Creació de l'App: NENS & CAT

TFM – Desenvolupament d'Aplicacions sobre Dispositius Mòbils

Nom Estudiant (MEI): Pere Rigo Vallbona

Nom Consultor/s: Jordi Ceballos Villach i Jordi Almirall López

Data Lliurament: 11/01/2017

UOC

ÍNDEX DE CONTINGUTS

1 – PLA DE TREBALL

A continuació es definiran, entre altres coses, els objectius del Treball Final i el pla de treball a seguir per tal d'assolir-los al final del semestre.

1.1 – CONTEXT I JUSTIFICACIÓ DEL TREBALL

Avui en dia estem a l'era digital i els nens que neixen en aquesta època ho fan amb 'un mòbil davall el braç (envers de amb un pa davall el braç)'.

Aquests nens, desde ben petits, estan envoltats de tecnologia i, per això, s'hi adapten de manera fàcil, com qualsevol altra cosa molt comuna.

Internet i les tecnologies mòbils són unes eines molt versàtils i les noves generacions les 'mamem' desde la infància.

Si els nens s'acostumen a aprendre coses a través d'un dispositiu mòbil, penso que és un avanç ja que, d'aquesta manera, desde petits veuran els dispositius com una eina per aprendre de manera divertida i no només un aparell per jugar. A més, crec que l'e-Learning és un tema anirà en augment i així els nens s'hi adaptaran de manera més fàcil. Per aquest motiu, he pensat fer la següent aplicació mòbil.

L'Aplicació que vull desenvolupar és una App educativa infantil EN CATALÀ. He cercat al Google Play i NO he trobat cap App educativa infantil per aprendre català. Per això, aquesta aplicació tractarà de omplir aquest 'buit'.

1.2 – OBJECTIUS DEL TREBALL

Els principals objectius del treball són:

- Aprendre desenvolupament en dispositius mòbils.
- Aprendre Android
- Aprendre a gestionar un projecte i dur-lo a terme fins al final
- Desenvolupar una App que pugui ajuda a nens a aprendre català (actualment no hi ha cap App al Google Play que faci aquesta funció).

L'App tindrà un menú principal. L'usuari podrà triar aprendre: colors, parts del cos, etc.

Per exemple, dins de l'apartat cos (hi haurà un botó per a veure totes les imatges d'una categoria en concret), es veuran les imatges d' un braç, un cap, un peu, un dit, etc. Pitjant

damunt l'imatge de la paraula o un botó 'altaveu', l'usuari escoltaria com es diu (pronuncia) la paraula en concret.

A més, davall cada imatge, l'usuari pot escriure (o dir), el nom de la paraula i l'App li dirà si ho ha escrit correctament o no i també podrà veure la solució, en cas de no saber-ho escriure.

Amb aquesta App, per tant, s'aconsegueix que un nen pugui aprendre moltes paraules de diferents temes (parts del cos, colors, etc.), en català. Per a cada paraula, aprendrà com es pronuncia i com s'escriu. Dit d'una altra manera, a parlar i escriure català.

Per tant, la llista de funcionalitats, seria:

- Triar categoria

- Passar a la paraula anterior o següent

- Veure categoria

- Veure solució (Paraula)

- Escoltar paraula

- Dir paraula

- Escriure paraula

- Comprovar si és correcte

- Ocultar 'Veure solució (Paraula)'

- Avís de temps excessiu amb el mòbil

L'App estarà adaptada per ser utilitzada tant en mòbils com tabletes amb plataforma Android. D'aquesta manera, els pares (si volen) no hauran de deixar el seu mòbil personal i el nen podrà emprar la tableta.

1.3 – ENFOCAMENT I MÈTODE SEGUIT

L'estratègia triada per dur a terme el treball és desenvolupar un producte nou. Aquesta estratègia és la més adequada per aconseguir el objectius citats anteriorment (aprendre desenvolupament en dispositius mòbils, aprendre Android...) ja que, d'aquesta manera, faré tot el disseny i implementació i em servirà per aprendre més.

A més, es tracta d'una aplicació no molt complicada i no val la pena adaptar codi d'altres aplicacions, etc.

1.4 – PLANIFICACIÓ DEL TREBALL

L'aplicació funcionarà utilitzant recursos propis del telèfon mòbil, no hi haurà component servidor. Per tant, no hi haurà infraestructura de servidors propis ni tampoc servidors externs ni serveis al núvol, ja que l'aplicació treballarà sempre en local.

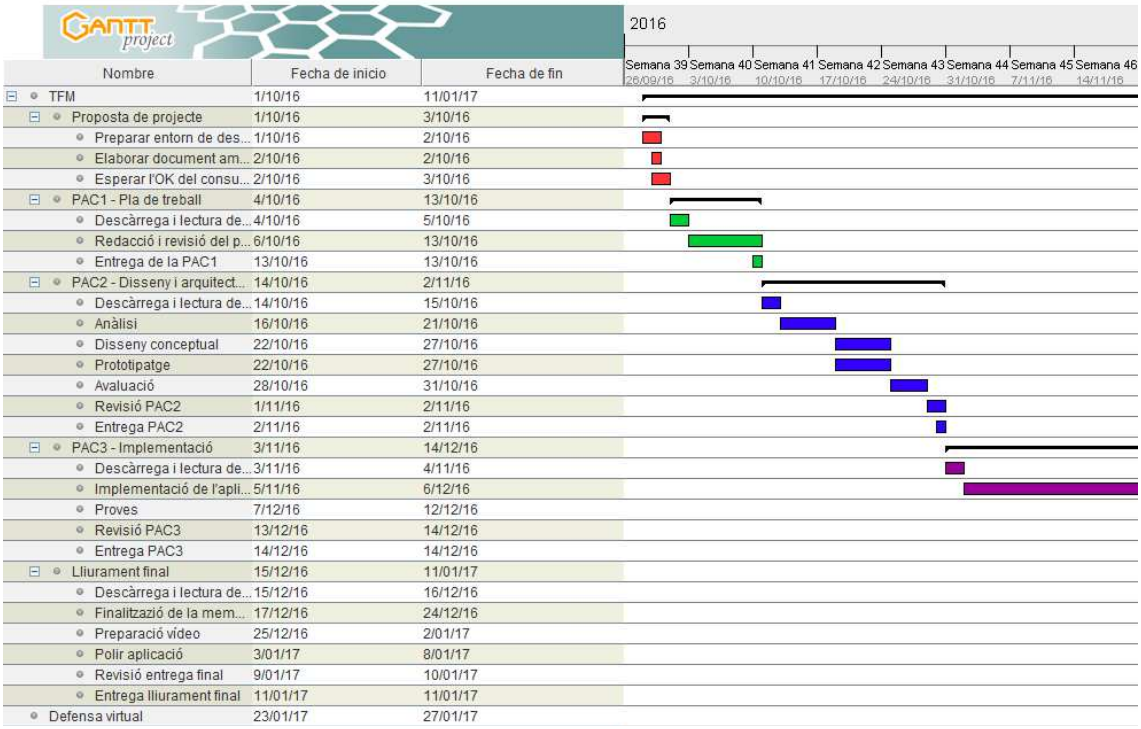
Sabent això, anem a anomenar els principals recursos de programari i de maquinari que s'empraran. Recursos de programari:

Programari	Funció
Microsoft Office 2007	Per a la documentació del projecte (Word) i la fer presentació (PowerPoint)
Android Studio	Entorn de desenvolupament de l'aplicació Android
GanttProject	Per a la planificació del projecte
Mozilla Firefox	Per a la cerca d'informació, etc.
ArgoUML	Fer diagrames UML de casos d'ús i classes
Camtasia Studio	Gravar pantalla i realitzar part de la gravació del vídeo
GIMP	Retocar imatges

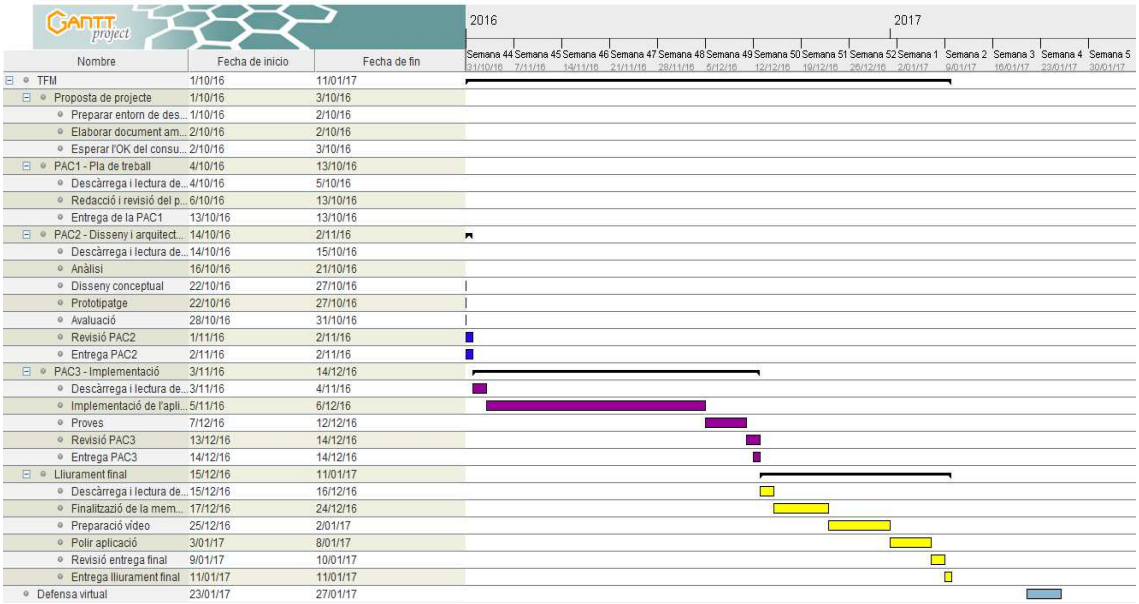
Recursos de maquinari:

Maquinari	Funció
Ordinador (Acer Aspire 5750G)	Per fer la documentació del projecte, el disseny i el desenvolupament de l'App
Telèfon Mòbil (S. Galaxy A5)	Servirà per provar l'aplicació en un dispositiu real
Samsung Galaxy Tab A	Servirà per provar l'aplicació en un dispositiu real

Les tasques a realitzar durant el projecte i la planificació de dites tasques és pot veure en el següent diagrama de Gantt:



Nombre	Fecha de inicio	Fecha de fin
TFM	1/10/16	11/01/17
Proposta de projecte	1/10/16	3/10/16
Preparar entorn de des...	1/10/16	2/10/16
Elaborar document am...	2/10/16	2/10/16
Esperar l'OK del consu...	2/10/16	3/10/16
PAC1 - Pla de treball	4/10/16	13/10/16
Descàrrega i lectura de...	4/10/16	5/10/16
Redacció i revisió del p...	6/10/16	13/10/16
Entrega de la PAC1	13/10/16	13/10/16
PAC2 - Disseny i arquitect...	14/10/16	2/11/16
Descàrrega i lectura de...	14/10/16	15/10/16
Anàlisi	16/10/16	21/10/16
Disseny conceptual	22/10/16	27/10/16
Prototipatge	22/10/16	27/10/16
Avaluació	28/10/16	31/10/16
Revisió PAC2	1/11/16	2/11/16
Entrega PAC2	2/11/16	2/11/16
PAC3 - Implementació	3/11/16	14/12/16
Descàrrega i lectura de...	3/11/16	4/11/16
Implementació de l'apli...	5/11/16	6/12/16
Proves	7/12/16	12/12/16
Revisió PAC3	13/12/16	14/12/16
Entrega PAC3	14/12/16	14/12/16
Lliurament final	15/12/16	11/01/17
Descàrrega i lectura de...	15/12/16	16/12/16
Finalització de la mem...	17/12/16	24/12/16
Preparació vídeo	25/12/16	2/01/17
Polir aplicació	3/01/17	8/01/17
Revisió entrega final	9/01/17	10/01/17
Entrega lliurament final	11/01/17	11/01/17
Defensa virtual	23/01/17	27/01/17



