del Package i el nom de la Activitat que ens crearà automàticament.

Per crear el projecte, s'anirà a File i New Android Project:

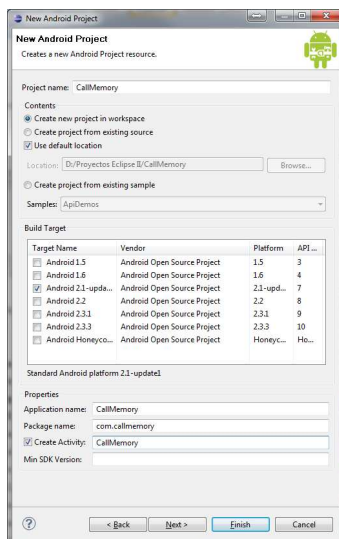


Figura 8.1. New Android Project.

Observarem que ens a definit un projecte amb el nom CallMemory, amb una estructura d'arbre, on trobarem les diferents àrees i on es treballarà l'aplicació a desenvolupar.

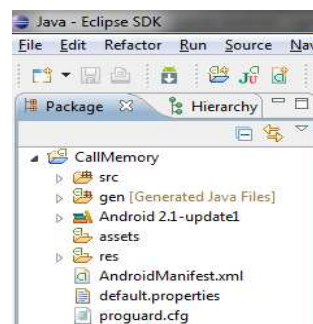


Figura 8.2. Estructura Projecte Android

- **Project name**_ CallMemory.
- **Conténts**_ Create new project in workspace.
- **Build Target**_ Android 2.1-update1.
- **Properties**_ Application name: CallMemory, Package name: com.callmemory, Create Activity: CallMemory.

Les diferents àrees són les següents:

- **/src/**_ conté tot el codi font de l'aplicació, codi de la interfície gràfica, classes auxiliars, etc. Inicialment, Eclipse crearà el codi bàsic de la pantalla (Activity) principal d l'aplicació.

- **/res/** _ Conté tots els fitxers de recursos necessaris pel projecte, com són imatges, vídeos, cadenes de text, etc. Aquests recursos són distribuïts entre les següents carpetes:
 - **/res/drawable/**_ Conté les imatges de les aplicacions.
 - **/res/layout/**_ Conté els fitxers de definició de les diferents pantalles de la interfície gràfica. La interfície gràfica serà definida mitjançant etiquetes de codi XML.
 - **/res/anim/**_ Conté la definició de les animacions utilitzades per l'aplicació.
 - **/res/menú/**_ Definició dels menús de l'aplicació.
 - **/res/values/**_ Conté altres recursos de l'aplicació, com la definició de les cadenes de text (strings.xml), els estils (styles.xml), els colors (colors.xml)...

El contingut de la carpeta res, de l'aplicació CallMemory, disposarà dels següents continguts:

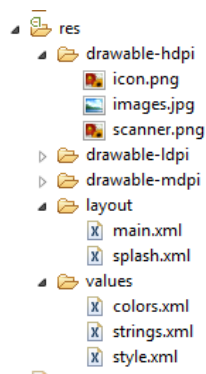


Figura 8.3. Estructura /res/ CallMemory

- **/gen/** _ Conté una sèrie d'elements de codi generats automàticament al compilar el projecte. Cada vegada que generem un projecte, la màquina de compilació Android genera una sèrie de fitxers font, en codi Java, dirigits al control dels recursos de l'aplicació.

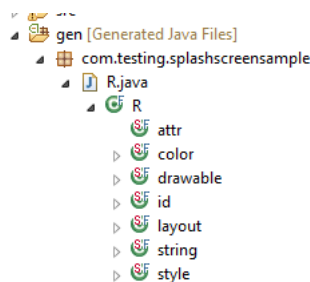


Figura 8.4. Estructura /gen/ Callmemory

El fitxer més important és el que es pot observar a la imatge, el fitxer R.java i la classe R.

Aquesta classe disposa en tot moment d'una sèrie de constants amb els ID de tots els recursos de l'aplicació.

- **/assets/** _ Conté la resta de fitxers auxiliars necessaris per l'aplicació, com per exemple els fitxers de configuració, de dades, etc.

La diferència entre els recursos de la carpeta `/res/raw/` i els de la carpeta `/assets/` és que en els primers es genera un ID a la classe R i és pot accedir a ells mitjançant diferents mètodes d'accés al recurs. Però pels segons no es genera ID i s'accedirà a ells per la seva ruta com qualsevol altre fitxer del sistema.

