

Disseny d'una Aplicació Android



Data: gener 2013

ÍNDEX

1. Introducció	4
2. Elecció del sistema operatiu mòbil.....	6
3. Elecció del tipus de programació	6
4. App Inventor	7
4.1. Instal·lació	8
4.2. Funcionament Online	8
4.3. Component Designer	9
4.4. Blocks Editor	10
4.5. Compilació	11
5. Com es dissenya una app	12
5.1. Plantejar-se objectius	12
5.2. Idear el disseny de l'aplicació.....	12
5.3. Tria de components al <i>Component Designer</i>	13
5.4. Convencions	13
5.5. Programació de blocs.....	14
5.6. Depuració i cerca d'errors	14
5.7. Publicar a <i>Google Play</i>	15
6. Exemple d'app senzilla	16
7. Programació i manteniment de l'App <i>CineClub</i>	16
7.1. Plantejament	16
7.2. Pantalla Inicial	19
7.3. Pantalla Inici.....	24
7.4. Pantalla Pel·lícules	24
7.5. Pantalla Crèdits.....	26
7.6. Pantalla Registra't	27
7.7. Pantalla Valoracions.....	30

7.8. Pantalla Estadístiques.....	36
7.9. Manteniment de l'app.....	43
7.10. Visualització al mòbil.....	44
8. <i>Conclusions</i>	45
9. <i>Agraïments</i>	46
10. <i>Bibliografia</i>	47

Avís: Aquest treball conté paraules explícitament sense accentuar o amb símbols especials no admesos pel corrector. Cal tenir en compte que en l'ús de variables o etiquetes de programació s'han omès per tal del seu correcte funcionament.

1. Introducció

El principal objectiu d'aquest treball és poder dissenyar, programar i pujar¹al *Google Play* (lloc des d'on es descarreguen totes les aplicacions per el sistema operatiu *Android*) una aplicació destinada al **CineClub** organitzat per l'Institut La Bisbal.



Logotip CineClub

Quan vaig haver de decidir quin treball fer no vaig triar aquest tema instantàniament ja que tenia bastants idees en ment i totes em semblaven interessants. Tot i això, vaig fer un balanç de coses que em podien aportar cada una i vaig acabar escollint aquest. Vaig pensar que intentant crear una aplicació podria aprendre alguns llenguatges de programació tals com *Java* o altres que mencionaré més endavant. També hi havia la possibilitat d'idear un bon disseny per fer-la més atractiva i aquest apartat va ser un dels que més em va agradar fer degut a la gran quantitat de coses que es poden fer amb el programa *Photoshop*. Finalment, quan vaig adonar-me que la majoria d'adolescents, incloent-m'hi jo, estem tot el dia amb el mòbil, vaig veure que podria fer alguna cosa que pogués arribar a força gent i a més, que fos útil.

Un cop vaig haver triat el tema,el següent pas era escollir sobre què feia l'aplicació. Aquí el meu tutor em va donar la idea de fer-la sobre l'esdeveniment que es fa cada dos divendres al Cine Mundial de La Bisbal on es passen pel·lícules en VOS². De seguida em va semblar una bona idea ja que a mi m'agrada el cinema i a més vaig veure que podria servir d'ajuda per als que ho organitzen, ja que la manera que utilitzaven per comunicar a la gent que hi havia un passi de pel·lícula era poc eficient.

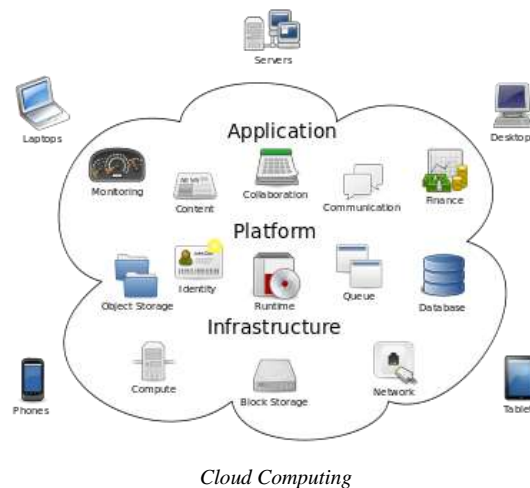
Finalment, vaig fixar aquests objectius per l'aplicació:

- Hauria d'avisar a la gent que la tingués que hi havia sessió de cine.
- Hi hauria d'haver un registre d'usuaris.
- Hi hauria d'haver una llista de totes les pel·lícules passades i per emetre.

¹Pujar un arxiu és enviar-lo des del teu ordinador a través de la xarxa d'Internet al servidor que té el projecte.

²Sigles de *Versió Original Subtitulada*.

- L'usuari registrat podria valorar les pel·lícules i a partir d'aquí generar unes estadístiques gràfiques.
- Intentaria pujar-la al *Google Play* per tal que tothom que volgués se la descarregués gratuïtament.
- Utilitzar alguns serveis de *Cloud Computing*³.
- El disseny hauria de ser atractiu, amè, divertit i fàcil d'entendre.
- Finalment, hi hauria un apartat amb la localització d'on es troba el CineClub, gent que ho organitza, etc.



³La informàtica en núvol (de l'anglès *Cloud Computing*) és una forma de computació que té els seus fonaments a Internet i que mitjançant la qual, els recursos compartits, programari i informació, es proporcionen a ordinadors i d'altres dispositius a la carta com a serveis.

2. Elecció del sistema operatiu mòbil

Una de les primeres coses que vaig haver de plantejar-me quan vaig idear el meu treball va ser el sistema operatiu de mòbil per al qual volia fer l'aplicació. Avui en dia n'hi ha bastants: *Symbian*, *BlackBerry*, *Windows Phone*... Tot i això, els que realment porten la batuta del mercat són l'*IOS*(emprat per a l'*Iphone*) i l'*Android*. Durant els últims anys les vendes de mòbils amb el sistema *Android* han augmentat radicalment de tal manera que han aconseguit superar *IOS*, que tenia el monopoli del mercat. Tal com indiquen les vendes, *Android* s'ha expandit arreu i aquest és el motiu pel qual gairebé el total d'alumnes de la meua classe i del meu curs tenen mòbils amb aquest sistema. Basant-me amb aquesta dada, vaig decidir fer servir el sistema operatiu *Android*. A més de ser popular, també té moltes altres avantatges:

- No té cap **restricció**. Altres sistemes estan dissenyats només per un tipus de mòbils, per tant, tenen una expansió limitada.
- Utilitza **codi lliure**: el programador pot modificar i millorar el sistema. Això també fa que, si hi ha algun error o *bug*, es pugui arreglar fàcilment per l'usuari ja que no hi ha cap barrera legal que ho impedeixi.
- Et dóna total **llibertat** per instal·lar qualsevol aplicació, sigui provinent del *Google Play* (lloc oficial de descàrrega) o bé qualsevol altre arxiu executable.



Icona Android

3. Elecció del tipus de programació



Java

Logotip Java

El llenguatge que utilitzen els programadors d'*Android* és el *Java*⁴. Al començament vaig pensar en aprendre a escriure codi *Java*. Vaig comprar un llibre que parlava del tema i em vaig posar a practicar amb el programa que recomanen per fer-ho, l'*Eclipse*. Al cap de poc temps vaig veure que era força complicat entendre totes aquelles instruccions en tant poc temps la qual cosa

⁴El *Java* és un llenguatge de programació dissenyat el 1990 per *Sun Microsystems* a partir de *C++*. Des del seu naixement fou pensat com un llenguatge orientat a objectes.

suposaria per si mateixa un treball de recerca. Havia fet alguna aplicació simple però que no tenia res a veure en qüestió de dificultat amb el que volia fer jo. Així doncs vaig deixar-ho estar i vaig buscar altres solucions.

Durant dues setmanes del mes de juliol vaig assistir al campus que es feia a la *Universitat de Girona*. Allà em van assignar dos tutors que eren especialistes en el tema i que em van ajudar a aclarir alguns dubtes que tenia sobre el meu projecte. Va ser allà on em van aconsellar que utilitzés un programa anomenat *AppInventor*.



Logotip App Inventor

4. AppInventor

AppInventor es tracta d'un servei *Online* que permet a l'usuari crear aplicacions pel sistema *Android*. El seu principal avantatge és que és simple i no fa falta tenir una gran base de programació ja que utilitza un sistema similar a l'*Scratch*⁵.

L'avantatge que jo tenia era que dos cursos anteriors havia treballat molt amb l'*Scratch* i això em donava moltes facilitats a l'hora de fer servir l'*AppInventor*. També té la possibilitat de treballar simultàniament connectat a un *smartphone*⁶ on es pot veure el que estàs fent a la web. L'únic defecte que vaig trobar a aquest programa van ser alguns errors que té ja que es troba en fase *beta*⁷.

Consta bàsicament de tres parts ben diferenciades:

- el *component designer*: on afegim els components de l'aplicació(o disseny)
- el *blocks editor* on construïm el comportament dels components afegits (o programació)

⁵És una aplicació informàtica destinada principalment als nens, mitjançant la qual poden explorar, experimentar i manifestar idees de manera creativa i divertida. És desenvolupat pel grup *MIT Media Lab*. Es pot instal·lar i distribuir lliurement en qualsevol ordinador amb *Windows*, *MacOSX* o *Linux*.

⁶Un telèfon intel·ligent és un ordinador de butxaca que integra les funcions de telèfon mòbil, organitzador personal i sovint altres funcions de connectivitat mòbil.

⁷Un beta, en argot informàtic, és un programari en els que en alguns aspectes del seu desenvolupament estan pendents d'acabar de dissenyar-se i comprovar-ne el seu funcionament òptim, però que encara no són completament estables.

- l'emulador⁸ o dispositiu mòbil: on veiem el resultat final



Pantalla del Component Designer, Blocks Editor i Emulador

4.1. Instal·lació

Primer de tot cal anar a l'adreça <http://AppInventor.mit.edu/> i clicar el botó “*invent*”. Tot i no treballar directament amb la sintaxi de *Java*, el nostre ordinador necessita tenir-lo instal·lat. El mateix *AppInventor* té un enllaç (*Learn*) on pots fer una prova per saber si el tens instal·lat amb la versió adient i que funcioni correctament.

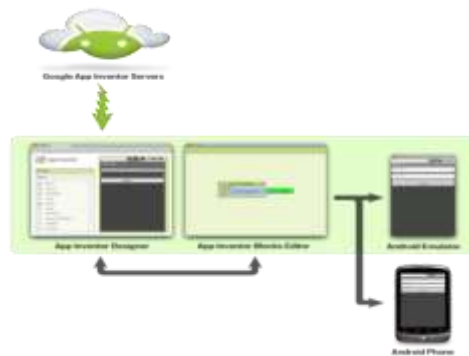
Seguidament s'ha d'instal·lar el software addicional de l'*AppInventor* (versió *Windows* o *Mac*) que et proporciona la mateixa pàgina. Seguint els passos que t'indiquen ja tindràs el programari operatiu. Ara hauràs d'associar-hi el teu compte *gmail* per poder accedir finalment al creador d'aplicacions. S'utilitza aquest servei ja que abans que l'agafés *MIT*⁹, aquest programa era de *Google* i feia servir aquest correu per registrar-se.

4.2. Funcionament Online

El programa *AppInventor* funciona *Online* (en línia, és a dir, necessita connexió a Internet). Aquest tipus de funcionament té pros i contres. Els pros serien que pots treballar-hi des de qualsevol ordinador i les contres, la ralentització del procés. A la pantalla d'inici es pot crear un nou projecte (*New*) o gestionar els existents. Tots aquests projectes es carreguen quan accedeixes amb un compte *gmail* al programari. En el moment de la redacció d'aquest treball estem a la versió: *Built: December 12 2012 Version: v132*

⁸Un emulador és un programari que permet executar programes d'ordinador en una plataforma diferent de aquella per la qual van ser escrits originalment.

⁹El *Massachusetts Institute of Technology (MIT)* és una institució i universitat estatunidenca dedicada a la recerca i la investigació, situada a la ciutat de *Cambridge, Massachusetts*. Es dedica sobretot a la tecnologia, ciència i enginyeria, però també a altres temes.