Nombre	Fecha de inicio	Fecha de fin
TFM	1/10/16	11/01/17
Proposta de projecte	1/10/16	3/10/16
Preparar entorn de des...	1/10/16	2/10/16
Elaborar document am...	2/10/16	2/10/16
Esperar l'OK del consu...	2/10/16	3/10/16
PAC1 - Pla de treball	4/10/16	13/10/16
Descàrrega i lectura de...	4/10/16	5/10/16
Redacció i revisió del p...	6/10/16	13/10/16
Entrega de la PAC1	13/10/16	13/10/16
PAC2 - Disseny i arquitect...	14/10/16	2/11/16
Descàrrega i lectura de...	14/10/16	15/10/16
Anàlisi	16/10/16	21/10/16
Disseny conceptual	22/10/16	27/10/16
Prototipatge	22/10/16	27/10/16
Avaluació	28/10/16	31/10/16
Revisió PAC2	1/11/16	2/11/16
Entrega PAC2	2/11/16	2/11/16
PAC3 - Implementació	3/11/16	14/12/16
Descàrrega i lectura de...	3/11/16	4/11/16
Implementació de l'apli...	5/11/16	6/12/16
Proves	7/12/16	12/12/16
Revisió PAC3	13/12/16	14/12/16
Entrega PAC3	14/12/16	14/12/16
Lliurament final	15/12/16	11/01/17
Descàrrega i lectura de...	15/12/16	16/12/16
Finalització de la mem...	17/12/16	24/12/16
Preparació vídeo	25/12/16	2/01/17
Polir aplicació	3/01/17	8/01/17
Revisió entrega final	9/01/17	10/01/17
Entrega lliurament final	11/01/17	11/01/17
Defensa virtual	23/01/17	27/01/17

Com es pot veure al diagrama de Gantt, es segueix la planificació d'entregues de la UOC. Els lliuraments són els següents:

- Lliurament PAC 1 (05 oct, he posat al diagrama que jo ho he entregat el 13 oct): Pla de treball.

- Lliurament PAC 2 (02 nov): Disseny i arquitectura.

- Lliurament PAC 3 (14 des): Implementació.

- Lliurament final (11 gen): Memòria i Vídeo-demo amb la presentació del projecte.

- Defensa virtual (23 al 27 gen): Període de debat virtual entre tots els participants de l'aula.

IMPORTANT: DEDICARÉ UNES 4 HORES AL DIA AL TREBALL FINAL.

1.5 – RISCS I COSTOS DEL PROJECTE

A continuació es mostren riscs que poden afectar el projecte:

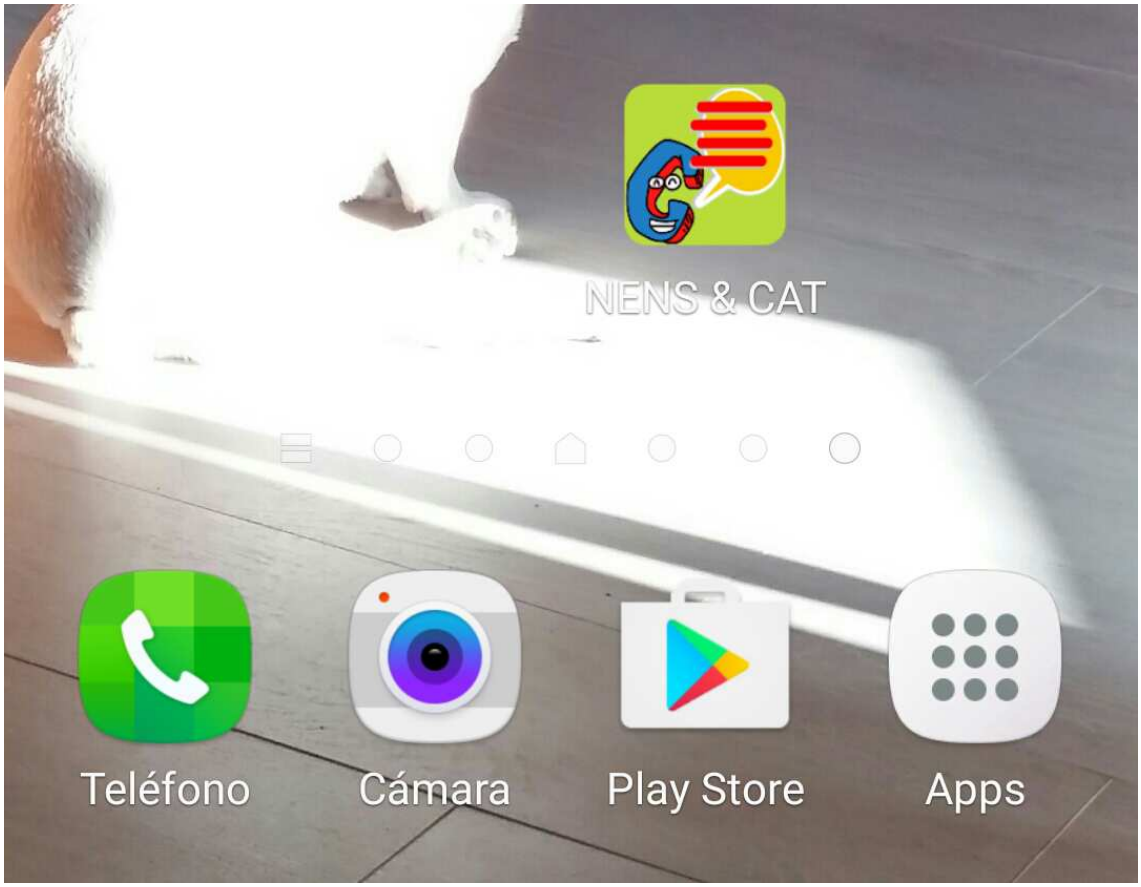
Risc	Descripció	Prob. Aparició	Impacte	Accions mitigadores
[R01] Planificació incorrecte i riscs no previstos	A vegades no es fa una bona planificació i, al seguir-la, ens donem compte d'això i també ocorren coses no previstes. Aquestes coses	Mitjà	Mitjà	- Seguiment estricte del calendari de lliuraments - Revisió constant de l'evolució del projecte

	fan que t'endarrereixis en el projecte			- Anàlisi periòdic de riscs
[R02] Averia de l'equipament	L'ordinador on es fa el desenvolupament de l'App deixa de funcionar	Baix	Alt	- Realitzar còpies de seguretat diàries
[R03] Manca d'experiència en desenvolupament Android	Si no es té experiència en desenvolupar aplicacions mòbils, tot és una novetat i es tarda més i et trobes amb més imprevistos. Per tant, s'endarrereix el projecte	Alt	Alt	- Cercar i llegir molta informació a Internet. - Anar practicant tot el temps possible

Pel que fa al costos del projecte, no hi ha costos a considerar, ja què no necessito adquirir hardware, llicències de software ni contractar un hosting per a realitzar el projecte.

1.6 – PRODUCTES OBTINGUTS

El resultat d'aquest Treball Final de Màster és una App senzilla e intuïtiva que permet a nens poder aprendre de, manera divertida, paraules en català.



A l'imatge anterior es pot veure l'icona de l'App un cop instal·lada. En els propers apartats d'aquesta memòria es veuran més imatges de l'App, aposta ara no les vull posar per no esser repetitiu pel que fa a imatges.

Apart de l'App obtinguda, també s'obtindrà la memòria del treball final i també un vídeo amb la presentació del projecte.

1.7 – DESCRIPCIÓ DELS ALTRES CAPÍTOLS

- El capítol 1 serà aquest apartat (Pla de treball).

- El capítol 2 contindrà el Disseny Centrat en l'Usuari (DCU).

- El capítol 3 contindrà el Disseny Tècnic de l'App (la definició dels casos d'ús i el disseny de l'arquitectura.

- El capítol 4 parlarà de com he implementat cadascuna de les funcionalitats de l'App.

- El capítols 5 i 6 seran les conclusions del treball i les fonts d'informació.

2 – DISSENY CENTRAT EN L'USUARI

Els següents apartats estant formats per quatre parts que segueixen les fases del DCU: (anàlisi – disseny – avaluació):

1. Usuaris i context d'ús [Anàlisi]
2. Disseny conceptual [Disseny]
3. Prototipatge [Disseny]
4. Avaluació [Avaluació]

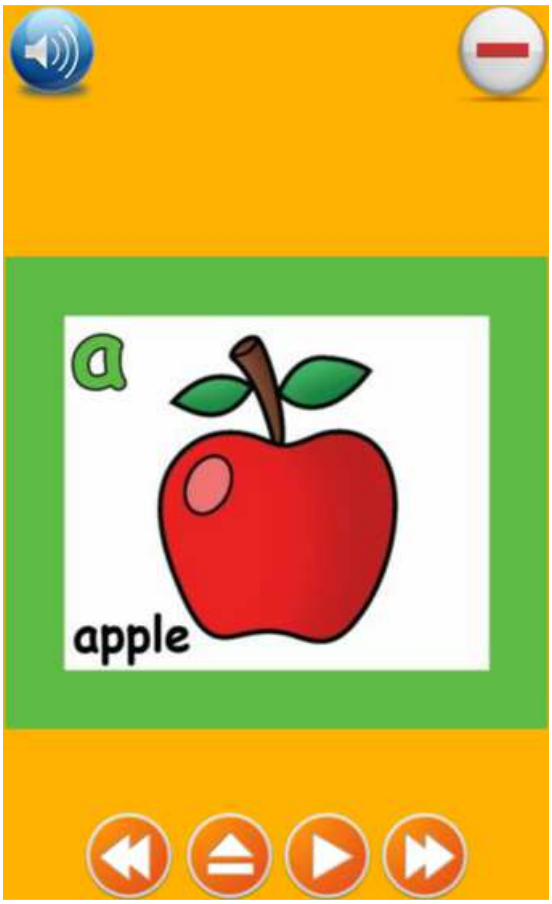
2.1 – USUARIS I CONTEXT D'ÚS

D'entre els mètodes d'indagació que hi ha, he triat dur a terme el d'observació i el de benchmarking ja que, en el meu cas, es tracta d'una App infantil. D'aquesta manera, crec que els mètodes d'indagació triats són els més adequats quan els usuaris, dels quals has d'extreure informació, són nens.

Les aplicacions que he fet servir per a dur a terme els mètodes d'indagació són:

- WordPic Catalan: (he anat a opcions i a 'Challenge Modes', he triat l'opció 'Enter translation'). D'aquesta manera, aquesta App es pareix molt a l'App que vull desenvolupar pel que fa a que es veu l'imatge d'un l'objecte i pots escriure o dir la paraula i comprovar que ho has escrit o dit bé. També es pot escoltar com es pronuncia la paraula i també veure directament la resposta correcte (la solució de com s'escriu correctament).