- **Fitxer AndroidManifest.xml** _ Conté la definició en XML dels aspectes principals de l'aplicació com per exemple la seva identificació (nom, versió...), els seus components (pantalles, missatges...), o els permisos necessaris per la seva execució, en el cas de l'aplicació CallMemory, permisos de Call_phone o Càmera.

8.2.- Arquitectura de classes

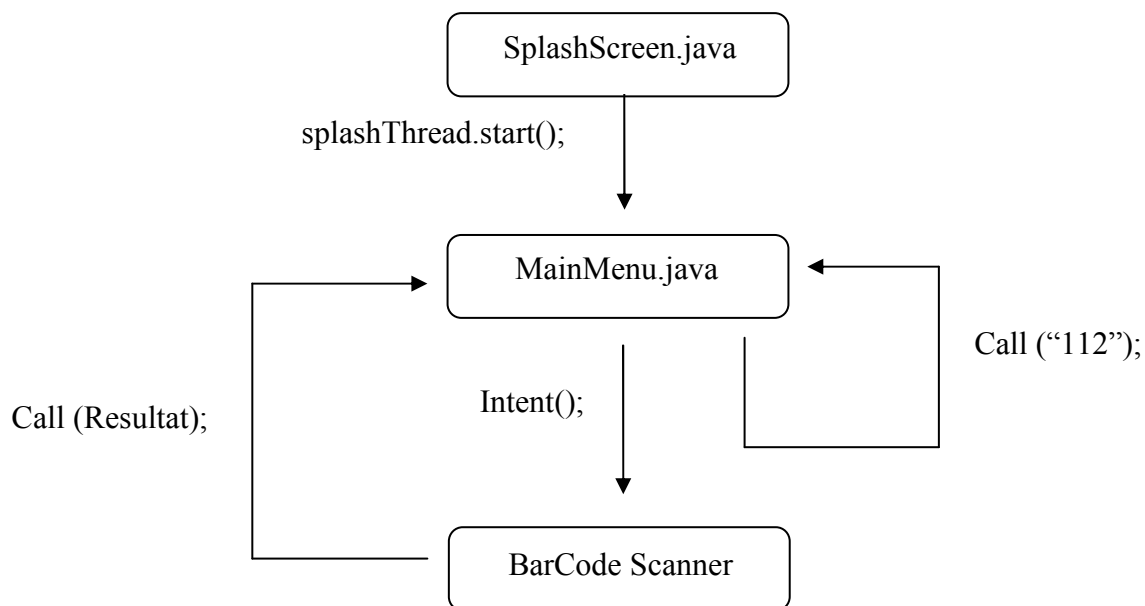


Figura 8.4.b. Diagrama de classes

Per explicar el correcte funcionament, de l'aplicació CallMemory, es pot observar en la Figura 8.4.b. Diagrama de classes, quin és el procés que segueix l'aplicació a l'hora de realitzar una tasca, i com varia per les diferents classes creades.

Inicialment, l'aplicació executa la classe, SplashScreen.java, en la que s'executa la visibilitat del Layout, Splash.xml, que es descriu en el apartat 8.4. Layouts, alhora que executa la funció wait(), que realitzà una espera de 5 segons, abans d'inicialitzar

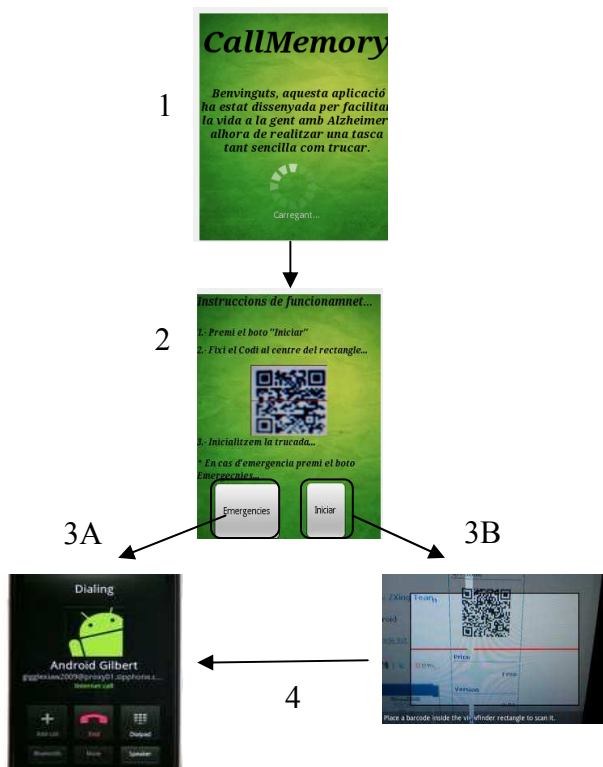
l'execució de la funció `splashThread.start()`, que realitza el canvi a la classe, `MainMenu.java`. Aquest codi és vist a l'apartat 8.4.1. `SplashScreen`. Un cop executada la classe `MainMenu.java`, es procedeix a executar la visibilitat del `Layout`, `Main.xml`, en el que es troba el cos real de l'aplicació, també definida al punt 8.5. `Layouts`.