Esquema d'interconnexió App Inventor – Web



Pantalla d'inici

4.3. Component Designer

És la part del programa on es mostren els elements que es poden afegir a la aplicació en una pantalla (*Viewer*) que simula la que tindrà la nostra aplicació. A l'esquerra hi ha els components disponibles dividits en diferents apartats depenent de la seva funció:

- A *Basic* hi podem trobar components essencials per a la nostra aplicació com la introducció d'una caixa per escriure-hi un text, posar una imatge o un botó que faci alguna funció.
- Dins *Media* trobem tot tipus de components que ens permetran afegir elements com vídeos o sons per tal de fer l'aplicació més multimèdia.
- *Animation*, tal com el seu nom indica, ofereix components per crear animacions.
- L'apartat *Social* conté diferents elements que interactuen amb la gent, per exemple accedir a *Twitter*.
- A la part de *Sensors* hi ha els elements que interactuen amb els sensors que té el dispositiu, per exemple, el giroscopi i el sistema de geolocalització.
- *Screen Arrangement* conté tres components que estableixen un quadre horitzontal, vertical o una taula per organitzar tots els components de la aplicació.
- *Lego Mindstorms* utilitza funcions molt específiques per robots *Lego*.
- Dins *Other Stuff*, elements per enviar notificacions, per utilitzar el *Bluetooth*, etc.

- Per últim, a *NotReady For Prime...*, elements nous com les *FusionTables* i d'altres.

A la dreta hi ha la finestra amb la llista dels components afegits (també inclou els arxius pujats a l'apartat “*Media*”) així com la finestra de les seves propietats (*Properties*).

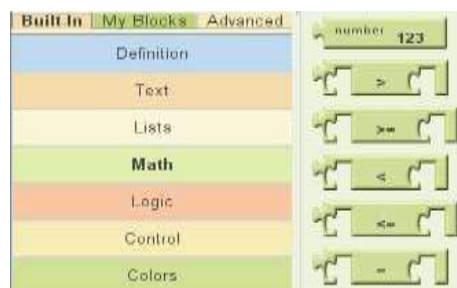


Pantalla Components Designer

4.4. Blocks Editor

Un cop han estat triats els components que volem utilitzar i arrossegats a la pantalla de disseny, podrem obrir el *blocks editor* per establir el comportament de cada element. Cada cop que el vulguem utilitzar haurem de descarregar una aplicació *java* clicant al botó “*Open the Blocks Editor*”. Aquest subprograma fa servir blocs que s'uneixen entre ells a manera de puzzle i cada un té una funcionalitat diferent. A l'esquerra hi ha una columna amb tres pestanyes: *Built-in*, *My blocks* i *Advanced*.

- *Built-in*: és on hi ha les ordres que seguiran els components que hagi posat. Estan agrupades per tipus: text, matemàtic, lògic, etc.



Pantalla blocks editor amb la pestanya Built-In

- *My blocks*: aquí es mostren els components que hem afegit al *component designer*. Quan cliquem un component d'aquests es veurà les funcions que poden exercir depenent del tipus de component que haguem triat.



Pantalla blocks editor amb la pestanya My Blocks

- *Advanced*: aquest apartat no l'he utilitzat ja que té unes funcions bastant complexes i que a mi no em calien.



Pantalla blocks editor amb la pestanya Advanced

4.5. Compilació

Un cop sabem al cent per cent que el nostre codi funciona bé, podem procedir a compilar-lo¹⁰ (format de fitxer *APK*) per instal·lar-lo al mòbil. Per fer-ho haurem d'anar a l'apartat del *component designer* on hi posa "*Package for phone*". Hi ha tres opcions:

- *Show Barcode*: aquesta opció crea automàticament un codi QR¹¹ el qual, si l'escanegem amb el mòbil, ens descarregarà la aplicació.
- *Download to this computer*: aquesta opció et permet descarregar l'aplicació a l'ordinador on estàs treballant en format de fitxer *ZIP*.



Opcions de compilació

¹⁰ Tradueix un llenguatge informàtic a un altre entenedor pel dispositiu.

¹¹ El *Quick Response Code*, codi de resposta ràpida, és un sistema per emmagatzemar informació en una matriu de punts o un codi de barres bidimensional creat per la companyia japonesa *Denso Wave* al 1994.

- *Download to connected phone:* aquesta opció descarrega i instal·la directament l'aplicació a qualsevol dispositiu mòbil que tinguem connectat a l'ordinador. És l'opció més ràpida i senzilla ja que no t'has de preocupar de fer res.

Un cop triada l'operació s'ha d'esperar que es compili (*Packaging*) i descarregui la aplicació. Aquest pot ser un procés llarg i, òbviament, com més codi tinguem més col·lapses estiguin els servidors que allotgen els nostres projectes, més trigarà.

5. Com es dissenya una app

El procés de creació d'una aplicació amb *App Inventor*, sigui per a mòbil o per a qualsevol altre dispositiu, és llarg, es necessita seguir una sèrie de passos i ser metòdic ja que l'error més petit pot comportar llargues hores de treball per identificar-lo. Abans de començar amb l'*App Inventor* hem de plantejar-nos què volem fer i estructurar-nos la feina seguint els passos següents:

5.1. Plantejar-se objectius

És bàsic establir uns propòsits clars a aconseguir per poder estar motivat i per saber fins on has d'arribar.

5.2. Idear el disseny de l'aplicació

Un cop tenim els objectius fixats, hem de fer un esbós de com ens agradaria que fos la nostra aplicació. Tot i que el disseny es deixa pel final durant la programació, està bé tenir una idea de com volem que sigui.

Quan comencem a programar utilitzarem botons amb imatges bàsiques i no ens centrarem en l'estètica ja que és primordial sobreposar el bon funcionament de l'aplicació davant la resta de coses. Recomano crear uns botons simples per començar. La web *cooltext.com* et dóna la possibilitat de fer aquestes icones pels botons de l'aplicació.

5.3. Tria de components al *Component Designer*

Quan afegiu un component a l'aplicació, com hem dit, a la dreta sortiran les seves propietats que podrem modificar-se. Depenent del tipus escollit es podrà modificar d'una manera o altra, per exemple en un component *Label* podràs definir la mida, la lletra, el color... En el cas especial d'un component *image*, primer hauríem de pujar la foto desitjada sota els components on hi diu “*Media*”.

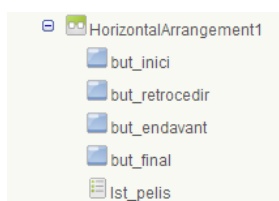


Propietats d'un Component



Llista d'imatges pujades

5.4. Convencions



Exemple convencions

Aquest punt, tot i semblar poc important, és un dels que més cal tenir en compte. Cada element que incorporem a la aplicació a l'apartat de *component designer* se li assigna un nom predeterminat per el programa. És gairebé obligatori canviar aquests noms per uns amb els quals puguem identificar quin tipus de component són ja que quan anem al *blocks editor* podríem tenir dubtes i equivocar-nos amb l'elecció del component. En el meu cas, he triat una abreviació on primer escric el tipus de component abreviat i seguidament la funció o utilitat que vull que faci. Per exemple: “but_endavant”, “but” es refereix a botó (*button*) i “endavant” la funció que ha de fer. Es recomana l'ús del símbol “_” per separar paraules i no fer servir accents ni caràcters especials.

5.5. Programació de blocs

Aquí arribem a la part més complicada de tot el procés de creació d'una aplicació: la programació de blocs. Aquí, tal com deia a l'apartat de *component designer*, no puc dir què s'ha de fer específicament ja que el que s'ha de programar és diferent en cada cas depenent de l'aplicació que es vol fer. El que puc donar són alguns consells per al correcte funcionament i per disminuir errors.

- La bona organització dels blocs et pot ajudar molt ja que a vegades en pot quedar algun bloc amagat darrere un altre fent que et sigui impossible de detectar.
- Si en algun cas es defineix una variable, s'ha de procurar recordar el nom que li hem donat i, quan l'utilitzem, s'ha d'escriure exactament com està definit.

Les estructures bàsiques de programació són:

- Ús de variables i llistes.
- Ús de blocs condicionals perquè l'aplicació agafi decisions.
- Ús de bucles per a processos repetitius.
- Ús de procediments per reutilitzar codi.
- Ús de APIs¹² per interaccionar amb altres serveis.

5.6. Depuració i cerca d'errors

Un cop haguem acabat la part de programació de la nostra aplicació, haurem de buscar possibles errors que puguin sorgir quan instal·lem la aplicació al mòbil. Un dels avantatges d'aquest programa és que pots treballar amb un emuladoro directament connectat al mòbil fent que es pugui veure directament al teu dispositiu cada canvi que apliquis. Quan s'utilitza aquests tipus de programes hi ha molts elements a tenir en compte, la qual cosa que fa que ens puguem equivocar fàcilment. En aquest cas, utilitzant aquest sistema de programació a base de connexió de blocs no podem tenir errades de sintaxi (manera que s'escriuen les ordres) ja que els blocs que utilitzem ja venen programats de sèrie i, per tant, és impossible que ens equivoquem. Per altra banda, un dels errors més freqüents és equivocar-se de variable o deixar espais en blanc dins els blocs. La millor manera de no cometre errors és intentar repassar cada acció que fem ja que d'altra manera anirem

¹²Una Interfície de Programació d'Aplicacions (*Application Programming Interface*) és un conjunt de declaracions que defineix el contracte d'un component informàtic amb qui farà ús dels seus serveis.

arrossegant els errors i pot ser gairebé impossible de detectar on hem fallat. Per comprovar el correcte funcionament de la aplicació, *blocks editor* ens dona dues opcions: execució al mòbil connectat o simulant-la a un emulador. Si volem utilitzar aquesta última opció, haurem de crear-ne un on hi diu “*New Emulator*”; en canvi si volem fer servir el mòbil, haurem de connectar-lo per USB¹³ a l'ordinador i anar a “*Connect to device*” i seleccionar-lo. A mi, personalment, em va millor utilitzar aquest mètode ja que sé segur que l'aplicació quedarà de la manera que es veu a la pantalla del dispositiu.

La cerca d'errors pot ser una tasca molt llarga i frustrant degut a que moltes vegades necessites una llarga estona per veure on t'has equivocat. En casos on no trobes on hi pot haver l'errada, el millor que es pot fer és deixar-ho per una estona, practicar algun esport, descansar...i tornar-hi al cap d'unes hores.

5.7. Publicar a Google Play

Google Play és la botiga oficial *Online* d'*Android* on es descarreguen la majoria d'aplicacions per a aquest sistema. Aquesta part en el programador d'una aplicació és opcional. Un dels avantatges que et dona *Android* és que deixa a tothom penjar aplicacions a la seva botiga *Online*. Per fer-ho es necessiten alguns requisits:

- El fitxer no pot pesar més de **50MB**.
- Has de fer almenys dues captures de pantalla de l'aplicació.
- És obligatori tenir una icona d'alta resolució.
- Ha de tenir un títol i una descripció de com a màxim 4000 caràcters per a la gent que la vulgui descarregar.
- És necessari un compte de programador.

Aquest últim pas es fa a aquest lloc web: <https://play.google.com/apps/publish/signup>. Per crear-ne un, s'ha de pagar 25\$, un preu bastant raonable si penses que la aplicació serà allà per sempre. Un cop fet, estaràs al *Developer Console* i hauràs de clicar a “*Upload Applications*” per continuar a pujar la teva aplicació. Seguidament et demanaran tot aquest seguit de normes esmentades anteriorment.

¹³L'*Universal Serial Bus* (Bus Universal en Sèrie) o abreujat comunament USB, és un port que serveix per connectar perifèrics a un ordinador. Va ser creat el 1996 principalment per IBM, Intel i d'altres.

6. Exemple d'app senzilla

Aquesta aplicació ben senzilla consisteix en un gat que reproduïx una miolada quan el toques. Per fer aquesta aplicació arrossegarem els següents components al *component designer* (pantalla de disseny) i pujarem els fitxers de so i imatge indicats:

<i>Component</i>	<i>Paleta</i>	<i>Nom</i>	<i>Finalitat</i>	<i>Propietat</i>	<i>Valor</i>
<i>Label</i>	Basic	lbl_missatge	Mostrar un missatge	Text	Toca el gat
<i>Button</i>	Basic	but_toca	Activar el so	Image	<i>kitty.png</i>
<i>Sound</i>	Media	so	Reproduir un so	Source	<i>meow.mp3</i>



Pantalles del Component Designer

El comportament el definirem al *Blocks Editor*:

- Dins *My Blockstriem* el component "but_toca" i dins la llista de blocs arrosseguem el bloc "When but_toca.Click – Do"
- Dins *My Blocks* triem el component "so" i arrosseguem el bloc "Call so.Play" dins l'anterior.