- Educational games for kids: aquesta App és pareix molt a l'App que vull desenvolupar pel que fa al menú principal, a les diferents categories d'objectes o paraules i que es pot escoltar com es pronuncia la paraula i també veure directament la resposta correcte (la solució de com s'escriu correctament) pitjant un botó que oculta o no com s'escriu la paraula.



L'observació l'he dut a terme de la següent manera:

Per a fer aquest mètode d'indagació he observat un nen de 4 anys (m'hagués agradat fer-ho amb més nens, però no m'ha estat possible) mentre utilitzava cadascuna de les dues Apps descrites a dalt.

En concret, amb l'App 'WordPic Catalan' he fet que el nen amb el micròfon gravés la paraula que jo li havia dit. Tot seguit, també l'hi he fet escriure una paraula concreta. Tant amb un cas com a l'altra, si s'equivocava dient o escrivint la paraula, em fitxava si sen adonava de on estava l'error i rectificava correctament.

Amb l'App 'Educational games for kids' he que el nen cerqui un color concret, un animal concret i una lletra concreta, i fitxar-me si es sabia moure correctament pel menú i recórrer les diferents paraules d'una categoria i també si reconeixia bé les imatges i l'App era intuïtiva.

En resum, els resultats/conclusions obtinguts amb l'observació són:

- L'App 'Educational games for kids' té un menú molt intuïtiu, faré un menú paregut.
- L'App WordPic Catalan' diu on t'equivoques quan escrius o dius una paraula. Això ho intentaré implementar, ja que es important saber si t'equivoques però és més important saber on t'equivoques.

El benchmarking l'he dut a terme de la següent manera:

Per començar, dir que he abans de triar les dues Apps que es descriuen a abans, he provat i descartat moltes altres Apps infantils fins a trobar les que es pareixen més a l'idea d'aplicació que tinc desde un principi i que ja he explicat a l'introducció de la memòria.

La meva App vull que sigui una mescla de les Apps WordPic Catalan (m'interessa sobretot la funcionalitat de poder tant escriure o dir la paraula, amb el reconeixement de veu) i Educational games for kids (m'interessa sobretot l'estètica i el menú principal i les diferents categories).

D'aquesta manera, he provat dites Apps per a extreure conclusions i veure com puc adaptar la App 'WordPic Catalan' per a fer-ho intuïtiu i fàcil per a nens. També m'ha servit provar 'Educational games for kids' per a veure quines categories de paraules puc afegir, quines imatges, colors, etc. posaria a la meva App i com fer un menú intuïtiu pels nens.

En resum, els resultats/conclusions obtinguts amb el benchmarking són:

- Cada botó ha de tenir una imatge, símbol, etc. que representi clarament el que fa.

- Una App infantil, com més senzilla sigui, millor.

- L'App ha d'esser atractiva pels nens.

A continuació, descriuré cadascun dels diferents perfils d'usuari identificats.

Ja que l'aplicació que faré és una App infantil senzilla i, per tant, els usuaris seran nens (encara que, excepcionalment, també poden esser usuaris d'aquesta App estrangers, nens o no, que volen començar a aprendre algunes paraules en català) no hi ha molts diferents perfils d'usuari.

Per aquest motiu, el lloc que fer diferents fitxes per cadascun dels diferents perfils d'usuari identificats, crec que quedarà més clar si, per una banda, dic els diferents perfils d'usuari i per l'altra, els contexts d'ús, tasques, etc.

Els perfils d'usuari podien esser:

- Nens que no coneixen les lletres.

- Nens que coneixen les lletres.

- Nens que comencen a aprendre a escriure.

- Nens que no saben escriure.

- Nens que no saben llegir.

- Nens que saben llegir.

- Nens de 2-3 anys.

- Nens de 3-4 anys.

- Nens de 4-5 anys.

- Els pares, ja que poden ensenyar als nens a utilitzar bé l'App, en cas de necessitar-ho.

-etc.

*He posat, com a diferents perfils, a nens amb les franges d'edat que es veuen a dalt. Ho he fet perquè, a aquestes edats, els nens canvien molt d'un any a l'altra. Per exemple, un nen de 4 o 5 anys, és més competitiu que un de 2 o 4 anys i, en el cas d'haver-hi un cronòmetre a l'App, intentarà solucionar el màxim de paraules amb el menor temps.

Els contexts d'us podrien esser:

- Ús acompanyat d'un adult o no.

- A l'escola o a casa.

- Per entreteniment o per aprendre.

Les tasques que ha de fer l'aplicació, com a mínim, són:

- Poder escoltar paraula.

- Poder llegir paraula.

- Poder passar d'una paraula a una altra.

- Poder escriure una paraula (i comprovar si és correcte).

- Poder dir una paraula (i comprovar si és correcte).

- Veure l'imatge de la paraula.

Les característiques o elements que han d'estar presents a l'interfície de l'aplicació, són:

- Botó per escoltar paraula.

- Botó per micròfon.

- Línea o espai on poder posar la paraula que has escrit o on el veu la paraula que has dit amb el micròfon.
- Imatges clares i botons grans.

- Colors atractius.

- Botons per passar a l'imatge anterior o següent.

- Botó per comprovar si la paraula esta ben dita o escrita.

- etc.

2.2 – DISSENY CONCEPTUAL

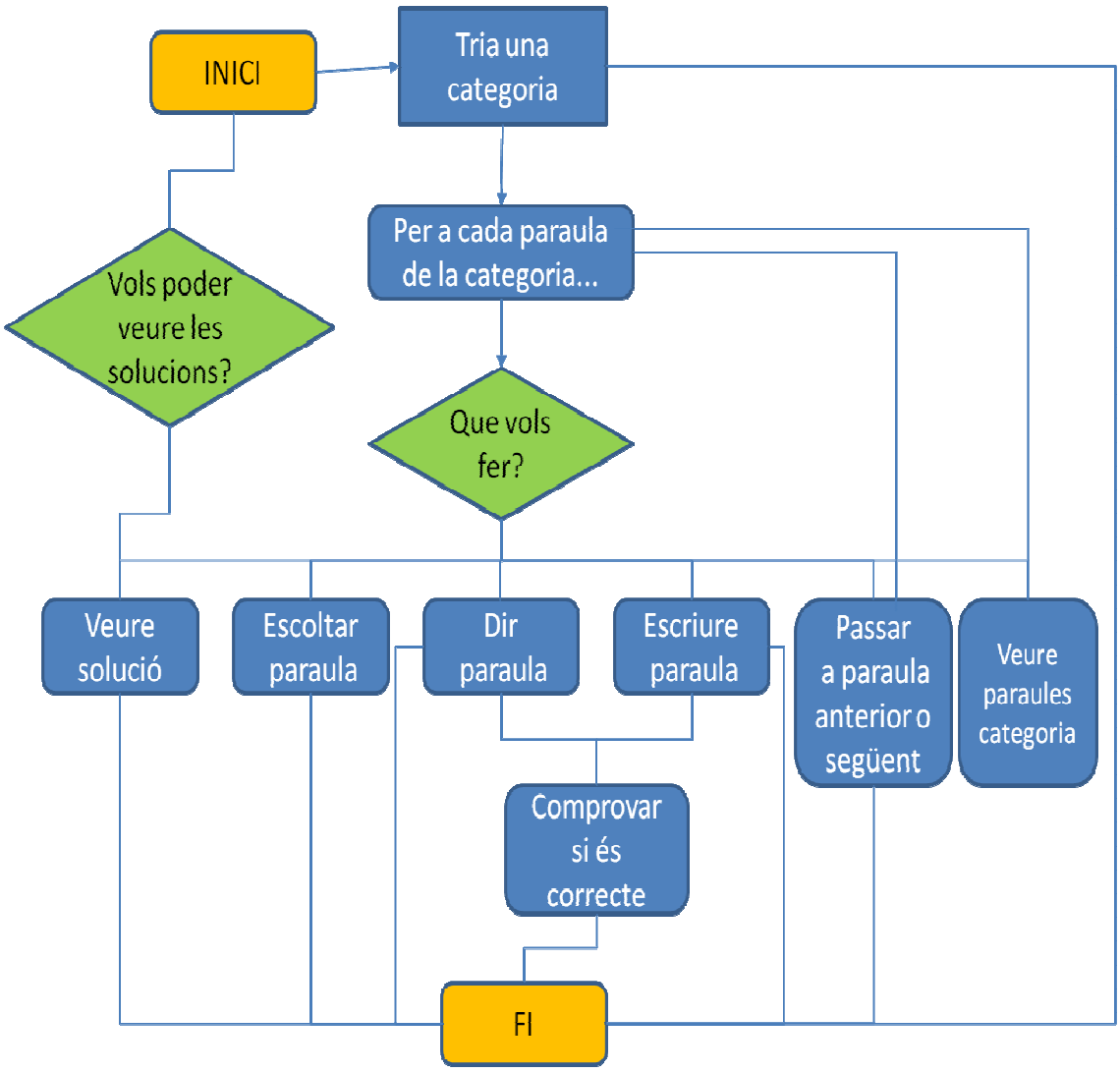
Al igual que en el punt anterior (on havia de dir els perfils d'usuari, context d'ús, etc) i ja que es tracta d'una aplicació infantil i el DCU enfocat a nens és diferent a altres casos, descriuré unes quantes situacions o casos d'ús que es poden donar utilitzant l'aplicació, sense entrar en detall del perfil d'usuari que intervé ni el context on es troba, etc, ja que ja els he anomenat al punt anterior i poden intervenir molts perfils d'usuari , contextos, etc.

- escenari 1: un nen entra en l'aplicació, tria una categoria i es posa a escoltar com es diuen les paraules de la categoria. Té com a objectiu escoltar les paraules. Per a escoltar cada paraula, pitja el botó amb una imatge d'un altaveu que hi ha a dalt de cada paraula. Per moure's per totes les paraules va pitjant les fletxes, per anar a l'imatge anterior o a l'imatge següent.

- escenari 2: un nen entra en l'aplicació, tria una categoria, veu una paraula i vol escriure el nom de la paraula. Per això, es posa damunt una línia que hi ha davall l'imatge i amb el teclat escriu la paraula. Una altra opció és pitjar el botó amb l'imatge d'un micròfon i dir la paraula i aquesta quedarà escrita a la línia que he anomenat abans. Tot seguit, si es té l'objectiu de saber si l'ha escrita o dita correctament es pitja el botó que hi ha a baix que posa 'ES CORRECTE?' i l'App li dirà si ho ha fet bé o no.

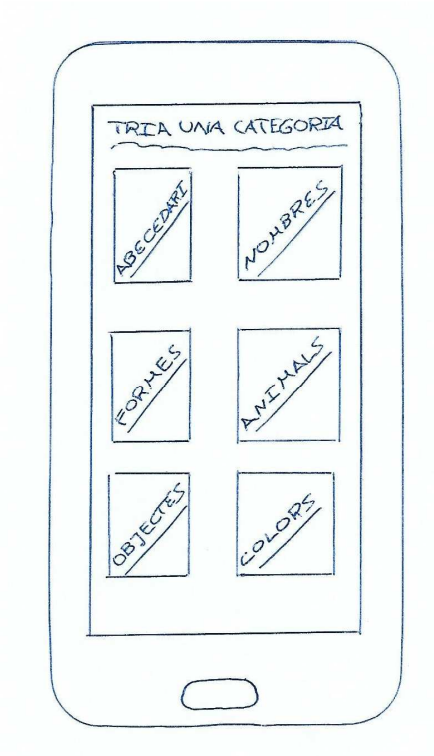
- escenari 3: un nen entra en l'aplicació, tria una categoria i té com a objectiu llegir les paraules de la categoria. Per a llegir cada paraula, es mou per totes les paraules pitjant les fletxes, per anar a l'imatge anterior o a l'imatge següent i, a cada paraula, selecciona el 'checkbox' que hi ha a baix de cada imatge que posa 'SOLUCIÓ'.