Com s'observa aquesta classe disposa de dues opcions a executar, segons el botó que es procedeix a escoltar. Una primera opció serà escoltar el botó `Iniciar`, que executa l'aplicació externa `BarCode Scanner`, que realitza la tasca de captura i descodificació del Codi QR, oferint com a resultat de la descodificació un valor `String`, que es enviat a la funció `call(String)`, realitzant la trucada al número descodificat, o retornant a la classe `mainMenu.java`, si el codi no ha estat possible de descodificar.

I com a segona opció, es pot escoltar el botó `Emergències`, que procedeix a enviar directament el valor "112" a la funció `call (String)`, que és l'escollit per realitzar una trucada de emergència. Un cop finalitzada aquesta trucada es torna a la classe `MainMenu.java`.

8.3.- Casos us.

Després de poder observar l'arquitectura de classes de l'aplicació `CallMemory`, i una breu explicació, dels estats de cadascuna d'elles,. Es procedirà a realitzar una descripció gràfica dels casos d'ús d'aquesta:



1.- Inicialització de l'aplicació `CallMemory`.

2.- Inicialització del `MainMenu`, un cop passats 5 segons.

Tenim l'opció de seleccionar o l'opció `Emergències` (3A) o l'opció `Iniciar` (3B).

3.A.- És procedeix a realitzar una trucada al número 112 "Emergències".

3.B.- Execució de l'aplicació `Barcode Scanner`, per realitzar la descodificació.

4.- Es procedeix a realitzar la trucada al número descodificat per el `BarCode`.

8.4.- Fitxers Values

Per facilitar la creació de l'aplicació CallMemory, es aconsella la realització dels fitxers color.xml, strings.xml i style.xml.

Aquests facilitaran les tasques de manteniment de l'aplicació, així com en cas de voler modificar qualsevol text, estil del text o color del text, només es modificarà la declaració de les etiquetes XML, definides en els fitxers anteriors evitant la modificació del codi Java, de l'aplicació ja creada.

En el cas de l'aplicació CallMemory el contingut dels fitxers serà els següents:

Color.xml:

```
<resources>
  <color name="rojo">#f00</color>
  <color name="verde">#0f0</color>
  <color name="azul">#00f</color>
  <color name="blanco">#fff</color>
  <color name="negro">#000</color>
  <color name="amarillo">#ffff00</color>
  <color name="magenta">#ff00ff</color>
</resources>
```

Strings.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <string name="hello">Hello World, activity!</string>
  <string name="app_name">CallMemoRy</string>
  <string name="mScan">mScan</string>
  <string name="Inici">Inici</string>
  <string name="Emergencies">Emergencies</string>
  <string name="loading">Carregant...</string>
  <string name="introduccio"> Benvinguts, aquesta aplicació ha
  estat dissenyada per facilitar la vida a la gent amb Alzheimer,
  ahora de realitzar una tasca tant sencilla com trucar.</string>
</resources>
```

Style.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <style name="StyleText1">
    <item name="android:textSize">20sp</item>
    <item name="android:textColor">#EC9200</item>
    <item name="android:gravity">center_horizontal</item>
  </style>
  <style name="StyleText2">
    <item name="android:textSize">200sp</item>
    <item name="android:textColor">#f00</item>
    <item name="android:gravity">center_horizontal</item>
  </style>
  <style name="StyleButton1">
    <item name="android:layout_width">wrap_content</item>
    <item name="android:layout_height">wrap_content</item>
  </style>
```

8.5.- Layouts



Figura 8.5. Layout Splash



Figura 8.6. Layout Main

El desenvolupament de l'aplicació CallMemory, a estat realitzat amb una pantalla inicial anomenada Layout Splash, que serà carregada durant un instant de temps definit, que en el punt 8.5.1 del capítol Codi Java, es demostrà com ha estat programat.