Blocs botó toca

L'aplicació ja està preparada per compilar i ser descarregada al mòbil o bé a l'emulador.

7. Programació i manteniment de l'App CineClub

7.1. Plantejament

Aquí comença el que es consideraria la part principal del meu treball de recerca. Com he comentat a la introducció, l'objectiu d'aquest treball és aconseguir fer una aplicació

mòbil que expliqui què és el CineClub de La Bisbal d'Empordà, quines pel·lícules fan, on les fan i els dies que s'emeten a més d'altres opcions.

A l'utilitzar multi pantalles això em suposarà una pèrdua de temps considerable ja que l'emulació no pot treballar amb aquest sistema i cal descarregar el programa complet al dispositiu. Des del *component designer* cal crear les següents pantalles (*New Screen*):

- Pantalla **Inicial** (*Screen1*) on hi han les diferents icones per accedir a tots els apartats i un avís que et diuquina és la pròxima sessió.
- Pantalla **Inici** on hi hauria informació bàsica sobre la aplicació.
- Pantalla **Pel·lícules** on hi han totes les pel·lícules que s'han emès i s'emetràn al CineClub i les seves característiques (sinopsi, actors, director, tràiler...)
- Pantalla **Valoracions** on els usuaris registrats poden votar uns ítems d'una sessió.
- Pantalla **Crèdits** on hi ha informació sobre el CineClub: ubicació, organitzadors...
- Pantalla **Registra't** on la gent es registrar per poder fer votacions.
- I pantalla **Estadística** on es poden visualitzar el resultat gràfic de les votacions.

Ha estat necessari també la utilització dels següents serveis:

- *Google FusionTables*¹⁴:
 - **pelis**: base de dades que conté la totalitat de les pel·lícules amb el seu número de sessió, data, títol i enllaç *IMDb*¹⁵.



ID	Data	Títol	Link
1	22/01/2010	My Blueberry Nights	http://www.imdb.com/title/tt10295120/
2	05/03/2010	Jurt a Rise	http://www.imdb.com/title/tt10265470/
3	12/03/2010	Pink Floyd The Wall	http://www.imdb.com/title/tt0094150/
4	19/03/2010	Pani Je T'Aine	http://www.imdb.com/title/tt0401711/

Fragment de la FusionTable: pelis

¹⁴ *Google Fusion Tables* és una eina gratuïta per generar, compartir i publicar informació. Permet crear les nostres pròpies taules i representar les nostres dades, així com accedir a taules de caràcter públic.

¹⁵ *Internet Movie Database* (base de dades de pel·lícules a Internet) és una base de dades en línia amb informació sobre actors, pel·lícules, programes de televisió i videojocs.



Web IMDb

- **valors_atzar**: base de dades que conté les valoracions (de 1 a 5) de cada usuari de la sessió segons: guió, actors, música i efectes.
- **TinyWebDB**:
 - **usuaris_prova**: base de dades dels usuaris registrats (el seu número de mòbil). La creació d'aquestes bases de dades web s'explica a un enllaç de la bibliografia.
- **TinyDB**: és la base de dades interna de cada mòbil on es guarden una sèrie de variables permanents que s'utilitzen durant la programació.
- **Chart Tools de Google**: web que permet la creació de gràfics personalitzats a través d'una adreça web.

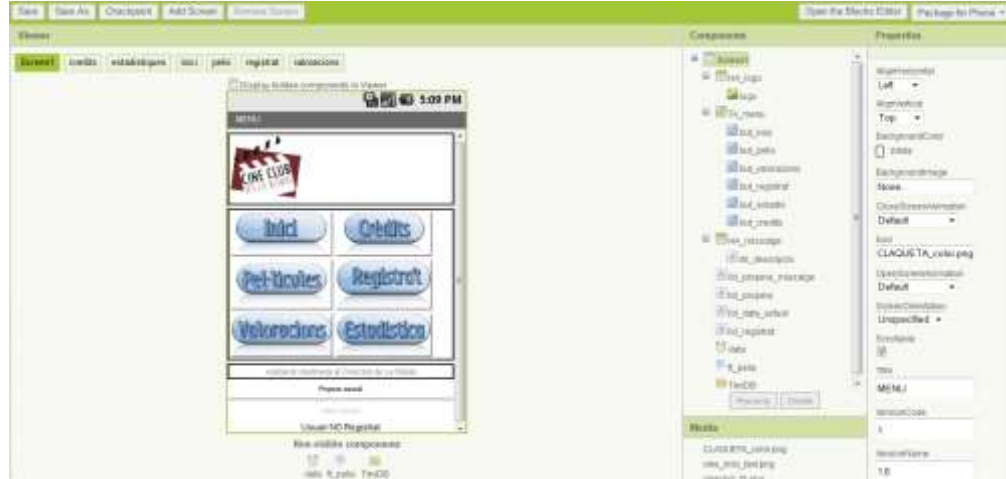


Possibilitats de gràfics amb Google Charts

7.2. Pantalla inicial

Aquesta és la pàgina amb la que s'obre l'aplicació. Conté informació i el menú de botons que permeten la navegació per la mateixa.

Component designer



Pantalla Inicial al Component Designer

- Icona del CineClub: he cregut necessari en benefici de la qualitat vectoritzar¹⁶ la icona original en mapa de bits amb el programari *Inkscape* per tal de poder-la redimensionar en les millors condicions. Per poder-la centrar cal posar-la dins un component *HorizontalArrangement*.

- Botons de navegació: per poder accedir als diferents apartats que ha de tenir el menú, cal



Web CoolText

posar sis *buttons* que ens permetin canviar de pantalla organitzant-los en una *TableArrangement* de 3x3. Activant “*Fill parent*” a la propietat “*Width*” de cada botó, farem que s'ajustin a l'amplada de la pantalla o arranjament.

Aquesta serà la opció per defecte per a la resta de programari. Per crear les icones dels botons he usat la web <http://cooltext.com/> la qual et permet dissenyar logos entre d'altres coses i descarregar-los a l'ordinador. Un cop creades cal pujar-les al programa amb l'opció “*Media*” sota

la llista de components on hi ha “*Upload new*”. Després modifiquem la propietat “*image*” de cada botó amb la icona corresponent. És recomanable treballar amb el format *PNG*¹⁷ que permet transparències a diferència del *JPEG*¹⁸.

¹⁶Convertir una imatge de mapa de bits en un gràfic vectorial.

¹⁷PNG: acrònim de *Portable Network Graphics*. Tipus de format d'imatge pensat especialment per a pàgines web amb la intenció de substituir el format GIF.

- Arrossega un *TinyDB* que ens serveix per recuperar variables importants.
- Arrossega un *FusionTable* `ft_pelis` per crear les llistes.
 - Caldrà adquirir un *API Key* (consultar a l'enllaç de la bibliografia) per emplenar aquesta propietat del component en tota la resta de la aplicació.
 - També caldrà l'identificador que té cada *FusionTable* que s'obté a *File>About This Table*. Aquesta haurà de ser pública per permetre l'accés als usuaris i compartida amb l'encarregat del manteniment de la mateixa.
- En tota aplicació va bé que a la pàgina principal hi hagi una breu descripció perquè l'usuari tingui una idea superficial de què tracta la aplicació. Així doncs, he posat un component *Label* que ho mostra.

Blocks editor

1. Defineix variables i llistes.
2. Arrossega el bloc “*When Screen1.Initialize-Do*”:
 - a. Mostra la data actual a través del component *clock*.
 - b. Crea una consulta (*Query*) de totes les columnes “*” de la *FusionTable* `ft_pelis` ordenades per data.
 - c. Executa la consulta (*DoQuery*).
3. Arrossega el bloc “*When ft_pelis.GotResult-Do*”:
 - a. Crea una llista del resultat de la consulta.
 - b. Guarda en una variable el total de registres.
 - c. Fes un bucle amb tots els registres per tal de crear quatre llistes: *pelis* (que ho conté tot), *pelis_titols* (només títols), *pelis_dates* (només dates) i *id_sessions* (només no. de sessions).
 - d. Emmagatzema a la *TinyDB* les llistes anteriors amb les etiquetes: *pelis*, *títols*, *dates* i *id* que necessitem per recuperar els valors des d'una altra pantalla.
 - e. Estableix un condicional que mira si l'usuari està registrat (*mòbil* ≠ *espai*), si ho està, mostra un missatge indicant-ho.
 - f. Crida el procediment *prop_sessió*.
4. Arrossega el bloc “*When but_inici.Click-Do*”:
 - a. Obre la pantalla *inici*.
5. Repetir el procés amb la resta de botons.

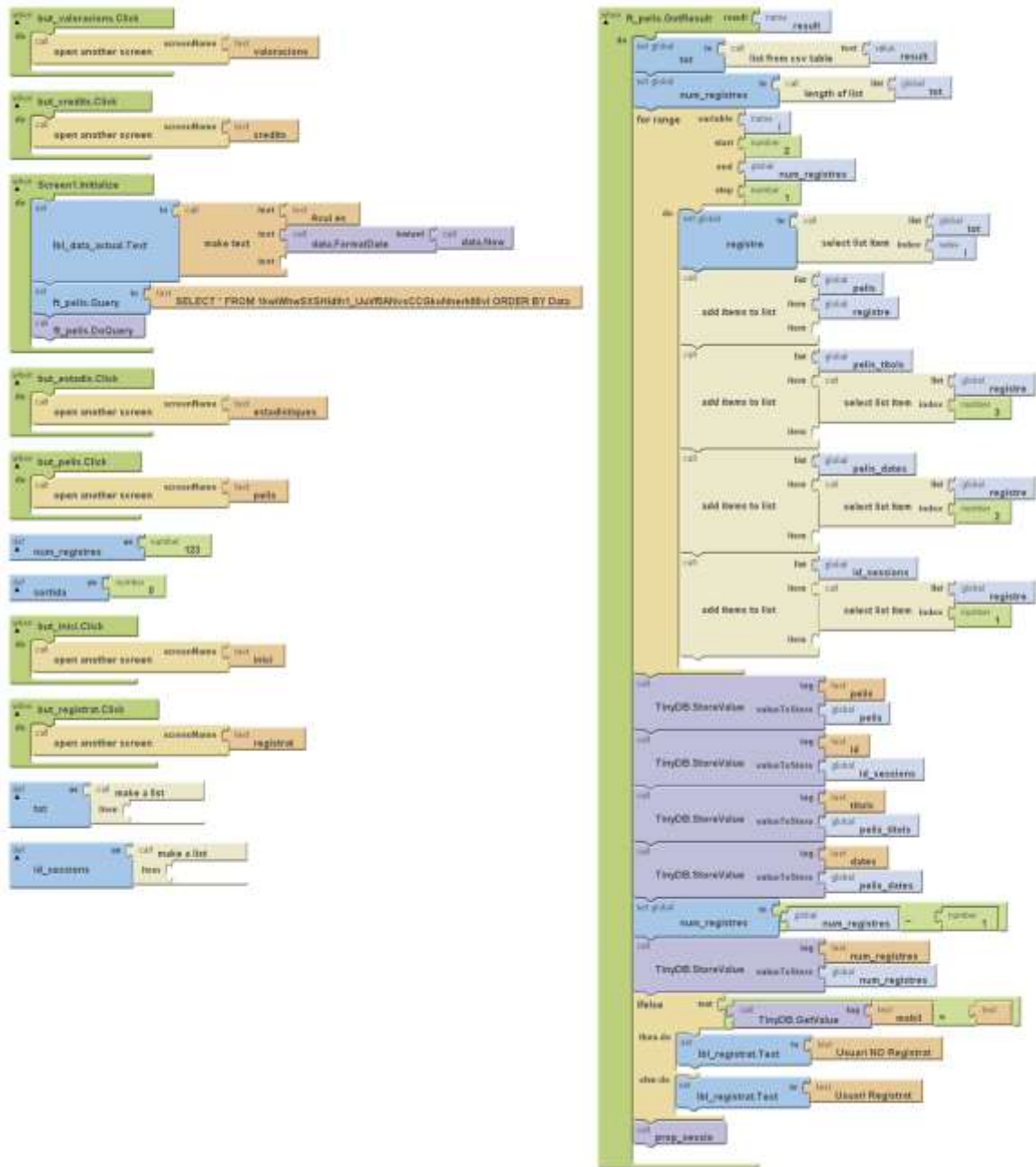
¹⁸JPEG: acrònim de *Joint Photographic Experts Group*. És un algorisme dissenyat per a comprimir imatges estacionàries amb 24 bits de profunditat o en escala de grisos.