Tot seguit, es mostra el diagrama de fluxos d'interacció de l'aplicació:

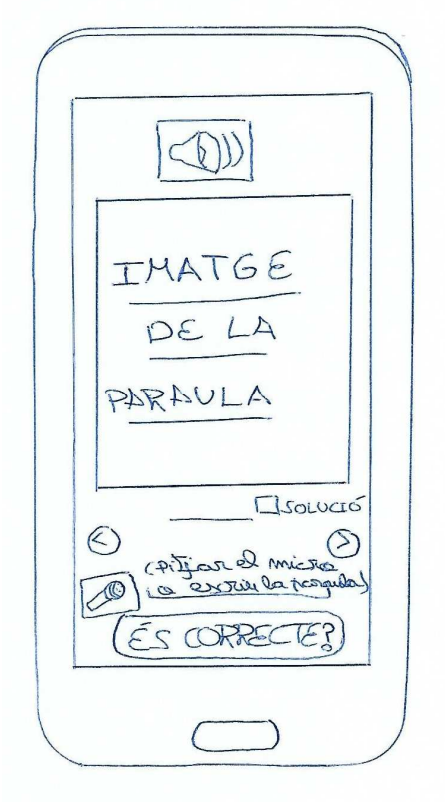


2.3 – PROTOTIPATGE

Tot seguit es veuen els “sketches” escanejats.



A l’imatge anterior és veu l’sketch de la pantalla del menú principal.



Al sketch anterior, es veu la pantalla principal, on es veurà (entre altres coses) cadascuna de les imatges de cada paraula. A continuació es mostren els prototips horitzontals d'alta fidelitat:

Menú Principal

Pantalla principal

Paraula correcte

Paraula incorrecte

2.4 – AVALUACIÓ

Les preguntes d'informació sobre l'usuari que faria són:

Quina edat tens?

Quin curs estàs cursant?

Coneixes les lletres?

Saps llegir?

Quin idioma parlen a casa teva?

T'ensenyen català a l'escola?

Ja que el test l'han de realitzar nens, es tracta de simplificar les tasques. Per tant, a cada nen li demanaré que faci les 3 següents tasques senzilles:

- Cercar l'imatge d'un animal i que escolti el nom del animal.

- Posar-li l'imatge d'un color i fer que amb el micròfon digui quin color és.

- Posar-li l'imatge d'una lletra i que escrigui amb el teclat la lletra.

De la mateixa manera que les tasques han d'esser senzilles, les preguntes referents a les tasques que es faran als nens, també seran molt senzilles. D'aquesta manera, per a cadascuna de les 3 tasques anteriors, es faran les següents preguntes al nen que realitzi el test:

- T'ha agradat l'aplicació?

- La tornaries a utilitzar?

- Que canviaries, llevaries o afegiries?

- Ha estat fàcil dur a terme la tasca?

3 – DISSENY TÈCNIC

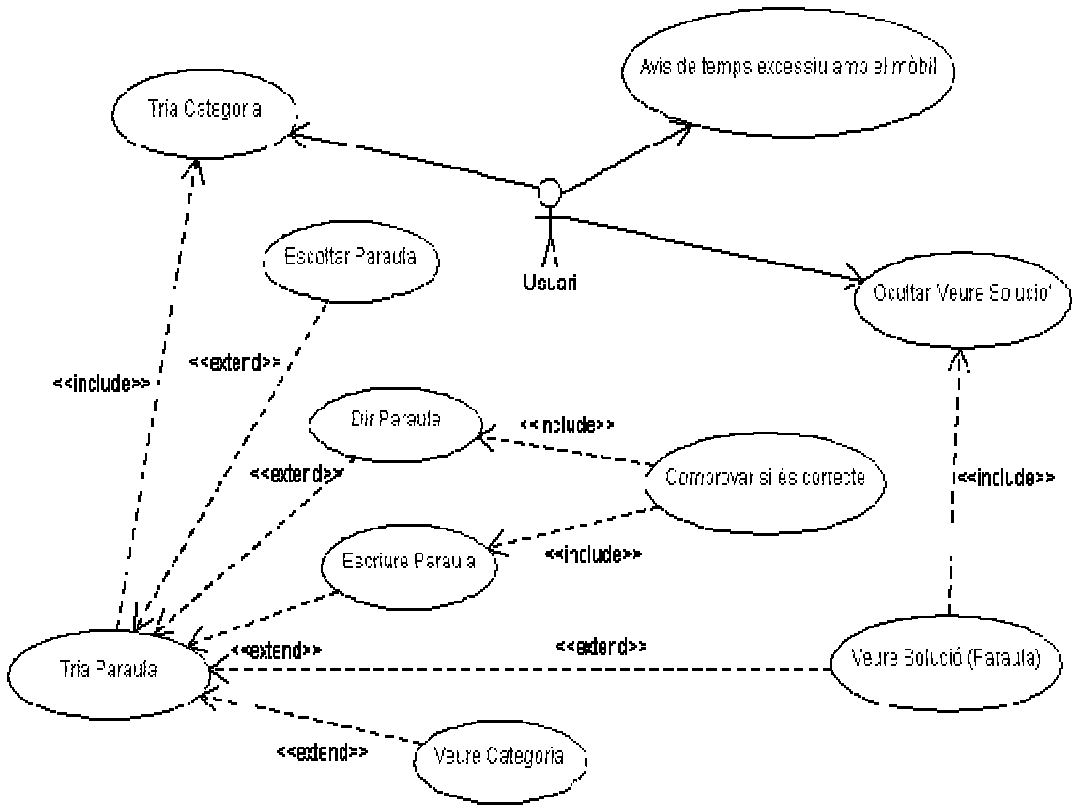
En aquest apartat es realitza el disseny tècnic. En concret es treballen els dos aspectes següents del disseny tècnic:

1 – Definició dels casos d'ús.
2 – Disseny de l'arquitectura.

A continuació es mostra com he fet cadascun d'aquests dos aspectes.

3.1 – DEFINICIÓ DELS CASOS D'ÚS

A continuació es mostra el diagrama UML dels casos d'us i, tot seguit, la llista dels casos d'ús on s'especifiquen actors, precondicions, flux, postcondicions, etc:



Identificador	CU-001
Nom	Tria Categoria
Prioritat	Alta
Descripció	L'usuari entra a l'App i, en el menú principal, tria una de les categories de paraules
Actors	Usuari (nens)
Pre-Condicions	S'ha entrat a l'App
Iniciat per	Botó del 'gridView' del menú principal corresponent a la categoria que es tria
Flux	Inici → Tria Categoria
Post-Condicions	Es veu una paraula de la categoria triada
Notes	--

Identificador	CU-002
Nom	Tria Paraula
Prioritat	Alta
Descripció	L'usuari es mou a través de les paraules de la categoria i tria una paraula
Actors	Usuari (nens)
Pre-Condicions	S'ha triat una categoria
Iniciat per	Botó 'endavant' i 'endarrera' o passant imatge amb el dit
Flux	Inici → Tria Categoria → Tria Paraula
Post-Condicions	Es veu l'imatge de la paraula, de la categoria triada, la pantalla principal
Notes	--

Identificador	CU-003
Nom	Escoltar Paraula
Prioritat	Normal
Descripció	L'usuari pitja el botó 'Altaveu' que hi ha damunt l'imatge de la paraula o fent 'longClick' amb el dit damunt l'imatge, per escoltar com es pronuncia la paraula
Actors	Usuari (nens)
Pre-Condicions	S'ha triat una paraula
Iniciat per	Botó altaveu
Flux	Inici → Tria Categoria → Tria Paraula → Escoltar Paraula
Post-Condicions	S'ha escoltat la paraula
Notes	--

Identificador	CU-004
Nom	Dir Paraula
Prioritat	Normal
Descripció	L'usuari pitja el botó 'Micròfon' que hi ha a baix de l'imatge de la paraula, per pronunciar la paraula
Actors	Usuari (nens)
Pre-Condicions	S'ha triat una paraula
Iniciat per	Botó micròfon
Flux	Inici → Tria Categoria → Tria Paraula → Dir Paraula
Post-Condicions	S'ha dit la paraula al micròfon i es mostra al 'editText'
Notes	--

Identificador	CU-005
Nom	Escriure Paraula
Prioritat	Normal
Descripció	L'usuari escriu a la línia que hi ha a baix de l'imatge de la paraula
Actors	Usuari (nens)
Pre-Condicions	S'ha triat una paraula
Iniciat per	Cursor / Teclat
Flux	Inici → Tria Categoria → Tria Paraula → Escriure Paraula
Post-Condicions	S'ha escrit la paraula i es mostra al 'editText'
Notes	--

Identificador	CU-006
Nom	Comprovar si és correcte
Prioritat	Alta
Descripció	L'usuari pitja el botó 'Verificar' que hi ha a baix de l'imatge de la paraula per comprovar si la paraula (que ha dit o escrit) és correcte
Actors	Usuari (nens)
Pre-Condicions	S'ha dit o escrit una paraula
Iniciat per	Botó 'Verificar'
Flux	Inici → Tria Categoria → Tria Paraula → Dir / Escriure Paraula → Comprovar si és correcte
Post-Condicions	L'App mostra si la paraula (dita o escrita) que hi ha al 'editText' és correcte
Notes	--

Identificador	CU-007
Nom	Veure Paraula
Prioritat	Normal
Descripció	L'usuari selecciona el 'checkbox' que hi ha a baix de l'imatge de la paraula per veure la paraula (solució)
Actors	Usuari (nens)
Pre-Condicions	S'ha triat una paraula
Iniciat per	Checkbox
Flux	Inici → Tria Categoria → Tria Paraula → Veure Paraula
Post-Condicions	Es veu la paraula (solució)
Notes	--

Identificador	CU-008
Nom	Ocultar 'Veure Paraula'
Prioritat	Normal
Descripció	L'usuari selecciona (o desselecciona) el 'checkbox' que hi ha al menú 'Overflow' del ActionBar del menú principal
Actors	Usuari (nens)
Pre-Condicions	L'App no s'ha iniciat
Iniciat per	CheckBox Solucions
Flux	Inici → Ocultar 'Veure Paraula'
Post-Condicions	Es mostra o no el 'CheckBox'(Solució) de la pantalla principal
Notes	--