En aquest Layout, que està estructurat pel nom de l'aplicació, una petita descripció de la finalitat d'aquesta, i un ProgressBar, que realitza la tasca de simular la càrrega de l'aplicació, aquest a estat definit amb el següent codi:

```
<ProgressBar android:layout_gravity="center"
    android:id="@+id/progressBar1"
    android:layout_height="wrap_content"
    android:layout_width="wrap_content"
    android:layout_marginRight="5dp"
    style="@android:style/Widget.ProgressBar.Large">
</ProgressBar>
<TextView android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:text="@string/loading"
    android:layout_height="wrap_content"
    android:layout_gravity="center">
</TextView>
```

Un cop passat aquest instant de temps, es carrega un segon Layout, anomenat "Main" on es pot observar l'estructura principal de l'aplicació. Aquesta esta estructurada per una definició de les instruccions a seguir pel correcte funcionament de l'aplicació, doncs al tractar-se d'una aplicació que esta destinada a gent gran i amb problemes de memòria, sempre va bé fer una menció al funcionament de l'aplicació.

Es pot observar una imatge carregada, com a exemple del correcte escaneig d'un codi QR, mitjançant l'aplicació BarCode Scanner.

I finalment, dos botons: un que executa l'aplicació BarCodeScanner per realitzar el escaneig del codi i un segon que permet trucar automàticament al número de emergències 112.

A continuació, al punt 8.5. Codi Java, es veurà com a estat desenvolupat el codi Java per aconseguir el correcte funcionament de la trucada a l'aplicació BarCode Scanner i de la funció trucar, a través d'escoltar el click del botó desitjat.

8.6.- Codi JAVA

El codi Java de l'aplicació CallMemory a estat dividit en dues classes, una per a cadascun dels Layouts, pel Layout Splash, s'anomenarà Splashscreensample, i pel Layout Main la classe Mainmenu. Poden diferenciar així les tasques que es realitzen en cadascun d'ells.

A continuació es tractarà el codi més important utilitzat en cadascun d'aquests fitxers per realitzar les tasques, com són: les del SplashScreen la funció Call(), la crida a l'aplicació BarCode Scanner, o com es defineix la funció de escoltar l'acció d'un botó.

8.6.1.- SplashScreen

Per realitzar el SplashScreen, del Layout Inicial, hem treballat amb la classe Thread i Intent, que disposa les llibreries de codi Java, realitzant l'import d'un parell de package necessaris.

La tasca d'aquest codi és la de posar visible el Layout Splash inicialment, definir un wait de 5 segons, i després executar la classe MainMenu, on s'executarà el Layout Main.

Codi SplashScreenSample:

```
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;

public class SplashScreen extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.splash);
        Thread splashThread = new Thread() {
            @Override
            public void run() {
                try {
```

```

        int waited = 0;
        while (waited < 5000) {
            sleep(100);
            waited += 100;
        }
    } catch (InterruptedException e) {
    } finally {
        finish();
        Intent i = new Intent();
        i.setClassName("com.testing.splashscreensample",
"com.testing.splashscreensample.mainmenu");
        startActivity(i);
    }
}
};
splashThread.start();
}}

```

8.6.2.- Crida BarCode Scanner

Com a estat comentat anteriorment, pel correcte funcionament de l'aplicació CallMemory, a estat utilitzada una segona aplicació anomenada BarCode Scanner, que realitzarà la tasca de descodificació del codi QR.

Per poder utilitzar aquesta segona aplicació, ha estat definida una crida a aquesta a través d'escoltar el click d'un dels botons definits en el Layout Main, en aquest cas el botó amb ID igual a Iniciar.

```

...
private
    Button CallButton;
...

    CallButton = (Button) findViewById(R.id.iniciar);
    CallButton.setOnClickListener(callm);
...

public Button.OnClickListener callm = new Button.OnClickListener() {
    public void onClick(View v) {
        Intent intent = new
        Intent("com.google.zxing.client.android.SCAN");
        intent.putExtra("SCAN_MODE", "QR_CODE_MODE");
        startActivityForResult(intent, 0);
    }
};

public void onActivityResult(int requestCode, int resultCode, Intent
intent) {
    if (requestCode == 0) {
        if (resultCode == RESULT_OK) {
            String conténts =
            intent.getStringExtra("SCAN_RESULT");
            String Resultado =
            intent.getStringExtra("SCAN_RESULT_FORMAT");
            Call();
        } else if (resultCode == RESULT_CANCELED) {
            String error= "ERROR: No ha estat possible,
            escanejar el codi";
        }
    }
}
}