6. Arrossega el bloc “*When Screen1.BackPressed-Do*”:
 - a. Si retorna a la pantalla inicial, estableix un condicional que mira si l'usuari s'ha registrat (mòbil \neq espai) i, si ho està, mostra un missatge indicant-ho.
7. Definir el procediment `prop_sessio` arrossegant el bloc “*To.Procedure-Arg-Do*”:

Cal aclarir que, com que les variables són del tipus text, la cerca de la sessió propera a la data actual ha estat més complexa.

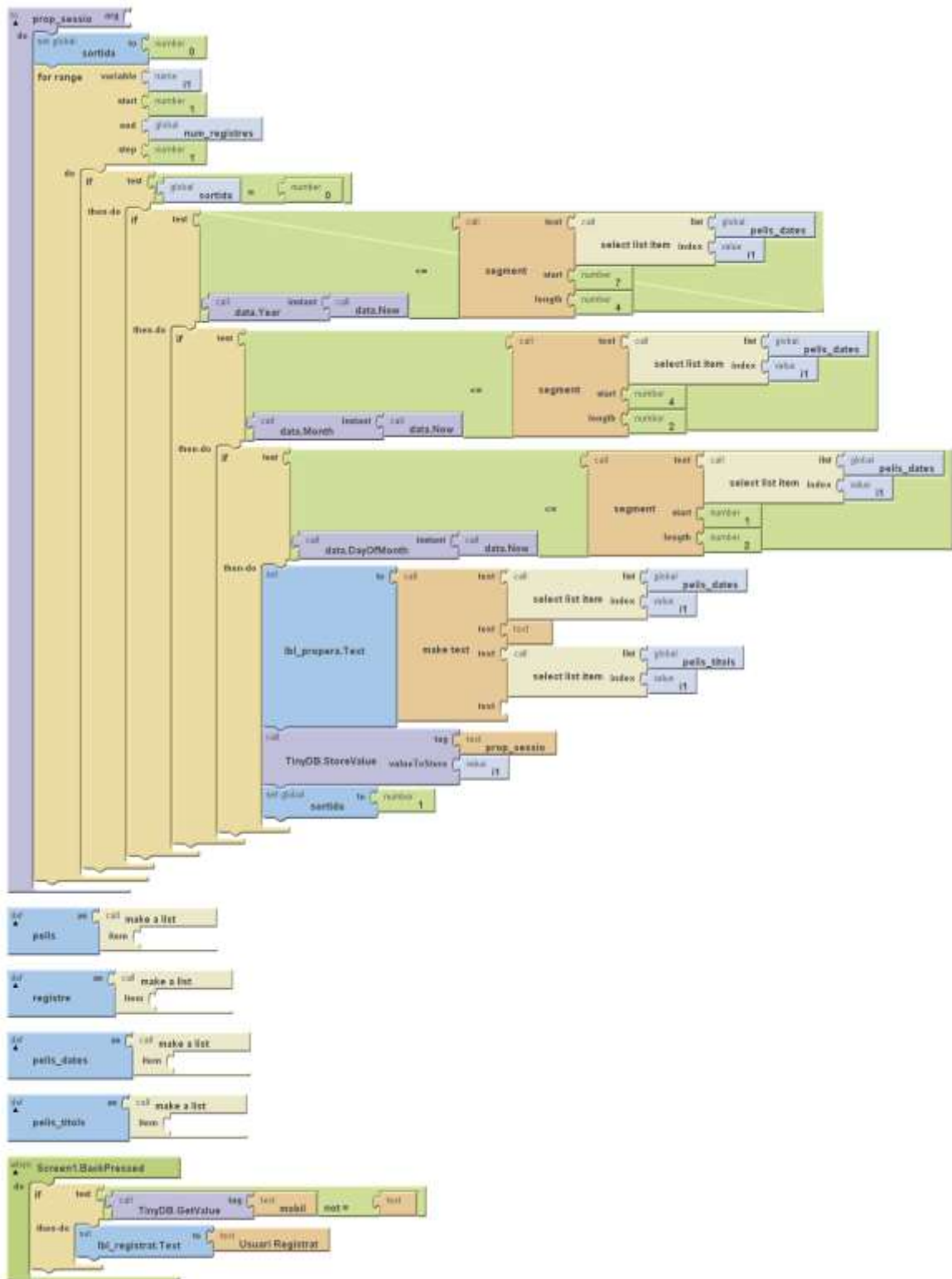
 - a. Posa la senyal de sortida a 0.
 - b. Crea un bucle per a tots els registres de la llista `pelis_dates` on es compara consecutivament l'any, mes i dia de la data seleccionada si és menor o igual als valors corresponents a l'actual. Si es compleix, es guarda la posició de la llista d'aquest valor dins la *TinyDB* amb l'etiqueta `prop_sessio`.
 - c. Si ha trobat un valor de `prop_sessio`, posa la senyal de sortida a 1.

Disseny d'una aplicació Android



Codi blocs pantalla Screen1 part 1ª

Disseny d'una aplicació Android



Codi blocs pantalla Screen1 part 2ª

7.3. Pantalla Inici

Aquesta és la pàgina més senzilla de l'aplicació. Conté una breu introducció de què és el CineClub de la Bisbal.

Component designer



Pantalla Inici

Blocks Editor: aquesta pantalla no té cap programació.

7.4. Pantalla Pel·lícules

Aquesta part de la aplicació està destinada a fer que l'usuari pugui veure totes les característiques de les pel·lícules que ofereix el CineClub.

Component designer



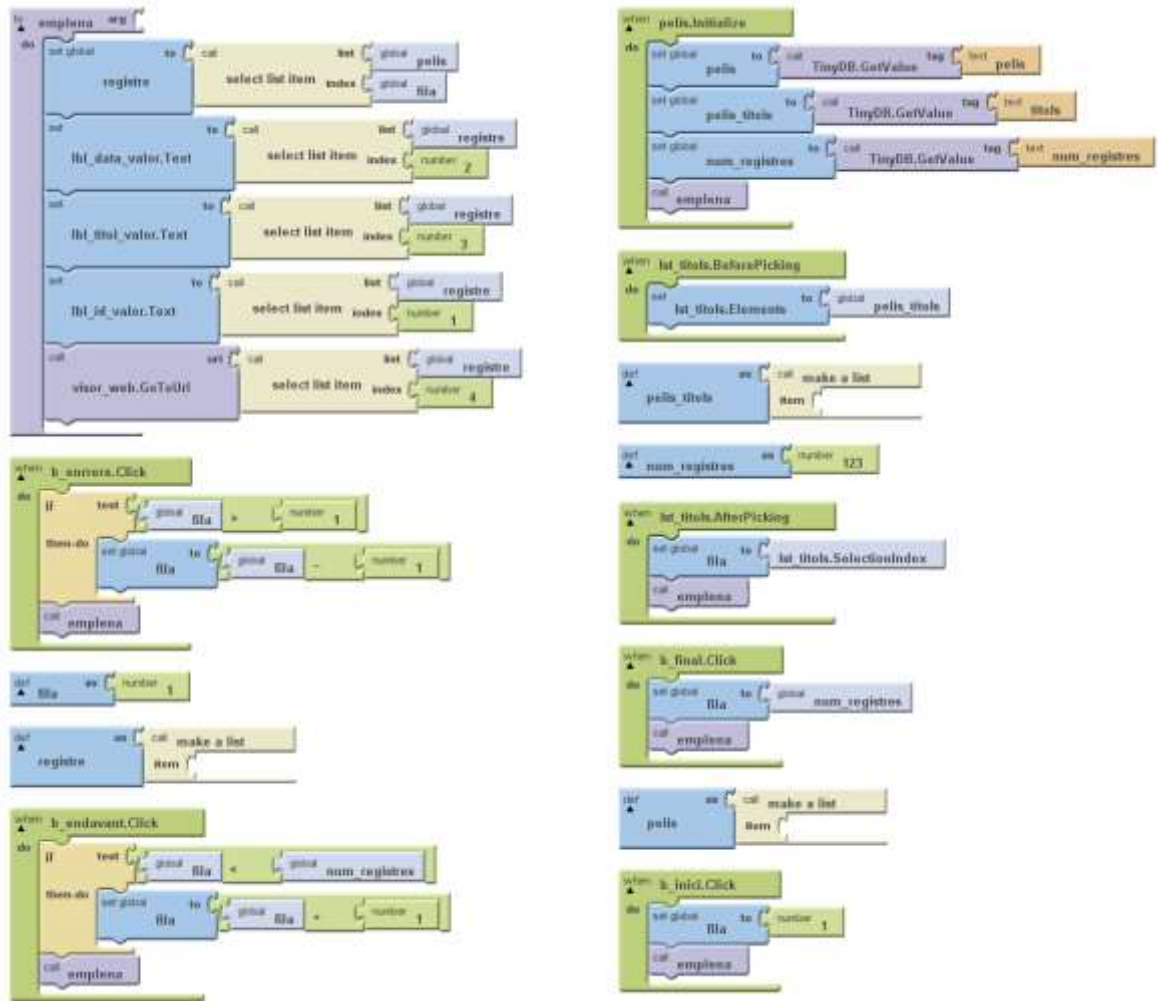
Pantalla Pelis

- Arrossega un *HoriztonalArrangement* amb tres *Labels*: Data, Títol i ID.

- Arrossega un *HoritzontalArrangement* per mostrar els valors dels anteriors.
- Arrossega un *HoritzontalArrangement* i insereix quatre *buttons* de navegació (<<, <, >, >>) i un *ListPicker* per accedir a la llista de pel·lícules.
- Arrossega un *WebViewer* per mostrar la web *IMDb* seleccionada.
- Arrossega un *TinyDB* que ens serveix per recuperar variables importants.

Blocks editor

1. Defineix variables i llistes.
2. Arrossega el bloc “*When pelis.Initialize-Do*”:
 - a. Recupera les llistes pelis, títols i num_registres de *TinyDB*.
 - b. Es crida el procediment emplena.
3. Definir el procediment emplena arrossegant el bloc “*To.Procedure-Arg-Do*”:
 - a. Recupera el títol, la sessió, la data i l'enllaç del registre seleccionat.
 - b. Mostra l'enllaç a través del bloc “*visor_web.GoToURL-Do*”.
4. Arrossega el bloc “*When but_inici.Click-Do*”:
 - a. Activa la posició 1 i cridem el procediment emplena.
5. Arrossega el bloc “*When but_final.Click-Do*”:
 - a. Activa l'última posició (num_registres) i crida el procediment emplena.
6. Arrossega el bloc “*When but_endavant.Click-Do*”:
 - a. Estableix un condicional per saber si estem al final. En cas negatiu, es resta una unitat a la posició actual i crida el procediment emplena.
7. Arrossega el bloc “*When but_enrere.Click-Do*”:
 - a. Estableix un condicional per saber si s'està a l'inici. En cas negatiu, se suma una unitat a la posició actual i crida el procediment emplena.
8. Arrossega el bloc “*When llista_titols.BeforePicking-Do*”:
 - a. Estableix la llista pelis_titols com a elements de la llista de selecció.
9. Arrossega el bloc “*When llista_titols.AfterPicking-Do*”:
 - a. Guarda la posició de la pel·lícula seleccionada i crida el procediment emplena.



Codi blocs pantalla pelis

7.5. Pantalla Crèdits

Aquest apartat conté informació addicional sobre el CineClub.

Component designer



Pantalla credits

- Arrossega cinc *HoritzontalArrangement* amb cinc *Labels*: Coordinació, Impulsor, Disseny App, Versió i Localització. Al costat dels quatre primers, quatre *Labelsmés* amb els valors corresponents.
- Arrossega un botó *Imatge* on sortirà la imatge indicant la localització del CineClub capturada del *Google Maps*.

Blocks Editor: aquesta pantalla no té cap programació.

7.6. Pantalla Registra't

L'apartat "Registra't" permet a l'usuari de l'aplicació registrar-se mitjançant el seu número de mòbil. Això permetrà que aquest accedeixi a les votacions de les pel·lícules.