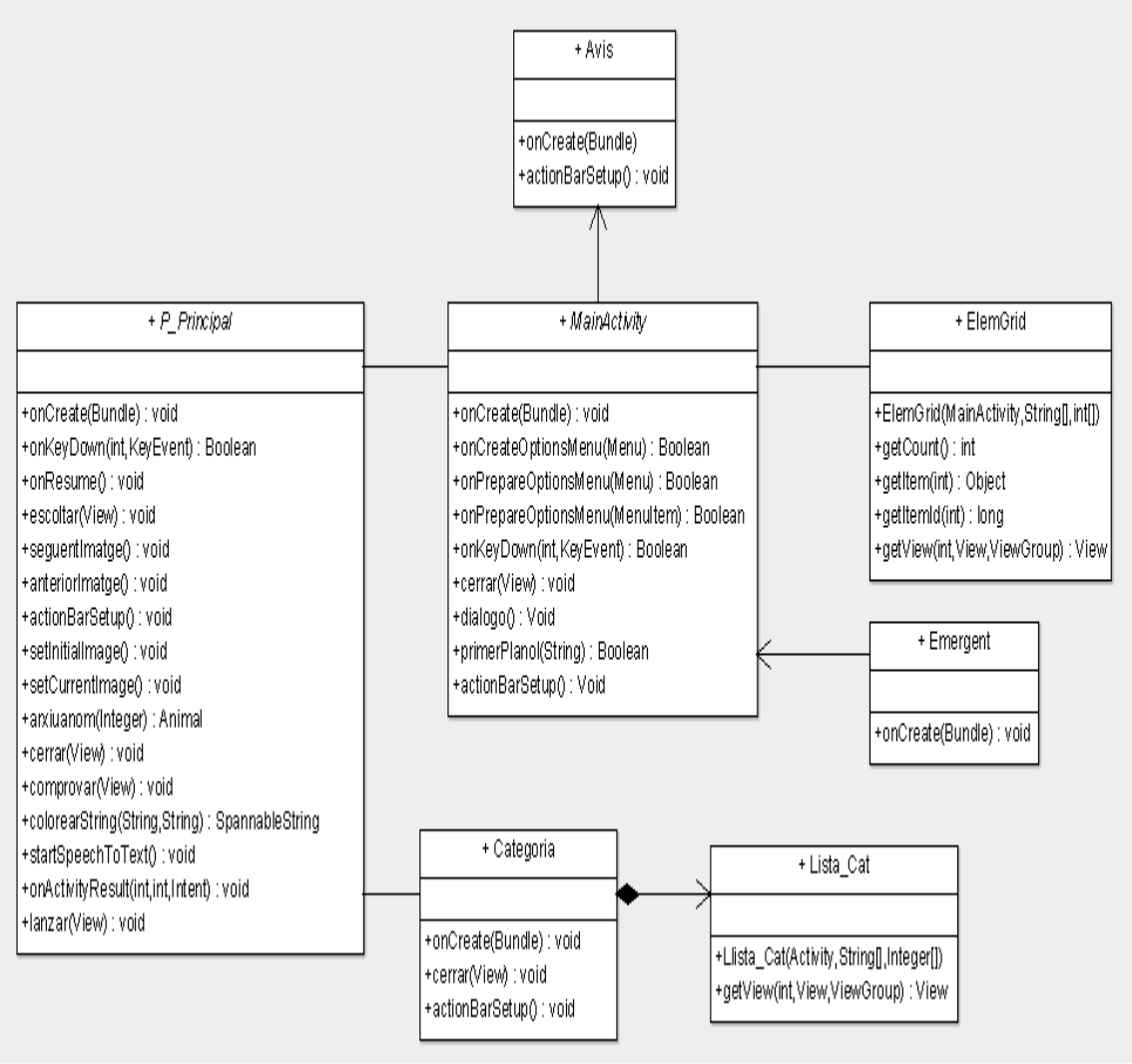
Identificador	CU-009
Nom	Veure Categoria
Prioritat	Normal
Descripció	L'usuari selecciona el botó 'Categoria' que hi ha al costat del botó 'Altaveu' a la pantalla principal
Actors	Usuari (nens)
Pre-Condicions	S'ha triat una paraula
Iniciat per	Botó 'Categoria'
Flux	Inici → Tria Categoria → Tria Paraula → Veure Categoria
Post-Condicions	Es mostra un llistat de la categoria corresponent, amb les imatges de cada paraula
Notes	--

Identificador	CU-010
Nom	Avís de temps excessiu amb el mòbil
Prioritat	Normal
Descripció	Al cap d'una hora d'estar a l'App, es mostra un avís aconsellant sortir de l'App
Actors	Usuari (nens)
Pre-Condicions	S'ha entrat a l'App
Iniciat per	Usuari
Flux	Inici → Avís de temps excessiu amb el mòbil
Post-Condicions	Es mostra un avís que ens informa que fa una hora que estem a l'App
Notes	--

3.2 – DISSENY DE L'ARQUITECTURA

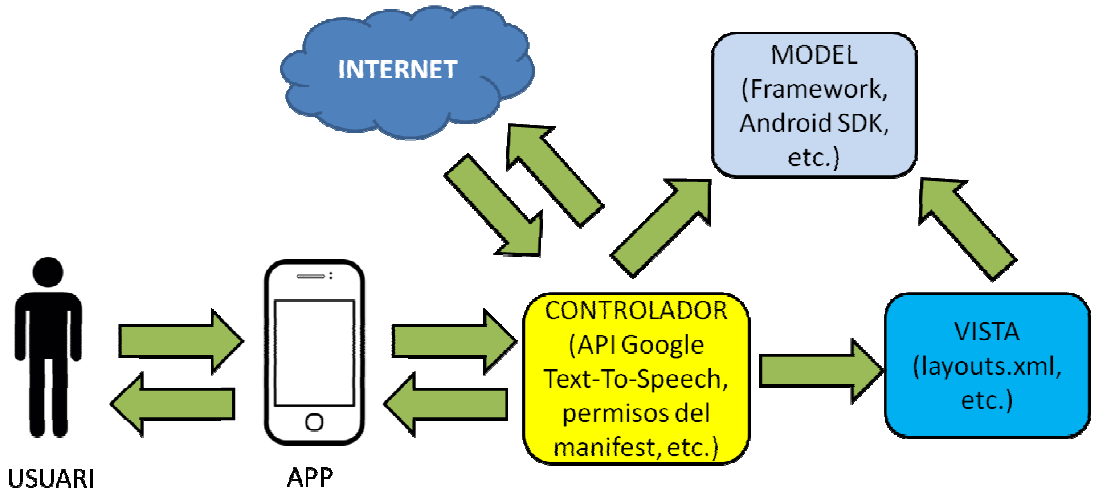
L'App que he implementat no té base de dades, per tant, no es mostra cap diagrama UML corresponent al disseny de la base de dades.

En canvi, a la següent imatge, es mostra un diagrama UML corresponent al disseny de les entitats i classes:



Al següent apartat es veurà el que he fet a cadascuna d'aquestes classes per tal de que l'App dugui a terme les funcionalitats indicades als apartats anteriors.

A continuació, es pot veure un diagrama (MODEL-VISTA-CONTROLADOR) on es mostra l'arquitectura del sistema:



4 – IMPLEMENTACIÓ

L'aplicació l'he desenvolupada amb l'Android Studio, ja que és l'eina oficial per a desenvolupar Android. Per al reconeixent de veu he emprat l'API de reconeixement de veu de Google.

A continuació es detalla com he implementat cadascuna de les funcionalitats que abans he descrit a la definició dels casos d'ús i es mostren imatges de l'App realitzant cada funció. També es descriu com he implementat alguns altres aspectes.

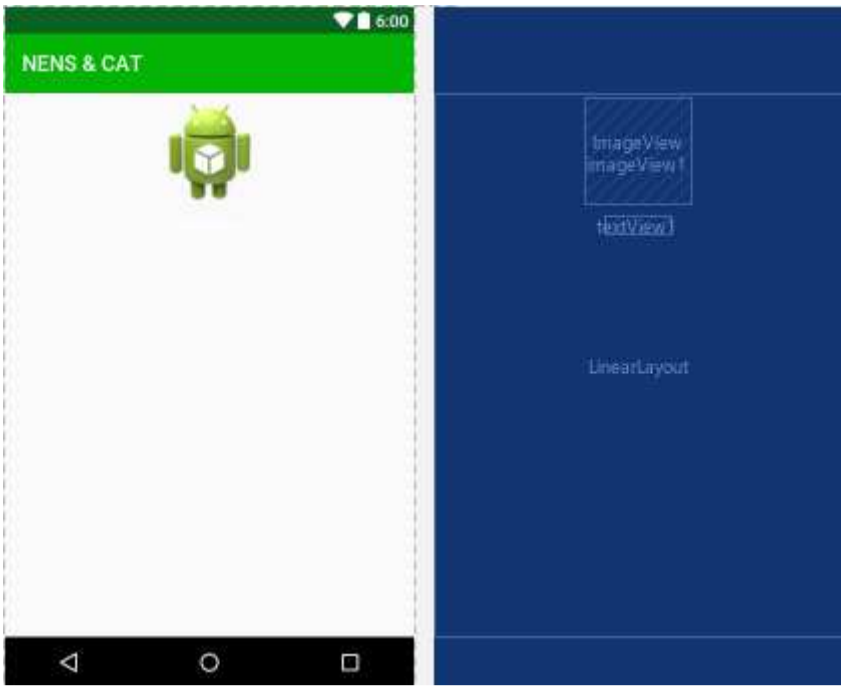
4.1 – TRIAR CATEGORIA

Per a poder triar la categoria al menú principal, bàsicament, he implementat el següent:
Al `onCreate` de la classe `MainActivity.java`, inicialitzem el GridView amb la llista de noms de categories i una llista d'imatges de categories:

```
gv=(GridView) findViewById(R.id.gridView1);
gv.setAdapter(new ElemGrid(this, catNomList,catImatges));
```

La classe `ElemGrid.java` posa cada nom de categoria (a un 'textView') amb la seva corresponent imatge (a un 'imageView'), per això, es crida al layout `llista_grid`.

Imatge de llista_grid.xml:

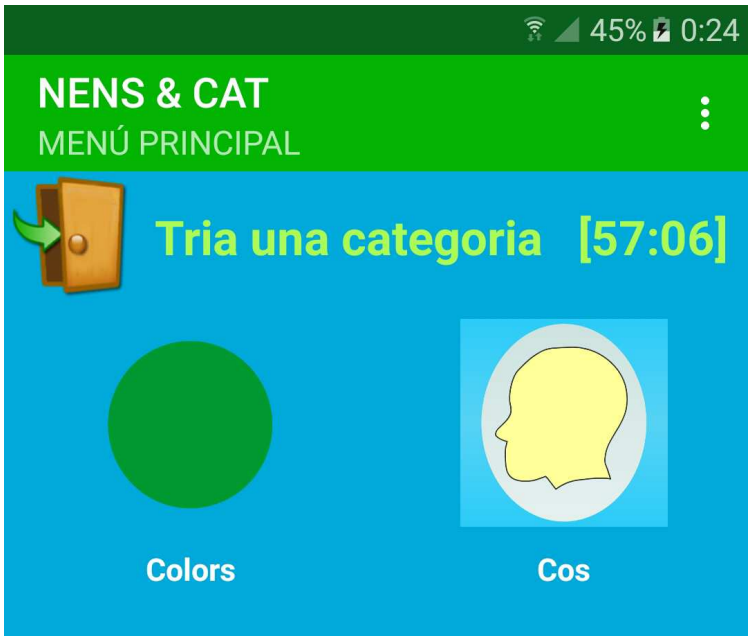


A continuació es mostra com s'ha implementat 'ElemGrid.java':