```

8.6.3.- Funció Call (String string)

Definitivament, un cop realitzada la descodificació del codi QR, a través de la crida a l'aplicació BarCode Scanner, l'aplicació a obtingut un resultat en format String. Aquest resultat serà enviat a la funció Call(), on realitzarà la tasca de trucar. Aquesta funció es la següent:

```
private void Call(String string) {  
    Intent callNumber = new Intent();  
    callNumber.setAction(android.content.Intent.ACTION_CALL);  
    callNumber.setData(Uri.parse("tel: " + string));  
    startActivity(callNumber);  
}
```


9. Possibles ampliacions del projecte.

Com a possibles ampliacions del projecte CallMemory, es podrien oferir noves funcionalitats per l'eina com:

- Enviar missatges a la persona desitjada a través d'un entorn agradable per a persones grans.
- La segona possible ampliació seria aconseguir la realització de video conferència, oferint la possibilitat de poder contactar via telefónica amb familiars, metges, emergencies,etc..
- Però una de les grans ampliacions possibles, al tractar-se d'una aplicació per a gent gran i aprofitant les eines que ens ofereix la programació per a sistemes Android, tractant-se del sistema operatiu de Google, seria la possibilitat de posar en contacte a la persona amb qualsevol altre oferint la seva posició a través de les eines de desenvolupament de Google Maps facilitades per les API de Google. Oferint en cas d'emergència la posició en un mapa de la persona.

10. Conclusió

Finalment, a estat possible la realització dels objectius inicials del projecte, aconseguint desenvolupar correctament l'aplicació per a sistemes Android "Call Memory".

Una aplicació que la seva funcionalitat era aconseguir realitzar una trucada mitjançant la descodificació d'un codi QR.

Per realitzar aquest desenvolupament ha estat utilitzat el software Eclipse, una eina de programació Java i XML juntament amb els complements i llibreries necessaris pel desenvolupament d'aplicacions Android, el plug-in ADT i el complement SDK Android.

A més a més s'ha utilitzat una segona aplicació anomenada Barcode Scanner, que facilita la descodificació del codi QR mitjançant la utilització de les llibreries ZXing.

La aplicació en definitiva, ha estat desenvolupada correctament i compleix els requisits que es demanaven al projecte.

10.1.- Conclusió Personal

Com a conclusions es pot dir que el treball de crear una aplicació per a mòbils a sigut una tasca entretinguda, que m'ha donat la oportunitat de descobrir un nou entorn de programació en el que poder experimentar i aprendre a la vegada, però laboriosa degut a que els meus coneixements són més telemàtics que informàtics el que ha suposat més dificultat a l'hora de fer un projecte que requeria uns coneixements més alts de programació que els obtinguts als meus anys acadèmics.

Per una altra banda també resulta interessant el descobriment del diferents tipus de codis existents, ja que a l'actualitat estem només acostumats a veure els típics codis lineals i en pocs casos algun codi bidimensional.

Per últim ha sigut molt important poder desenvolupar una aplicació que, a banda de tenir un interès tècnic per mi també té una finalitat social, ja que ajuda a persones amb problemes psíquics a ser més autònoms i independent.

11. Bibliografia

Que és un codi de barres:

<http://www.monografias.com/trabajos42/codigo-de-barras/codigo-de-barras4.shtml>

Tipus de codis de barres:

<http://codigodebarras.tecnoface.com/resumen/>

<http://www.slideshare.net/nozuan/codigos-bidimensionales-presentation>

Codis QR:

<http://www.techtear.com/2008/03/28/codigos-qr-¿que-son-¿por-que-existen-y-mas.html>

Evolució Tecnologia Mobil:

<http://www.eveliux.com/mx/el-abc-de-la-telefonía-celular-parte-1.php>

<http://www.eveliux.com/mx/la-evolucion-de-la-telefonía-movil.php>

Comparativa telefonía mòbil:

<http://156.35.151.9/~smi/5tm/09trabajos-sistemas/1/Memoria.pdf>

Evolució Sistemes Operatius Android:

<http://www.movilzona.es/2010/07/14/android-evolucion-del-sistema-operativo-desde-cupcake-a-gingerbread/>

Eclipse:

[http://es.wikipedia.org/wiki/Eclipse_\(software\)](http://es.wikipedia.org/wiki/Eclipse_(software))

JAVA:

www.java.com

XML:

http://es.wikipedia.org/wiki/Lenguaje_de_marcado

<http://www.informatica-hoy.com.ar/aprender-informatica/Que-es-el-lenguaje-XML.php>

KVM:

www.java.com/en/download/faq/what_kvm.xml

ZXing:

<http://www.qrcode.es/?p=324&language=es>

Instal·lació y Configuració Eclipse:

<http://developer.android.com/sdk/installing.html>

Estructura projecte Eclipse:

<http://www.weterede.com/2010/10/estructura-de-un-proyecto-android-en-eclipse/>

ADT Plug-in

<http://developer.android.com/sdk/eclipse-adt.html>

12. Annexos.

12.1.- Evolució històrica Sistemes Operatius Mòbils.



12.2.- Pressupost.

TecnoCampus
Mataró
Av. Ernest Lluch, 32
(Mataró) 08302
NIF: G-62034111

Pressupost 01/011

Pressupost, pel desenvolupament d'una aplicació per a telèfons amb OS Android, anomenat CallMemory, que esta destinada a facilitar la tasca de realització d'una trucada per a persones grans, que tenen una dis minució psiquica, en aquest cas Alzheimer.

	Preu unitari.	Unitats.	Preu Total
A.- Estudi i viabilitat del projecte			
"CallMemory".	15 €	180	2.700 €
B.- Material.			
Ordinador personal. Acer Aspire 4810 T.	350 €	1	350 €
Telèfon. HTC Wildfire Lliure	219 €	1	219 €
C.- Mà d'obra i desenvolupament del projecte.			
Hores Ingenier tècnic "Telecomunicacions"	40 €	200	8.000 €
<hr/> TOTAL PRESSUPOST			11.269 €

NOTA:

* No esta inclòs el 18 %, d'IVA.

12.3.- Codi final de l'aplicació.

Splash.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="@drawable/images"
    android:layout_gravity="center_vertical">
    <TextView android:layout_height="wrap_contént"
    android:text=""
    android:layout_width="fill_parent"
    android:id="@+id/name1"></TextView>

    <TextView android:id="@+id/Titulo"
    android:textColor="@color/negro"
    android:typeface="serif"
        android:textStyle="bold/italic"
    android:textSize="6mm"
    android:gravity="center"
    android:text="@string/app_name"
    android:layout_height="wrap_contént"
    android:layout_width="fill_parent"
    ></TextView>

    <TextView android:id = "@+id/name"
    android:layout_width="fill_parent"
    android:layout_height="6mm"
    />
    <TextView android:id="@+id/introduccio"
    android:layout_height="wrap_contént"
    android:gravity="center"
    android:typeface="serif"
        android:textStyle="bold/italic"
    android:text="@string/introduccio"
    android:layout_width="fill_parent"
    android:textColor="@color/negro"
    android:textSize="2.5mm" ></TextView>

    <TextView android:id="@+id/TextView01"
    android:layout_height="2mm"
    android:text="" android:layout_width="fill_parent"
    ></TextView>

    <ProgressBar android:layout_gravity="center"
    android:id="@+id/progressBar1"
    android:layout_height="wrap_contént"
    android:layout_width="wrap_contént"
    android:layout_marginRight="5dp"
    style="@android:style/Widget.ProgressBar.Large">
    </ProgressBar>
        <TextView android:id="@+id/textView1"
        android:layout_width="wrap_contént"
        android:text="@string/loading"

```

```

        android:layout_height="wrap_content"
        android:layout_gravity="center"></TextView>

```