Component designer

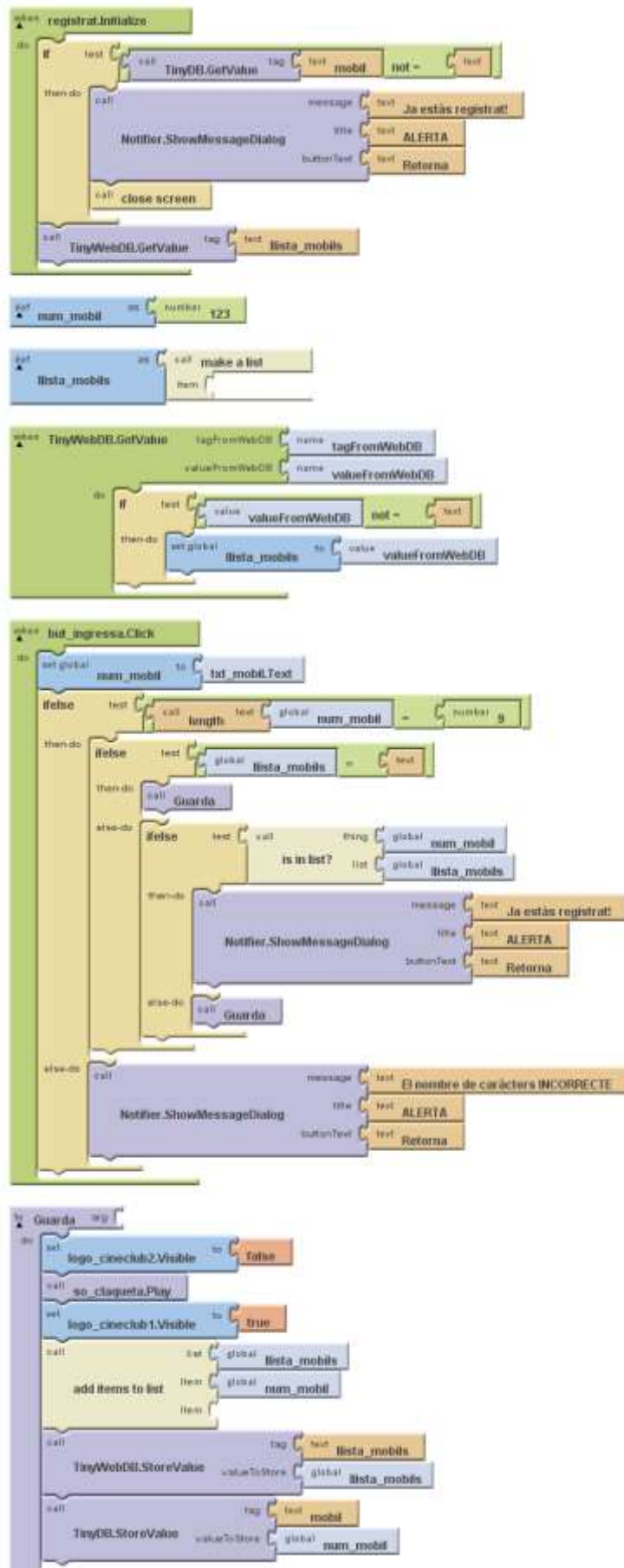


Pantalla registrat

- Arrossega un *HoritzontalArrangement* amb una *Label* que mostri "num_mobil".
- Arrossega un *TextBox* que servirà per introduir el número de telèfon de cada usuari.
- Arrossega un *Button* que servirà per ingressar el número que s'ha posat al *TextBox*.
- Arrossega dues *Image* que serviran per simular una claqueta quan es tanca.
- Arrossega una *Label* on hi haurà una descripció d'aquesta pàgina.
- Arrossega un *TinyDB* que ens serveix per recuperar variables importants.
- Arrossega un *Notifier* per mostrar futures alertes.
- Arrossega un *Sound* per reproduir el so d'una claqueta.
- Arrossega un *TinyWebDB* per recuperar dades web.

Blocks editor

1. Defineix variables i llistes.
2. Arrossega el bloc “*When registrat.Initialize-Do*”:
 - a. Estableix un condicional que mira si hi ha enregistrat un valor de mòbil a *TinyDB*. En cas afirmatiu, mostra un *Notifier* amb el missatge “Ja està registrat” i tanca la pantalla.
 - b. Crida el valor llista_mobils de la *TinyWebDB*.
3. Arrossega el bloc “*When TinyWebDB.GotValue-Do*”:
 - a. Estableix un condicional: si hi ha algun valor de la llista, el guarda a la variable llista_mobils.
4. Arrossega el bloc “*When but_ingressa.Click-Do*”:
 - a. Assigna el valor entrat a la variable num_mobil.
 - b. Estableix la condició: si la longitud d'aquest número és diferent de 9, s'alerta a l'usuari d'incorrecció. En cas contrari, estableix una nova condició per saber si el número ja està registrat. Si ho compleix, s'alerta a l'usuari i, en cas contrari, es crida el procediment guarda.
5. Defineix el procediment “guarda” arrossegant el bloc “*To.Procedure-Arg-Do*”:
 - a. Es fa l'efecte de tancar la claqueta alternant les dues imatges de la claqueta amb el so de la claqueta.
 - b. Afegir el num_mobil (o usuari) a la llista_mobils i la guardem a la *TinyWebDB*.
 - c. Guardar el num_mobil a la *TinyDB* amb l'etiqueta “mobil”



Codi blocs pantalla *registrar*

7.7. Pantalla Valoracions

La funció d'aquest apartat és que l'usuari que tingui l'aplicació pugui qualificar els efectes especials, els actors, la música i el guió de cada una de les pel·lícules que s'han emès. Per poder accedir a les votacions l'usuari haurà d'estar registrat.

Component designer



Pantalla valoracions

- Arrossega un *HoritzontalArrangement* amb dues *Label* que mostri els valors de data i sessió.
- Arrossega un *HoritzontalArrangement* amb una *Label* que mostri el títol.
- Arrossega un *HoritzontalArrangement* amb sis *Label* que mostren: Ítems, i agrupats: Dolent, Regular, Bo, Molt Bo i Excel·lent.
- Arrossega un *HoritzontalArrangement* amb un *Label* que mostra “Guió”, i agrupats cinc *Buttons* amb la propietat *Image* amb una imatge de la claqueta sense seleccionar.
- Repeteix el procés anterior per a actors, música i efectes.
- Arrossega un *HoritzontalArrangement* amb dos *Buttons*: Envia i Borra i un *ListPicker*, Llista.
- Arrossega un *TinyDB* que ens serveix per recuperar variables importants.
- Arrossega un *Notifier* per mostrar futures alertes.
- Arrossega un *FusionTables*, ft_valoracions, que conté les valoracions efectuades.

Blocks editor

1. Defineix variables i llistes.
2. Arrossega el bloc “*When valoracions.Initialize-Do*”:
 - a. Recupera el valor "mobil" de la *TinyDB*.
 - b. Estableix un condicional per mirar si “mobil” existeix. En cas contrari, s’alerta a l’usuari. En cas afirmatiu, recupera de la *TinyDB* les llistes *pelis_titols* i *pelis* i llista *sessions_valorades*. Assigna la llista *pelis_titols* als elements de la llista de selecció *lst_pelis*.
3. Arrossega el bloc “*When but_guiol.Click-Do*”:
 - a. Crida el procediment *apaga_guiol*.
 - b. Visualitza la claqueta de forma que quedi seleccionada.
 - c. Guarda el valor de la selecció a 1.
4. Repeteix el procés per a la resta de botons de guió augmentant el número del valor de la selecció guardat.
5. Repeteix el mateix procés per a cada ítem (actors, música...).
6. Defineix el procediment “*apaga_guiol*” arrossegant el bloc “*To.Procedure-Arg-Do*”:
 - a. Assigna el valor de la imatge de la claqueta no seleccionada a tots els botons.
7. Repeteix el procés per a cada ítem (actors, música...).
8. Arrossega el bloc “*When but_borra.Click-Do*”:
 - a. Crida el procediment “reset”.
9. Defineix el procediment “reset” arrossegant el bloc “*To.Procedure-Arg-Do*”:
 - a. Crida tots els procediments “*apagar_guiol/actors/...*”.
10. Arrossega el bloc “*When but_envia.Click-Do*”:
 - a. Estableix un condicional que mira si s’ha seleccionat una sessió. Si és així, mira que la sessió no hagi estat valorada. En cas negatiu, crida el procediment “guarda”. En cas afirmatiu, surt el missatge “Sessió ja valorada”.
11. Defineix el procediment “guarda” arrossegant el bloc “*To.Procedure-Arg-Do*”:
 - a. Estableix un condicional *Or* que mira si s’han valorat tots els ítems. En cas negatiu, notifica l’usuari. En cas positiu, crea una petició d’inserció de valors a la *FusionTables* *ft_valoracions* amb tots els valors dels ítems seleccionats.

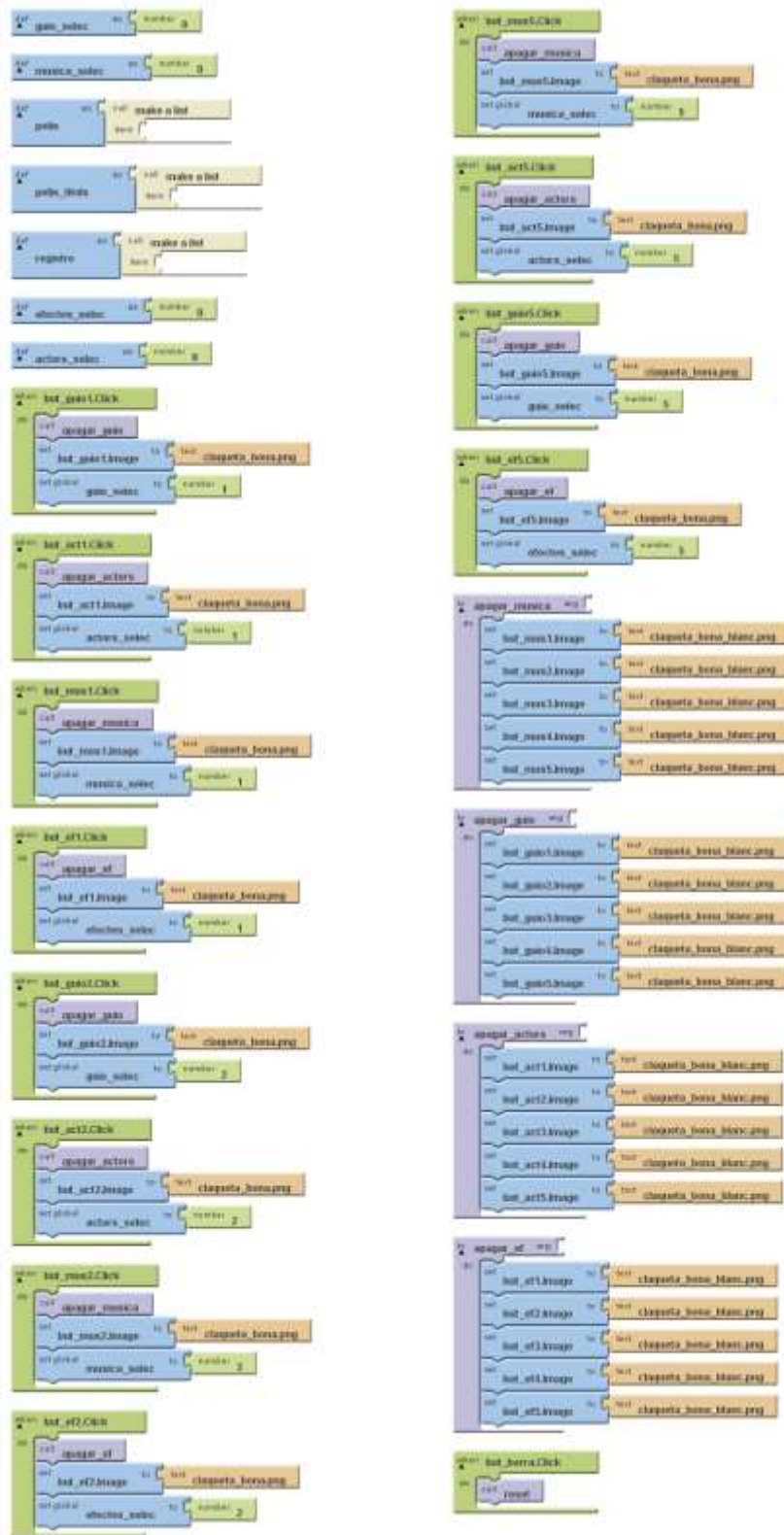
- b. Envia la petició (*SendQuery*).