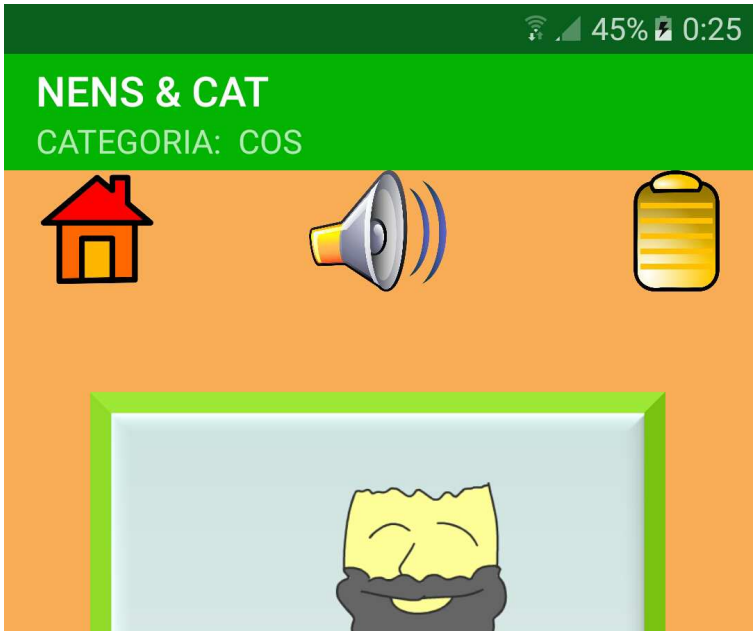
```
@Override
public View getView(final int position, View convertView, ViewGroup
parent) {
    // TODO Auto-generated method stub
    Holder holder=new Holder();
    View rowView;

    rowView = inflater.inflate(R.layout.llista_grid, null);
    holder.textview1=(TextView) rowView.findViewById(R.id.textView1);
    holder.imageview1=(ImageView)
rowView.findViewById(R.id.imageView1);

    holder.textview1.setText(noms[position]);
    holder.imageview1.setImageResource(imatgeId[position]);


    rowView.setOnClickListener(new OnClickListener() {

        @Override
        public void onClick(View v) {
            // TODO Auto-generated method stub

            //cream l'intent
            Intent intent = new Intent(v.getContext(),
P_Principal.class);

            //enviam el nom del element
            intent.putExtra("message", noms[position]);

            //iniciem l'activitat 'P_Principal'
            context.startActivity(intent);
        }
    });

    return rowView;
}
```

Com es pot veure, al fer click sobre qualsevol element del GridView, enviem a la classe P_Principal.java (pantalla principal) el nom del element a través d'un 'Intent'. Finalment iniciem la pantalla principal.

A la següent imatge es pot veure el menú principal amb el GridView i com seleccionem una categoria (en aquest cas, 'Cos'):



A la següent imatge es pot veure el la pantalla principal i, al ActionBar, es veu com hem anat a la categoria que em fet clic al menú principal:



Per tant, podem veure com aquesta funcionalitat es comporta de manera correcta.

4.2 – PASSAR A LA PARAULA ANTERIOR O SEGÜENT

Una vegada que em triat la categoria, a la pantalla principal, es mostra la primera paraula de cada categoria. Però si volem passar a la paraula anterior o següent de la categoria, he implementat el que es veurà a continuació.

Es pot passar a la paraula anterior o següent pitjant el botó 'endavant' o el botó 'endarrera', però també he fet que es pugui passar a la paraula anterior o següent amb el dit.
Per a poder passar les imatges amb el dit, al onCreate de la classe P_Principal.java, he implementat:

```
public int currImage = 0;
static final int MIN_DISTANCE = 250;

...
onCreate
...

imageView = (ImageView)findViewById(R.id.imageDisplay);


imageView.setOnTouchListener(new OnTouchListener() {
    public boolean onTouch(View v, MotionEvent event) {
        switch(event.getAction())
        {
            case MotionEvent.ACTION_DOWN:
                x1 = event.getX();
                break;

            case MotionEvent.ACTION_UP:

                x2 = event.getX();
                float deltaX = x2 - x1;
                if (Math.abs(deltaX) > MIN_DISTANCE)
                {
                    if (x2 > x1)
                    {
                        //d'esquerra a dreta
                        currImage--;
                        if (currImage == -1) {
                            currImage = images.length-1;
                        }

                        // Iniciam l'animació de l'imatge

imageView.startAnimation(AnimationUtils.loadAnimation(getApplicationCo
ntext(), R.anim.slide));

                        setCurrentImage();
                        ((CheckBox)
findViewById(R.id.checkBox)).setChecked(false);

((TextView)findViewById(textView2)).setVisibility(View.INVISIBLE);
                        EditText editText =
(EditText)findViewById(R.id.editText);
                        editText.setText("");
                        //guardam la current imatge a
```

```
                    sharedpreferences i ho enviem
                                SharedPreferences.Editor editor =
getSharedPreferences(MY_PREFS_NAME, MODE_PRIVATE).edit();
                                editor.putString("key",
String.valueOf(currImage));
                                editor.commit();
                            }
                            else
                            {
                                //de dreta a esquerra
                                currImage++;
                                if (currImage == images.length) {
                                    currImage = 0;
                                }

                                // Iniciam l'animació de l'imatge

imageView.startAnimation(AnimationUtils.loadAnimation(getApplicationCo
ntext(), R.anim.slide2));

                                setCurrentImage();
                                ((CheckBox)
findViewById(R.id.checkBox)).setChecked(false);

((TextView)findViewById(textView2)).setVisibility(View.INVISIBLE);
                                EditText editText =
(EditText)findViewById(R.id.editText);
                                editText.setText("");
                                //guardam la current imatge a
sharedpreferences i ho enviem
                                SharedPreferences.Editor editor =
getSharedPreferences(MY_PREFS_NAME, MODE_PRIVATE).edit();
                                editor.putString("key",
String.valueOf(currImage));
                                editor.commit();
                            }
                        }
                        else
                        {

                        }
                        break;
                }
                return true;
            }
        });
```

Com es pot veure, per implementar aquesta manera de passar imatges amb el dit, ens basem en la distància entre 2 punts (si és major a la variable ***MIN_DISTANCE***) i si un punt es major que l'altra per anar d'esquerra a dreta o de dreta a esquerra.

L'altra manera, de passar a l'imatge anterior o següent amb botons, l'he feta creant un mètode per a passar a l'imatge anterior (que es crida al pitjar el botó 'anterior') i un altra mètode per a passar a l'imatge següent (que es crida al pitjar el botó 'següent'):

```
private void seguentImatge() {
    final ImageButton botoseg = (ImageButton)
findViewById(R.id.imageButton5);
    botoseg.setOnClickListener(new OnClickListener() {
        @Override
        public void onClick(View arg0) {

            currImage++;
            if (currImage == images.length) {
                currImage = 0;
            }

            // Iniciem l'animació de l'imatge

imageView.startAnimation(AnimationUtils.loadAnimation(getApplicationCo
ntext(), R.anim.slide2));

            setCurrentImage();
            ((CheckBox)
findViewById(R.id.checkBox)).setChecked(false);

((TextView)findViewById(textView2)).setVisibility(View.INVISIBLE);
            EditText editText = (EditText)findViewById(R.id.editText);
            editText.setText("");
            //guardam la current imatge a sharedpreferences i ho
enviem
            SharedPreferences.Editor editor =
getSharedPreferences(MY_PREFS_NAME, MODE_PRIVATE).edit();
            editor.putString("key", String.valueOf(currImage));
            editor.commit();
        }
    });
}



private void anteriorImatge() {
    final ImageButton botoant = (ImageButton)
findViewById(R.id.imageButton4);
    botoant.setOnClickListener(new OnClickListener() {
        @Override
        public void onClick(View arg0) {

            currImage--;
            if (currImage == -1) {
                currImage = images.length-1;
            }

            // Iniciem l'animació de l'imatge

imageView.startAnimation(AnimationUtils.loadAnimation(getApplicationCo
ntext(), R.anim.slide));

            setCurrentImage();
            ((CheckBox)
findViewById(R.id.checkBox)).setChecked(false);

((TextView)findViewById(textView2)).setVisibility(View.INVISIBLE);
            EditText editText = (EditText)findViewById(R.id.editText);
            editText.setText("");
```

```
            //guardam la current imatge a sharedpreferences i ho
enviem
            SharedPreferences.Editor editor =
getSharedPreferences(MY_PREFS_NAME, MODE_PRIVATE).edit();
            editor.putString("key", String.valueOf(currImage));
            editor.commit();
        }
    });
}
```

Com podem veure, tant a la manera de passar les imatges amb el dit com amb els botons, per passar a l'imatge anterior restem a currImage i per passar a l'imatge següent sumem a currImage. També controlo que si estem a la darrera imatge i passem a la següent, estarem a la primera imatge i, pel contrari, si estem a la primera imatge i passem a l'anterior, estarem a la darrera imatge.

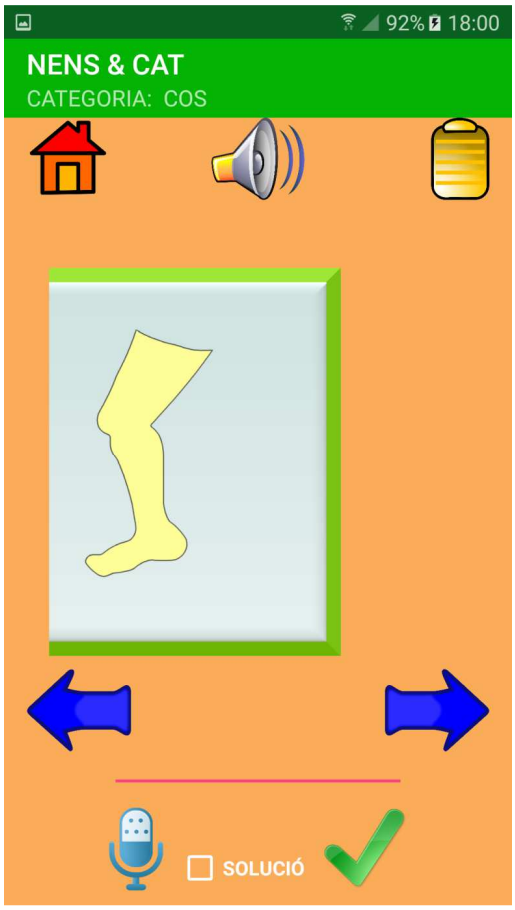
Quan es passa a l'imatge anterior o següent, el checkBox de la pantalla principal (el qual seleccionem per veure la paraula) es desselecciona i el textView on es veu la paraula s'oculta. Faig això perquè puc haver vist la solució d'una paraula en concret, però a lo millor no vull veure la solució de la paraula anterior o següent.