```
</LinearLayout>
```

Main.xml.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    android:id="@+id/widget30"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="@drawable/images"
    android:orientation="vertical"
    xmlns:android="http://schemas.android.com/apk/res/android"
    >
    <TextView
        android:id="@+id/widget31"
        android:layout_width="fill_parent"
        android:layout_height="36px"
        android:text="Instruccions de funcionamnet..."
        android:textSize="18sp"
        android:typeface="serif"
        android:textStyle="bold/italic"
        android:textColor="@color/negro"
    >
    </TextView>
    <TextView
        android:id="@+id/widget32"
        android:layout_width="fill_parent"
        android:layout_height="20px"
        android:text="1.- Premi el boto &quot;Iniciar&quot;."
        android:typeface="serif"
        android:textStyle="bold/italic"
        android:textColor="@color/negro"
    >
    </TextView>
    <TextView
        android:id="@+id/widget36"
        android:layout_width="fill_parent"
        android:layout_height="24px"
        android:text="2.- Fixi el Codi al centre del rectangle..."
        android:typeface="serif"
        android:textStyle="bold/italic"
        android:textColor="@color/negro"
    >
    </TextView>
    <ImageView
        android:id="@+id/scanner"
        android:layout_width="100px"
        android:layout_height="80px"
        android:background="@drawable/scanner"
        android:layout_gravity="center_horizontal"
    >
    </ImageView>
    <TextView
        android:id="@+id/widget40"

```



```

android:layout_width="fill_parent"
android:layout_height="24px"
android:text="3.- Inicialitzem la trucada..."
android:typeface="serif"
android:textStyle="bold/italic"
android:textColor="@color/negro"
>
</TextView>
<TextView
android:id="@+id/widget43"
android:layout_width="fill_parent"
android:layout_height="28px"
android:text="* En cas d'apros;emergencia premi el boto Emergencies..."
android:typeface="serif"
android:textStyle="bold/italic"
android:textColor="@color/negro"
>
</TextView>
<AbsoluteLayout android:id="@+id/absoluteLayout1"
android:layout_height="fill_parent"
android:layout_width="fill_parent">
    <Button android:layout_height="wrap_contént"
android:layout_width="wrap_contént"
android:text="Iniciar"
android:id="@+id/iniciar"
android:layout_x="185dip"
android:layout_y="0dip"></Button>
    <Button android:layout_height="wrap_contént"
android:layout_width="wrap_contént"
android:text="Emergencies"
android:id="@+id/emergencias"
android:layout_x="24dip"
android:layout_y="2dip"></Button>
</AbsoluteLayout>

</LinearLayout>

```

SplashScreen.java

```

package com.testing.splashscreensample;

import android.app.Activity;
import android.contént.Intent;
import android.os.Bundle;

public class SplashScreen extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.splash);
        Thread splashThread = new Thread() {
            @Override
            public void run() {
                try {
                    int waited = 0;
                    while (waited < 5000) {
                        sleep(100);
                    }
                }
            }
        };
    }
}

```

```

        waited += 100;
    }
} catch (InterruptedException e) {
    // do nothing
} finally {
    finish();
    Intent i = new Intent();
    i.setClassName("com.testing.splashscreensample",

"com.testing.splashscreensample.mainmenu");
    startActivity(i);
}
}
};
splashThread.start();
}
}
}

```

MainMenu.java

```

package com.testing.splashscreensample;

import android.app.Activity;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class mainmenu extends Activity {
    private
    Button CallButton;
    Button Emergencies;
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        CallButton = (Button) findViewById(R.id.iniciar);
        CallButton.setOnClickListener(callm);
        Emergencies = (Button) findViewById(R.id.emergencias);
        Emergencies.setOnClickListener(calleme);
    }
    public Button.OnClickListener callm = new Button.OnClickListener()
    {
        public void onClick(View v) {
            Intent intent = new
Intent("com.google.zxing.client.android.SCAN");
            intent.putExtra("SCAN_MODE", "QR_CODE_MODE");
            startActivityForResult(intent, 0);
        }
    };
    public void onActivityResult(int requestCode, int resultCode,
Intent intent) {
        if (requestCode == 0) {
            if (resultCode == RESULT_OK) {

```

```
        String conténts =
intent.getStringExtra("SCAN_RESULT");
        String Resultado =
intent.getStringExtra("SCAN_RESULT_FORMAT");
        Call("647514133");
        // Handle successful scan
    } else if (resultCode == RESULT_CANCELED) {
        String error= "ERROR: No ha estat
possible, escanejar el codi";
        //Handle cancel;
    }
}
}
private void Call(String string) {
// TODOIntent
Intent callNumber = new Intent();
callNumber.setAction(android.contént.Intent.ACTION_CALL);
callNumber.setData(Uri.parse("tel:" + string));
startActivity(callNumber);
}
public Button.OnClickListener calleme = new
Button.OnClickListener() {
    public void onClick(View v) {
        Call("112");
    }
};
}
```