12. Arrossega el bloc “*When ft_valoracions.GotResult-Do*”:

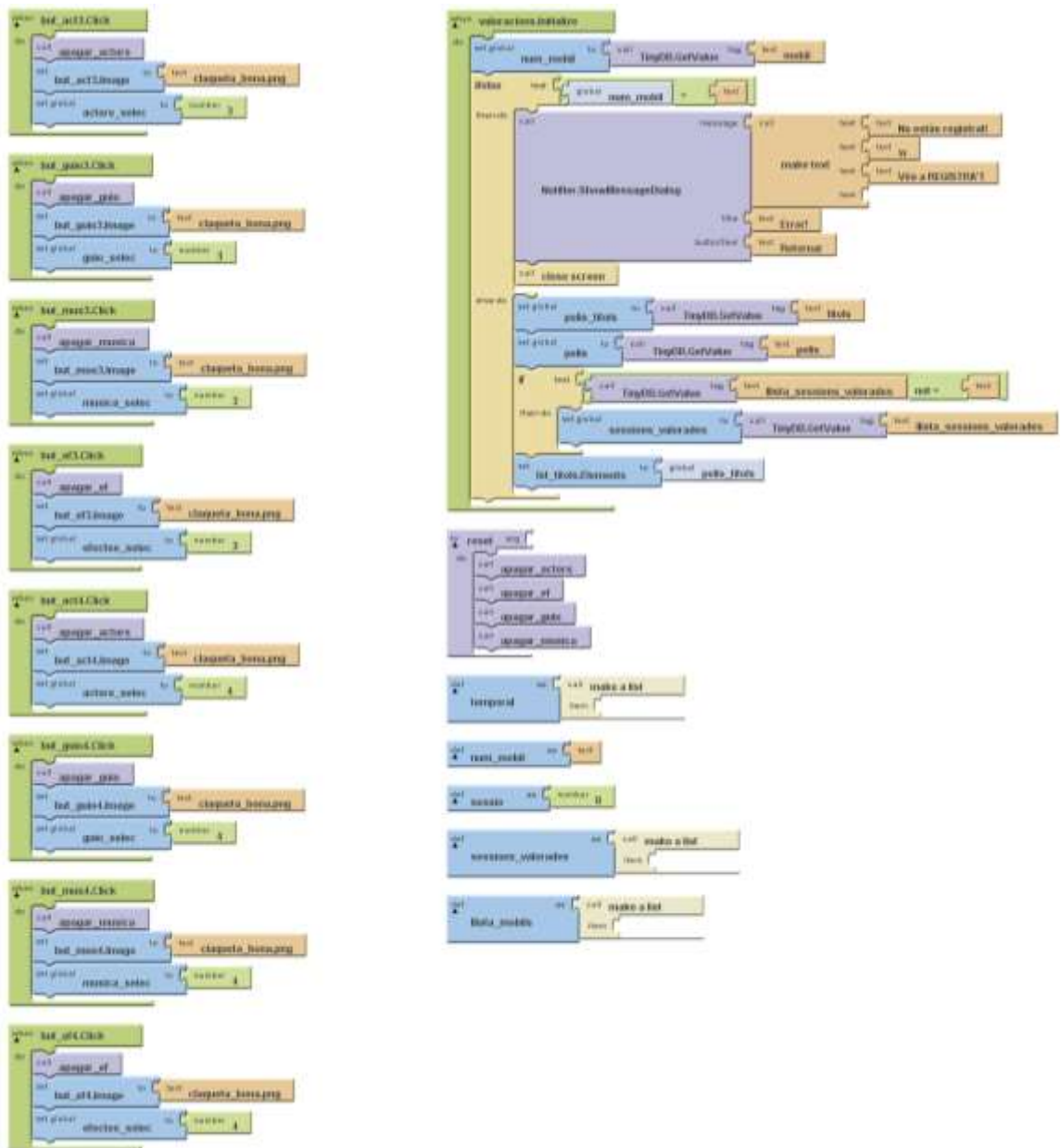
- a. Estableix un condicional que mira si el valor obtingut en la petició hi ha la paraula “rowid”, és a dir, que ha generat una nova fila. Si és així, surt el missatge valoració registrada i afegeix a la llista sessions_valorades la sessió actual. En cas contrari, ho notifica amb el missatge “Error de transmissió”.

13. Arrossega el bloc “*When llista_pelis.AfterPicking-Do*”:

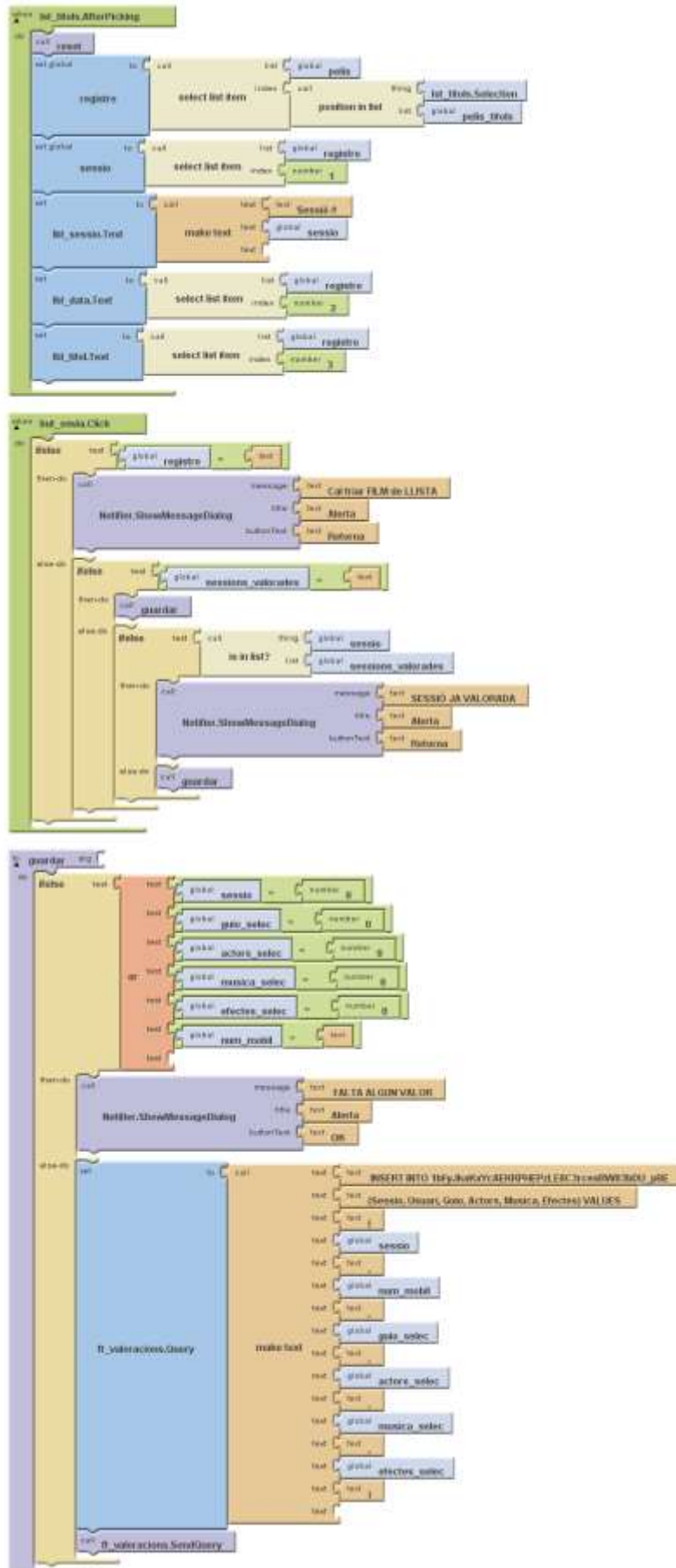
- a. Crida el procediment reseti amb la posició de la pel·lícula seleccionada mostra el títol, la data i la sessió.



Codi blocs pantalla valoracions 1ª part



Codi blocs pantalla valoracions2ª part



Codi blocs pantalla *valoracions* 3ª part

7.8. Pantalla Estadístiques

La seva funció principal és mostrar els resultats de les valoracions que han fet tots els usuaris que tenen la aplicació. Es pot consultar les mitjanes de cada pel·lícula o es pot mirar la mitjana ordenada dels deu primers resultats ordenats per guió, actors... i globalment.

Component designer



Pantalla estadístiques

- Arrossega un *HoritzontalArrangement* i insereix quatre *buttons* de navegació (<<, <, >, >>) i un *ListPicker* per accedir a la llista de pel·lícules.
- Arrossega un *HoritzontalArrangement* i insereix cinc *buttons* que serveixen per mostrar les gràfiques de cada ítem (música, actors...).
- Arrossega un *Label* que servirà per mostrar el nom de la sessió.
- Arrossega un *WebView* que servirà per mostrar les gràfiques generades per la web de *Chart API*.
- Arrossega un *FusionTables* per tal de poder accedir a les valoracions de cada pel·lícula.
- Arrossega un *TinyDB* per recuperar valors anteriorment enregistrats.

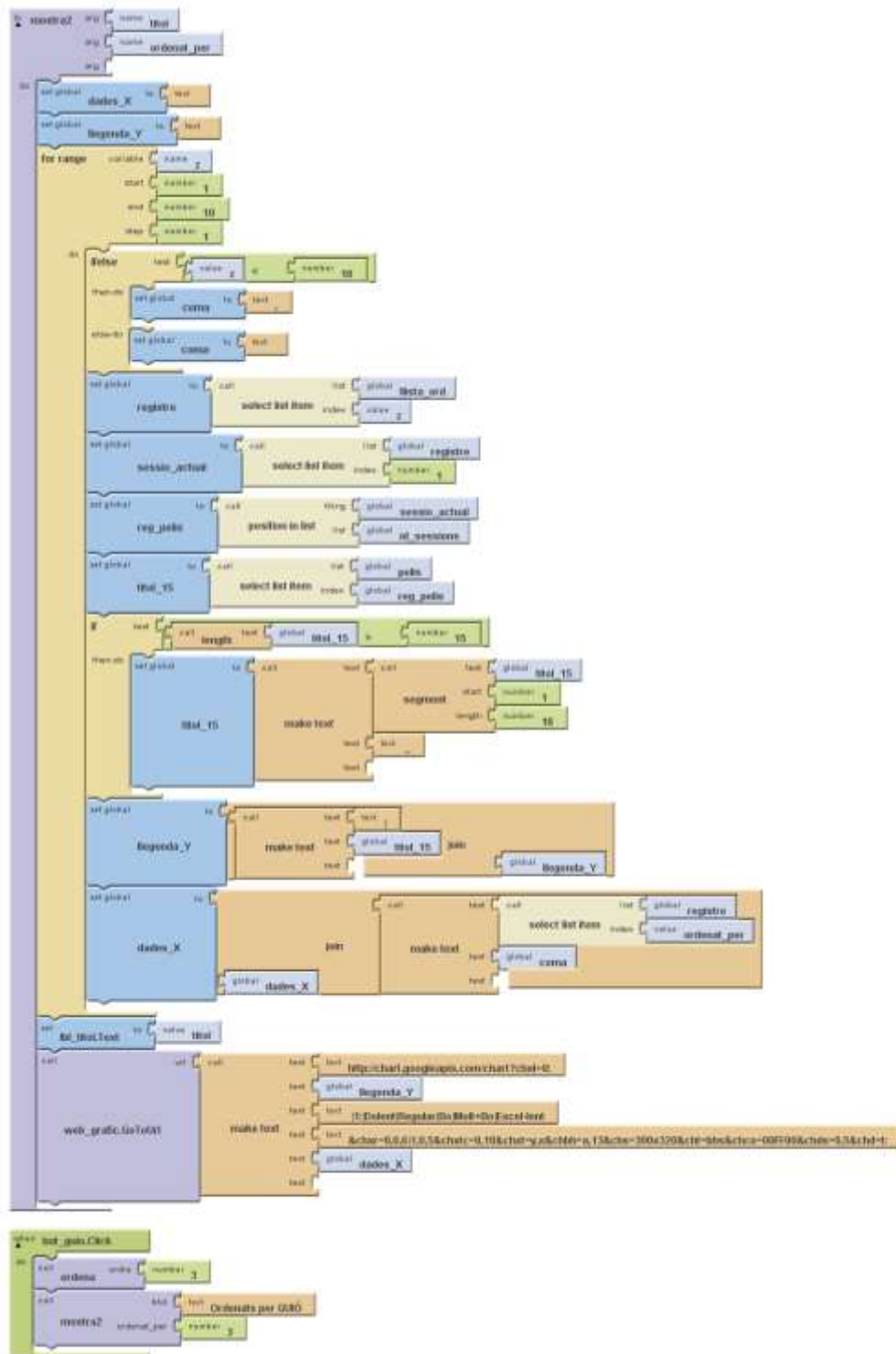
Blocks editor

1. Defineix variables i llistes.
2. Arrossega el bloc “*When estadistiques.Initialize-Do*”:
 - a. Fes una petició a la *FusionTable* per obtenir el total de sessions (COUNT()) i les mitjanes (AVERAGE(X)) dels ítems valorats de cada pel·lícula ordenats per sessió.