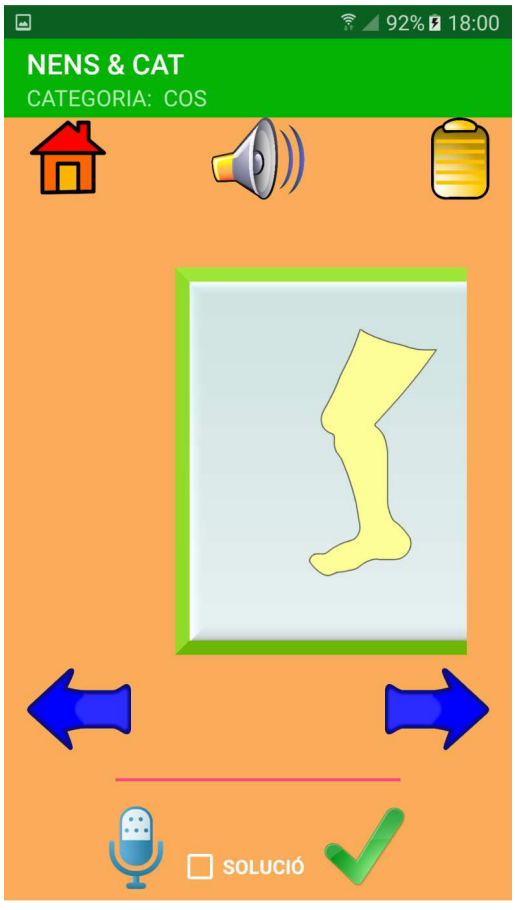
Al canviar d'imatge, també buidem l' (per si abans havíem dit o escrit la paraula) i guardem l'imatge actual a sharedpreferences, i així tenim guardat l'estat actual.
Per acabar, dir que al canviar d'imatge, s'inicia una animació de translació, en la que l'imatge anterior o següent apareix desde l'esquerra o dreta de la pantalla fins al centre d'aquesta:

```
<set
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:interpolator="@android:anim/linear_interpolator"
    android:fillAfter="true">

    <translate
        android:fromXDelta="-100%p"
        android:toXDelta="0%p"
        android:duration="1000" />
</set>
```

Aquesta animació és diferent segons si es passa a l'imatge anterior o a la següent (només canvia la línea **android:fromXDelta="-100%p"** a **android:fromXDelta="100%p"**) .

A les imatges següents es veu com es passa d'una imatge a l'anterior i a la següent. Com podem veure, aquesta funcionalitat es comporta de manera correcta:


92% 18:00
NENS & CAT
CATEGORIA: COS
SOLUCIÓ


92% 18:00
NENS & CAT
CATEGORIA: COS
SOLUCIÓ

4.3 – VEURE CATEGORIA

He afegit un botó 'categoria' per poder veure totes paraules (i respectives imatges) de la categoria on estem. Es mostra en forma de llista i podem anar directament a una paraula pitjant damunt el respectiu ítem de la llista. D'aquesta manera, es millora molt la navegació entre paraules, ja que sinó s'hauria de pitjar el botó 'anterior' o 'següent' (o passar imatges amb el dit) moltes vegades (si la categoria és llarga) fins a arribar a la paraula. He implementat aquesta funcionalitat fent que quan es pitja el botó 'categoria' a la pantalla principal es crida al mètode 'llançar' de 'P_Principal.java':

```
public void lanzar(View view) {

    Bundle b=new Bundle();

    Intent intent2 = new Intent(this, Categoria.class);

    Intent intent = getIntent();
    String message = intent.getStringExtra("message");

    if (message != null && message.contentEquals("Colors")) {
        ArrayList<Integer> arraycolors = new ArrayList<Integer>();
        b.putIntegerArrayList("images", arraycolors);
        for (int i=0; i<colors.length; i++){
            arraycolors.add(colors[i]);
        }
        intent2.putExtras(b);
    }

    if (message != null && message.contentEquals("Cos")) {
        ArrayList<Integer> arraycos = new ArrayList<Integer>();
        b.putIntegerArrayList("images", arraycos);
        for (int i=0; i<cos.length; i++){
            arraycos.add(cos[i]);
        }
        intent2.putExtras(b);
    }

    //enviem també en nom de la categoria, per posar al
actionBarSetup()
    intent2.putExtra("message", message);

    startActivity(intent2);
    EditText editText = (EditText)findViewById(R.id.editText);
    editText.setText("");
}
```

La funció del mètode 'lanzar' és que, segons el valor del missatge de l'intent (que s'envia al triar la categoria al menú principal), s'envia el contingut imatges d'una categoria o una altra dins un ArrayList a l'activitat 'Categoria.java' a través d'un 'Intent' i s'inicia l'activitat 'Categoria'. Al codi de dalt, a manera d'exemple, només he posat dues categories, ja que sinó el codi era llarg.

Després, a 'Categoria.java' obtenim l'informació enviada (ArrayList) desde l'activitat 'P_Principal' i assignem un nom a cada element del Array (en aquest cas, és el mateix text a cada element). Tot seguit cream un objecte 'List_Cat' per a cada element de la categoria i ho

posam a una llista. Per acabar, assignem les accions al fer 'click' a un dels elements de la llista (ListView del layout 'categoria.xml'). Al fer 'click' s'envia a SharedPreferences l'element seleccionat (posició) de la llista d'elements de la categoria, s'inicia una animació i es tanca l'activitat.

El codi de 'Categoria.java' que realitza el que s'acaba d'explicar és:

```
protected void onCreate(Bundle savedInstanceState) {
   super.onCreate(savedInstanceState);
   setContentView(R.layout.categoria);

   //obtenim l'informació enviada (ArrayList) desde l'activitat
'P_Principal'
   Bundle b=getIntent().getExtras();
   array=b.getIntegerArrayList("images");

   Integer[] elem = array.toArray(new Integer[array.size()]);

   //Assignem um nom a cada element, en aquest cas, el text 'Anar-hi'
   itemname = new String [elem.length];
   for (int i=0; i<elem.length; i++){
      itemname[i] = "  Anar-hi";
   }

   //cream un objecte 'List_Cat' per a cada element de la categoria
   Llista_Cat elemCat=new Llista_Cat(this, itemname, elem);
   list=(ListView)findViewById(R.id.list);
   list.setAdapter(elemCat);

   list.setOnItemClickListener(new OnItemClickListener() {

      @Override
      public void onItemClick(AdapterView<?> parent, View view,
                        int position, long id) {
         // TODO Auto-generated method stub
         //enviem a sharedpreferences l'element seleccionat de la
llista d'elements d'una categoria
         SharedPreferences.Editor editor =
getSharedPreferences(MY_PREFS_NAME, MODE_PRIVATE).edit();
         editor.putString("key", String.valueOf(position));
         editor.commit();
         Animation animation1 = new AlphaAnimation(0.3f, 1.0f);
         animation1.setDuration(4000);
         view.startAnimation(animation1);
         cerrar(view);
      }
   });

}
```

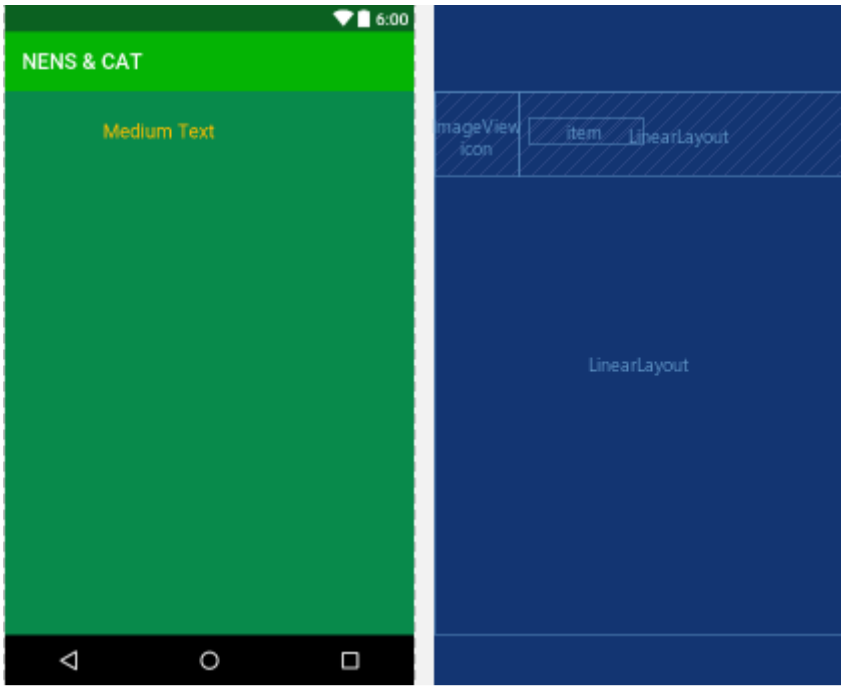
La classe 'Llista_Cat' és la següent:

```
public Llista_Cat(Activity context, String[] itemname, Integer[]
imgid) {
   super(context, R.layout.llistacat, itemname);
   // TODO Auto-generated constructor stub
```

```
    this.context=context;
    this.itemname=itemname;
    this.imgid=imgid;
}

public View getView(int position,View view,ViewGroup parent) {
    LayoutInflater inflater=context.getLayoutInflater();
    View rowView=inflater.inflate(R.layout.llistacat, null,true);

    TextView txtTitle = (TextView) rowView.findViewById(R.id.item);
    ImageView imageView = (ImageView) rowView.findViewById(R.id.icon);

    //per a cada element de la llista (elements de la categoria) es
mostra la imatge corresponent (imageView) i un text
    txtTitle.setText(itemname[position]);
    imageView.setImageResource(imgid[position]);
    return rowView;

};
```

Com es pot veure, cada objecte de la classe 'Llista_Cat.java' és una fila del ListView de la categoria. Per això, es crida al layout 'llistacat.xml' que conté un ImageView i un TextView (ja que es mostra una imatge i un text per a cada element) posat en una fila.

Imatge de 'llistacat.xml':



Per acabar aquesta funcionalitat, és important parlar del mètode '`onResume()`' de la classe 'P_Principal.java' on recollim l'informació de SharedPreferences que s'ha enviat desde 'Categoria.java' i posem com a imatge actual 'currImage' (i segons el valor del missatge de l'intent que s'envia al triar la categoria al menú principal) l'element seleccionat a la llista de la categoria:

```
@Override
public void onResume()
{
    // 1. obtenim l'intent del MainActivity
    Intent intent = getIntent();

    // 2. obtenim el valor del missatge de l'intent
    String message = intent.getStringExtra("message");

    super.onResume();

    // obtenim de sharedpreferences l'element seleccionat a
l'activitat Categoria,
    // mostrarem a 'P_Principal' l'element que em seleccionat de la
llista d'elements de la categoria a l'activitat 'Categoria'
    SharedPreferences prefs = getSharedPreferences(MY_PREFS_NAME,
MODE_PRIVATE);
    String elemCat = prefs.getString("key", "0");
    ImageView imageView = (ImageView) findViewById(R.id.imageDisplay);

    if (message != null && message.contentEquals("Colors")) {
        imageView.setImageResource(images[Integer.parseInt(elemCat)]);
        currImage=Integer.parseInt(elemCat);
        setCurrentImage();
        ((CheckBox) findViewById(R.id.checkBox)).setChecked(false);

((TextView)findViewById(textView2)).setVisibility(View.INVISIBLE);
    }
    else if (message != null && message.contentEquals("Cos")){
        imageView.setImageResource(images[Integer.parseInt(elemCat)]);
        currImage=Integer.parseInt(elemCat);
        setCurrentImage();
        ((CheckBox) findViewById(R.id.checkBox)).setChecked(false);

((TextView)findViewById(textView2)).setVisibility(View.INVISIBLE);
    }
}
```

A ‘onResume()’ també ocultem el TextView on es mostra la paraula i desseleccionem el CheckBox que es marca per veure la paraula (solució).