- b. Envia la petició.
 - c. Recupera de la TinyDB els valors de les etiquetes “títols” i “id” i assigna’ls a pelis i id_sessions respectivament.
3. Arrossega el bloc “*When ft_valoracions.GotResult-Do*”:
 - a. Crea la llista tots amb el resultat de la consulta i elimina el primer element que conté els títols de les columnes.
 - b. Estableix un bucle per crear la llista sessions que tindrà per cada sessió el número d’usuaris que han votat. Cal calcular la mitjana global del ítems i ho ajusta als dos primers decimals. Crea la llista de tots_global amb les mitjanes anteriors.
 - c. Crea una còpia de la llista tots_global anomenada llista_ord.
 - d. Assigna els elements lst_pelis amb la llista pelis.
 - e. Crida el procediment “mostra” amb el valor 1.
4. Defineix el procediment “mostra” arrossegant el bloc “*To.Procedure-Arg-Do*”:
 - a. Mostra el títol de la sessió i, entre parèntesi, el nombre d’usuaris que han votat.
 - b. Visualitza un gràfic de barres amb les mitjanes dels ítems de la sessió.
5. Arrossega el bloc “*When but_endavant.Click-Do*”:
 - a. Segueix la mecànica explicada a l’apartat “Pel·lícules” amb la única diferència que es crida el procediment “mostra”.
6. Arrossega el bloc “*When lst_pelis.AfterPicking-Do*”:
 - a. Segueix la mecànica explicada a l’apartat “Pel·lícules”.
7. Arrossega el bloc “*When but_guiio.Click-Do*”:
 - a. Crida el procediment “ordena” amb un valor 3 ja que és la posició que ocupa dins el registre de la llista.
 - b. Crida el procediment “mostra2”.
8. Repeteix elprocés per a cada ítem (actors, música...).
9. Defineix el procediment “ordena” arrossegant el bloc “*To.Procedure-Arg-Do*”:
 - a. Estableix un condicional que mira si hi ha dos o més elements a la llista. En cas afirmatiu, continua i, en cas contrari, surt del procediment.

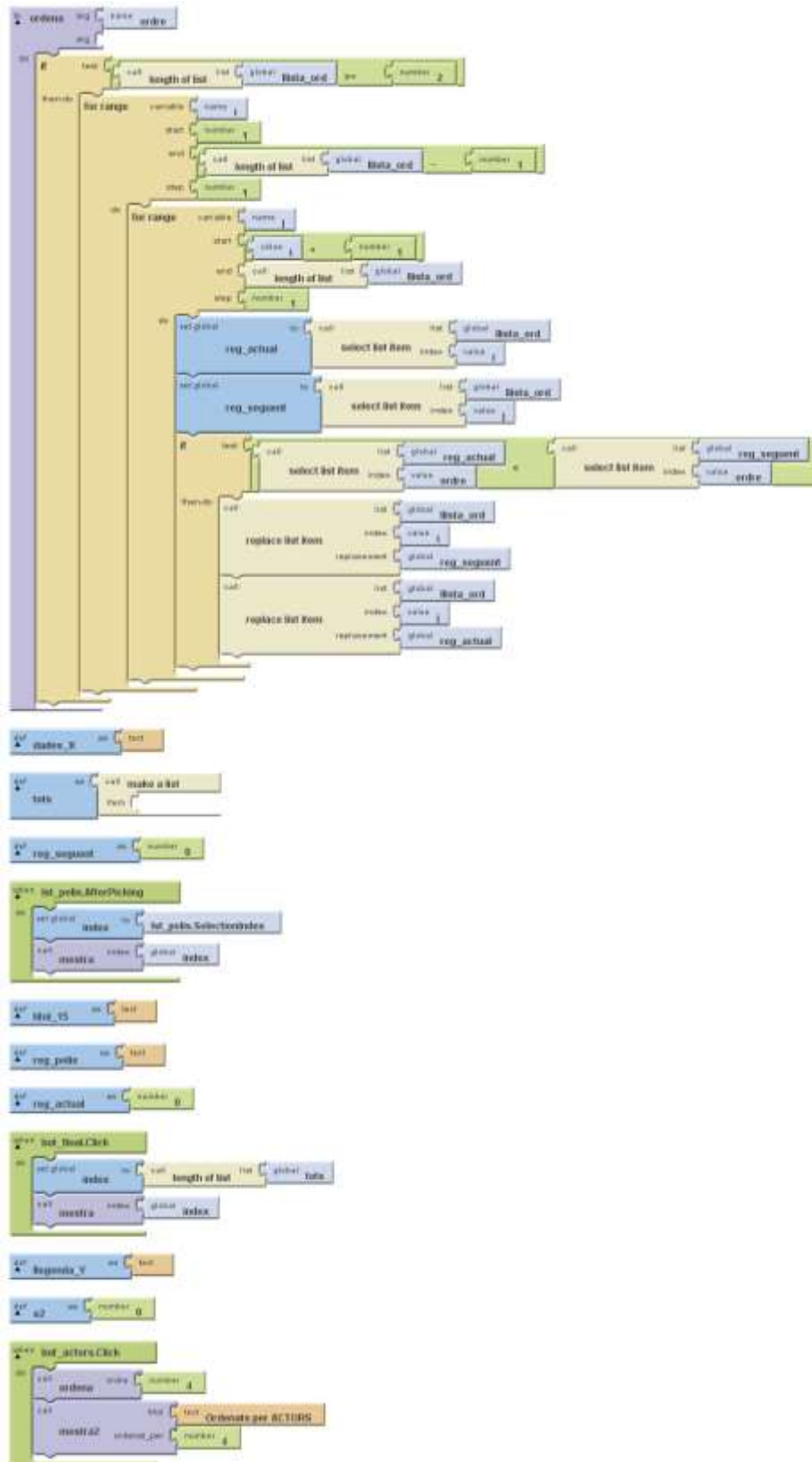
- b. Estableix un bucle per ordenar la llista `llista_ord` seguint el mètode de la bombolla¹⁹ segons l'ordre demanat.
10. Defineix el procediment “mostra2” arrossegant el bloc “*To.Procedure-Arg-Do*”:
- a. Repeteix el procés de “mostra” amb la particularitat de:
 - i. Cal crear les etiquetes pels eixos X (títols).
 - ii. Segmentar el títol als primers deu caràcters.

¹⁹El *Bubble Sort* (ordenació de bombolla en anglès) és un senzill algorisme d'ordenació. Funciona revisant cada element de la llista a ordenar amb el següent, intercanviant-de posició si estan en l'ordre equivocacat. Cal revisar diverses vegades tota la llista fins que no es necessitin més intercanvis, la qual cosa significa que la llista està ordenada.



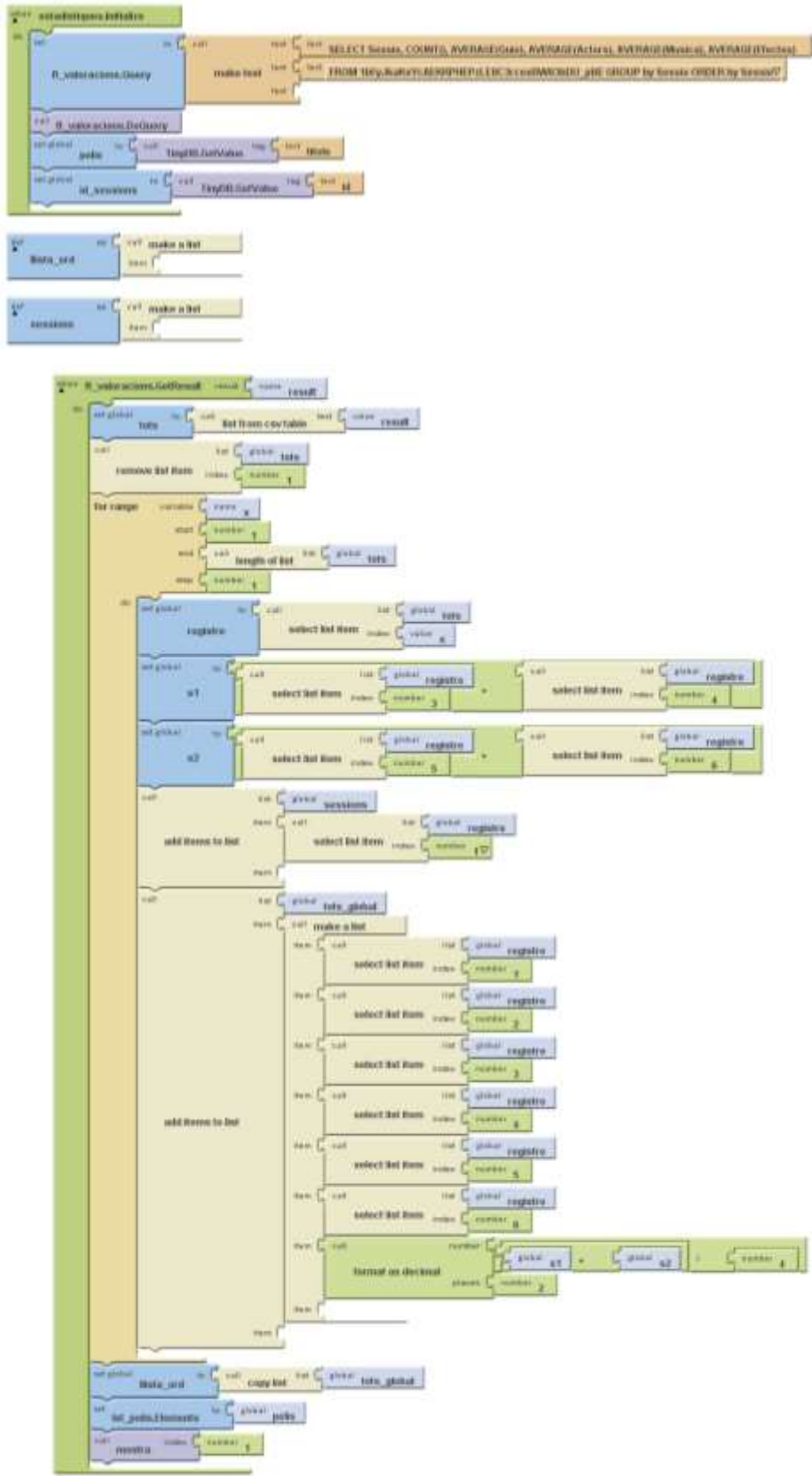
Codi blocs pantalla estadistiques 1ª part

Disseny d'una aplicació Android



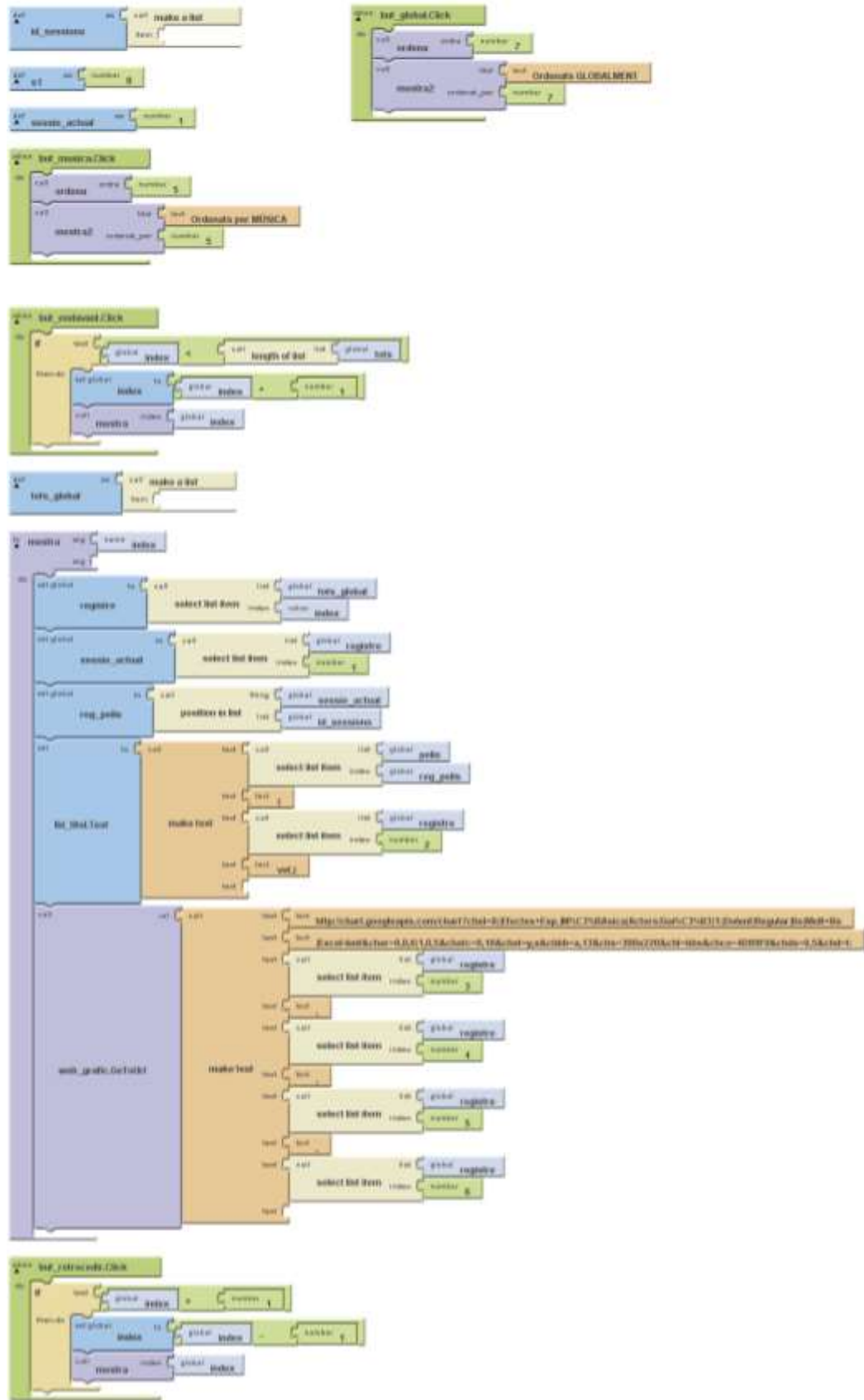
Codi blocs pantalla estadistiques2ª part

Disseny d'una aplicació Android

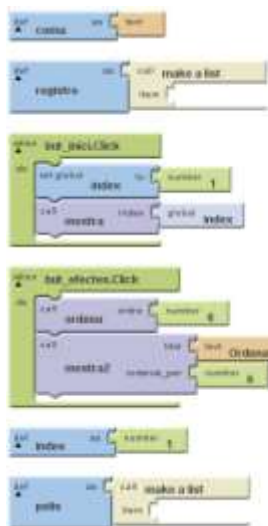


Codi bloc pantalla estadistiques3ª part

Disseny d'una aplicació Android



Codi blocs pantalla estadistiques4ª part



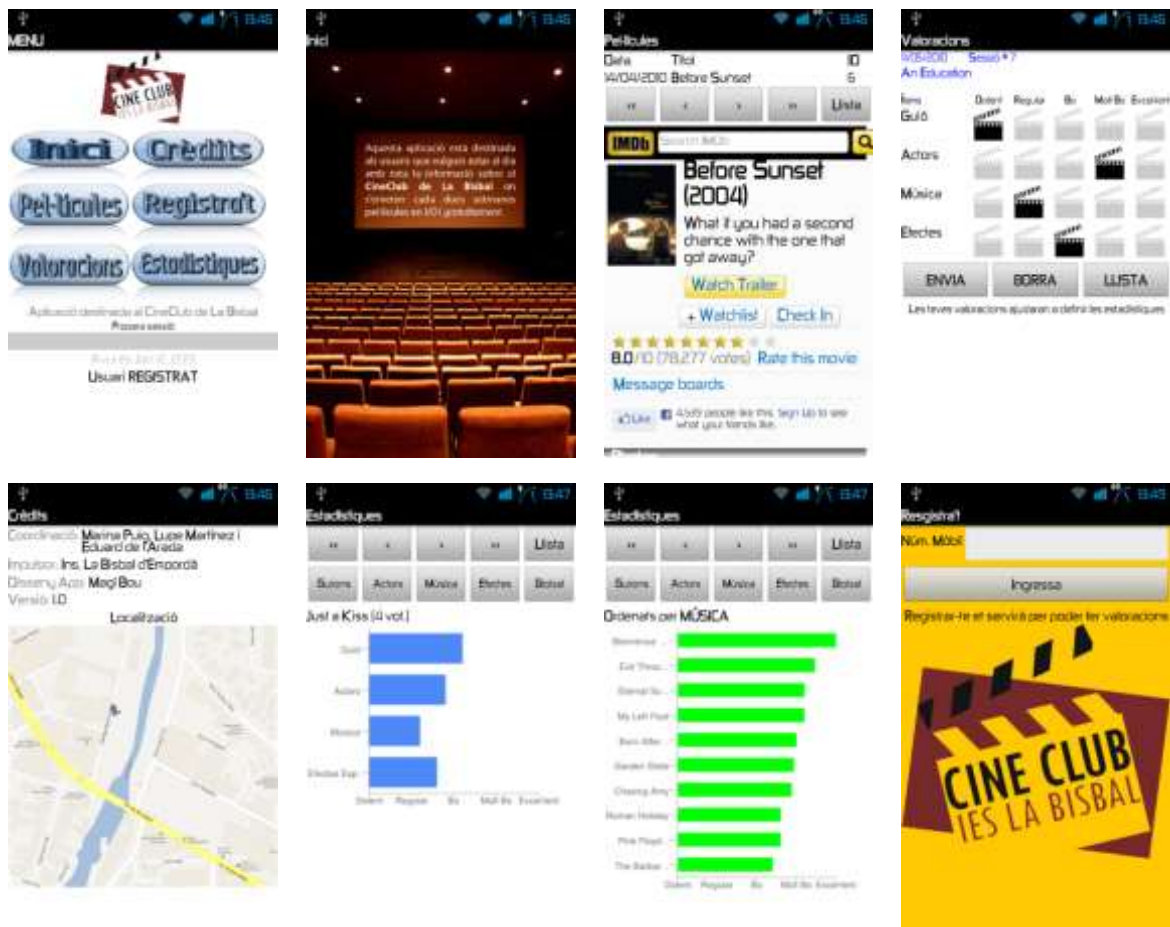
Codi blocs pantalla estadistiques5ª part

7.9. Manteniment de l'app

El manteniment de l'aplicació consisteix en afegir a la *FusionTable* “pelis”, les noves sessions que hi haurà entrant els següents camps: n° de sessió, data, títol i l'enllaç vinculat a la web *IMDb* que prèviament s'haurà hagut de cercar. També cal reiniciar el valor de la *FusionTable* “valoracions” ja que les que he emprat eren generades a l'atzar amb un full de càlcul.

7.10. Visualització al mòbil

A continuació les captures de pantalla de mòbil mostrant el resultat



8. Conclusions

Aquest treball és el resultat de moltes hores de dedicació en les quals he après molt sobre varis temes de programació. Es podria dir que he aconseguit dominar en gran part l'*App Inventor* ja que per arribar a fer aquesta aplicació es necessitava utilitzar moltíssims components d'aquest.

Tot i que primer vaig començar amb una idea de com seria la aplicació, al llarg del temps he canviat coses ja que moltes vegades a la pràctica es veuen coses que quan et plantes a la teoria no veus. Seguidament hi ha una llista de coses que vaig modificar:

- Al principi em vaig plantejar posar un apartat "Calendari" on sortís un calendari assenyalant cada pel·lícula el dia que es passava. Finalment el vaig canviar per un apartat on es pogués valorar les pel·lícules veient que no aportava res un calendari.
- Volia posar un apartat "Configuració" però vaig veure que era innecessari, per tant, vaig substituir-lo per un apartat "Estadística" on es mostrarien els gràfics de les valoracions enregistrades per cada usuari.
- Vaig pensar que podia posar un mapa mitjançant el component *WebView* i el *Google Maps* però a la llarga vaig veure que era més útil posar una imatge del *Google Maps* fixa ja que per algun motiu no acabava de funcionar del tot bé i tampoc s'ajustava a la pantalla.

Encara que la aplicació funciona correctament, hi ha certs punts que encara no he assolit a causa de diferents motius:

- No s'ha penjat la aplicació al *Google Play* ja que volia esperar el vistiplau del tutor, posar-la en pràctica amb més persones (només s'ha provat amb dues) i dissenyar una millor interfície per ella.
- No s'ha creat un bon disseny per la aplicació perquè he tingut certs problemes recentment amb qüestions de programació que afectaven el funcionament de la aplicació. Com és normal, vaig sobreposar el bon funcionament de l'app davant una millor estètica cosa que en posteriors versions es pot millorar.

9. Agraïments

Darrere de qualsevol treball d'aquesta magnitud, la persona que ha elaborat el projecte de ben segur que ha necessitat ajuda d'algun tipus i jo no sóc menys. He d'agrair a bastant gent la seva col·laboració al meu treball ja que de molt segur que sense la seva aportació no l'hagués pogut finalitzar amb èxit.

Primer de tot donar les gràcies la meva família pel suport moral i en especial al meu pare ja que m'ha ajudat en gran part de la programació i en altres temes com la correcte redacció del treball escrit.

Dono gràcies als meus amics, sobretot a en Roger Pi i l'Arnau Llobet per provar l'aplicació en totes les seves fases i a en Joan Seguí per reptar-me a fer una aplicació millor. Agraïxo els consells els dos tutors que em van assignar a la UdG, sobretot en Sergio Gómez durant l'estiu ja que, com que era el començament del treball, els seus consells van servir per orientar-me.

Per últim però no menys important, dono les gràcies al meu tutor del treball de recerca, l'Eduard de la Arada. Gràcies a ell vaig triar el tema de l'aplicació i em va orientar amb dubtes que em van anar sorgint al llarg dels mesos.

10. Bibliografia

Llibres

WOLBER, David. *App Inventor. Create your own Android apps*. Trad. de Javier Díaz Domés. Madrid: Anaya Multimedia, 2011.

TOMÁS GIRONÉS, Jesús. *El gran libro de Android*. Espanya: Marcombo, 2012.

JIMÉNEZ MARÍN, Alfonso. *Aprende a programar con Java*. Espanya: Paraninfo, 2012.

Webs

<http://appinventor.mit.edu/> (Pàgina oficial en anglès)

<http://beta.appinventor.mit.edu/learn/reference/other/tinywebdb.html> (Pàgina en anglès sobre la creació de *TinyWebDB* personalitzada)

<http://cooltext.com/> (Pàgina en anglès per la creació de logos)

<http://support.google.com/fusiontables/answer/2571232/?hl=en> (Pàgina en anglès sobre *FusionTables*)

<http://www.imdb.com/> (Pàgina en anglès amb base de dades de pel·lícules)

<http://www.tair.info/> (Pàgina en anglès)

<http://www.tuappinventorandroid.com/> (Pàgina en castellà)

<https://code.google.com/apis/console/#project:596742715831> (Pàgina en anglès per la creació d'una *API Key*)

https://developers.google.com/chart/image/docs/chart_wizard (Pàgina en anglès sobre creació de gràfics)

<https://sites.google.com/site/appinventormegusta/> (Pàgina en castellà)

<https://sites.google.com/site/appinventorspain/home> (Pàgina en castellà)