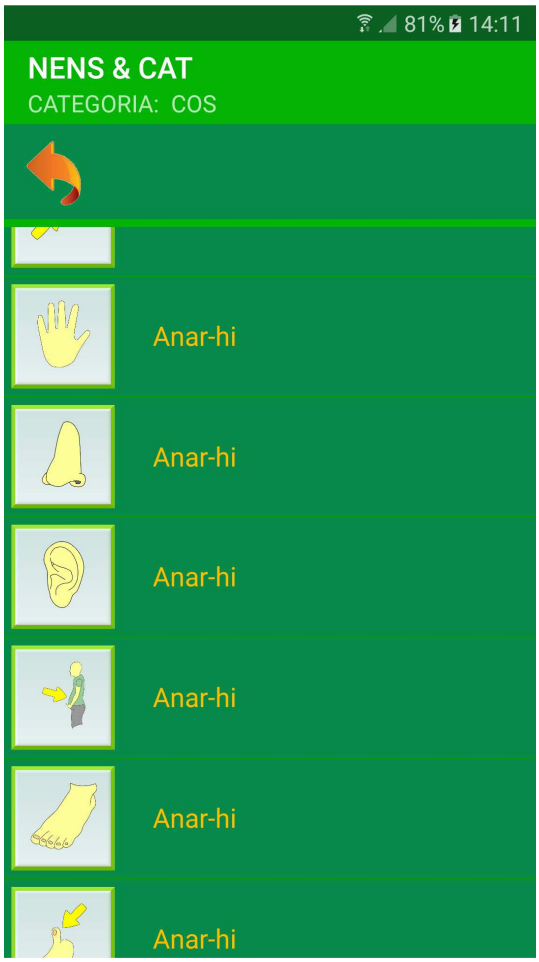
IMPORTANT. Faig servir SharedPreferences en lloc d’un intent ja que m’interessava fer servir alternatives als intents per passar informació entre activitats i, d’aquesta manera, practicar amb més opcions d’Android.

Amb SharedPreferences no he tancat en cap moment l’Activitat ‘P_Principal’ (en lloc de tancar-la desde el mètode ‘lanzar’ de ‘P_Principal’ i després fer iniciar-la de nou amb startActivity a través d’un intent a la classe ‘Categoria.java) al anar a ‘Categoria’, sinó que el que faig és anar ‘actualitzant’ la pantalla principal (‘P_Principal.java’) segons el SharedPreferences i ‘currImage’. Per això, al passar a l’imatge anterior o següent també envio a SharedPreferences l’imatge actual ‘currImage’.

És important dir que he poso el valor '0' a SharedPreferences abans de sortir de l'activitat 'P_Principal.java' (pantalla principal) per a que, s'inicii amb la primera imatge al triar una categoria. Per això, al pitjar el botó 'Sortir' o el botó 'back' del mòbil, faig que s'executi el següent codi:

```
//posem 0 (primera imatge) sharedpreferences
SharedPreferences.Editor editor = getSharedPreferences(MY_PREFS_NAME,
MODE_PRIVATE).edit();
editor.putString("key", String.valueOf(0));
editor.commit();
```

A continuació es mostra la llista d'una categoria:



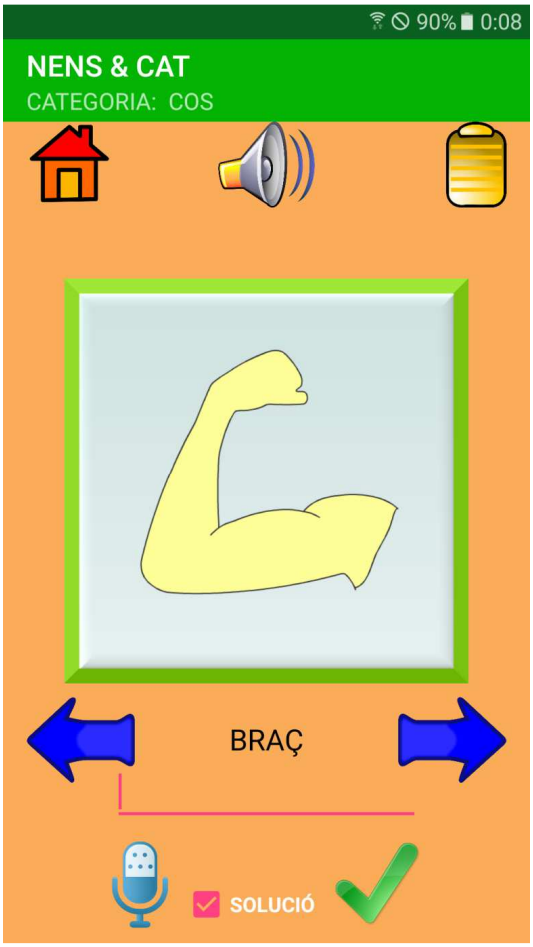
Podem veure com es mostra correctament un text i l'imatge de cada element de la llista.

4.4 – VEURE SOLUCIÓ (PARAULA)

Es tracta de poder ocultar/mostrar la paraula mitjançant un 'CheckBox'. Per implementar aquesta funcionalitat, al onCreate de la classe P_Principal.java, he fet:

```
final CheckBox checkbox = (CheckBox) findViewById(R.id.checkBox);
checkbox.setOnClickListener(new OnClickListener() {
    public void onClick(View v) {
        //Si el checkbox es selecciona o no, es veu la paraula o no
        if (((CheckBox) v).isChecked()) {

((TextView)findViewById(textView2)).setVisibility(View.VISIBLE);
        } else {

((TextView)findViewById(textView2)).setVisibility(View.INVISIBLE);
        }
    }
});
```

Per tant, fem visible o no el textView segons estigui o no seleccionat el CheckBox.



A l'imatge anterior es veu com, amb el CheckBox seleccionat, es veu la paraula. A més, a altres imatges anteriors que he posat, podem veure que amb el CheckBox desseleccionat, no es veu la paraula . Per tant, podem dir que la funcionalitat es comporta correctament.

4.5 – ESCOLTAR PARAULA

Es pot escoltar la paraula pitjant el botó 'altaveu' i també ('longClick') damunt l'imatge. Per a que s'escolti la paraula al pitjar el botó 'altaveu' he fet que, al pitjar dit botó, es cridi al següent mètode:

```
public void escoltar(View v) {
    MediaPlayer mp = MediaPlayer.create(P_Principal.this,
sonidos[currImage]);
    mp.start();
}
```

L'array 'sonidos' conté els àudios de cada paraula de la categoria. A aquest array s'hi posa uns sons o uns altres depenent de la categoria triada al menú principal:

```
...
onCreate
...

// 1. obtenim l'intent del MainActivity
Intent intent = getIntent();

// 2. obtenim el valor del missatge de l'intent
String message = intent.getStringExtra("message");

//posam el contingut d'uns arrays o uns altres (segons el valor del
missatge de l'intent) dins els arrays images i sonidos
if (message != null && message.contentEquals("Colors")) {
    images = new Integer[colors.length];
    sonidos = new Integer[colors.length];
    for (int x = 0; x < colors.length; x++) {
        images[x] = colors[x];
        sonidos[x] = socolors[x];
    }
}
else if (message != null && message.contentEquals("Cos")){
    images= new Integer[cos.length];
    sonidos = new Integer[cos.length];
    for (int x=0;x<cos.length;x++){
        images[x] = cos[x];
        sonidos[x] = socos[x];
    }
}
```

També es pot veure que fem el mateix amb l'array d'imatges.

Per a fer que s'escolti la paraula al fer 'longClick' damunt l'imatge, al 'onCreate()' de 'P_Principal.java':

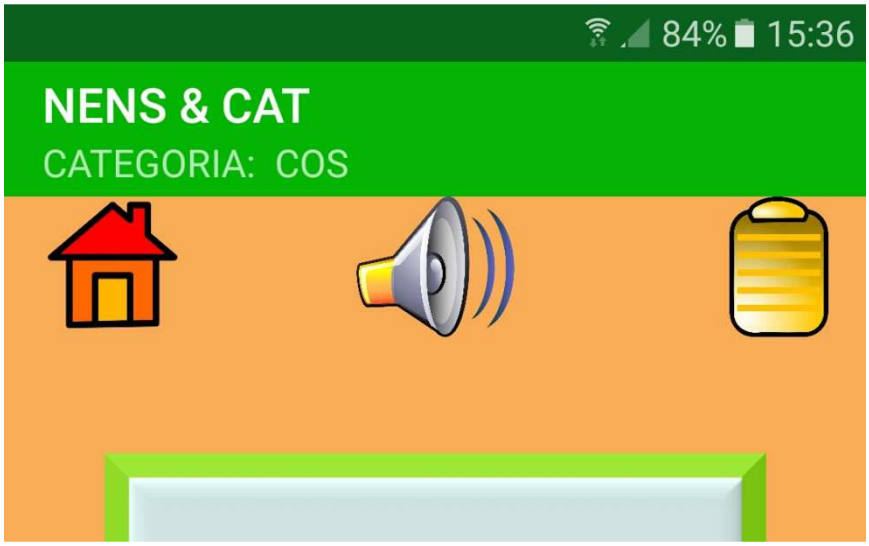
```
private long tempsClick;
private int longClickDuration = 600;

...
onCreate
...
```

```
imageView = (ImageView)findViewById(R.id.imageDisplay);


imageView.setOnTouchListener(new OnTouchListener() {
    public boolean onTouch(View v, MotionEvent event) {
        switch(event.getAction())
        {
            case MotionEvent.ACTION_DOWN:

                tempsClick = (long) System.currentTimeMillis();

                break;
            case MotionEvent.ACTION_UP:

                if ((System.currentTimeMillis() - tempsClick) >
longClickDuration) {
                    MediaPlayer mp =
MediaPlayer.create(P_Principal.this, sonidos[currImage]);
                    mp.start();
                }
                else
                {

                }
                break;
        }
        return true;
    }
});
```

Obtenim el temps que pitgem a l'imatge i, quan es major al temps que em posat a la variable 'longClickDuration', s'escolta la paraula.

IMPORTANT. He combinat aquest 'setOnTouchListener' amb el que ja havia mostrat més a dalt (per passar a l'imatge següent o anterior amb el dit).

A la següent imatge es veu el botó 'altaveu':



4.6 – DIR PARAULA

Pitjant al boto 'micròfon' podem dir la paraula i, mitjançant el reconeixent de veu, es veu el que em dit a un 'editText'. A continuació explico com s'ha implementat:

```
...
onCreate
...

txtOutput = (EditText) findViewById(R.id.editText);
microfon = (ImageButton) findViewById(R.id.imageButton6);

microfon.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        startSpeechToText();
    }
});
```

Per tant, a l'onCreate() fem que al pitjar el botó (ImageButton) 'micròfon' es crida al mètode 'startSpeechToText()', el qual està implementat de la següent manera:

```
private void startSpeechToText() {
    Intent intent = new
Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
    //posam l'idioma català
    intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE, "ca-ES");
    intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
            RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
    intent.putExtra(RecognizerIntent.EXTRA_PROMPT,
            "DIGUES LA PARAULA...");
    try {
        startActivityForResult(intent, SPEECH_RECOGNITION_CODE);
    } catch (ActivityNotFoundException a) {
        Toast.makeText(getApplicationContext(),
                "Ho sento! El reconeixement de veu no és compatible
amb aquest dispositiu.",
                Toast.LENGTH_SHORT).show();
    }
}

//la resposta del reconeixement de veu es posa al edittext
@Override
protected void onActivityResult(int requestCode, int resultCode,
Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    switch (requestCode) {
        case SPEECH_RECOGNITION_CODE: {
            if (resultCode == RESULT_OK && null != data) {

                ArrayList<String> result = data

.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
                String text = result.get(0);
                txtOutput.setText(text);
```

```
                }
                break;
            }

        }
}
```

A 'startSpeechToText()' he posat com a idioma el català. Per acabar, al mètode 'onActivityResult' es posa es resultat a l' 'txtOutput'.

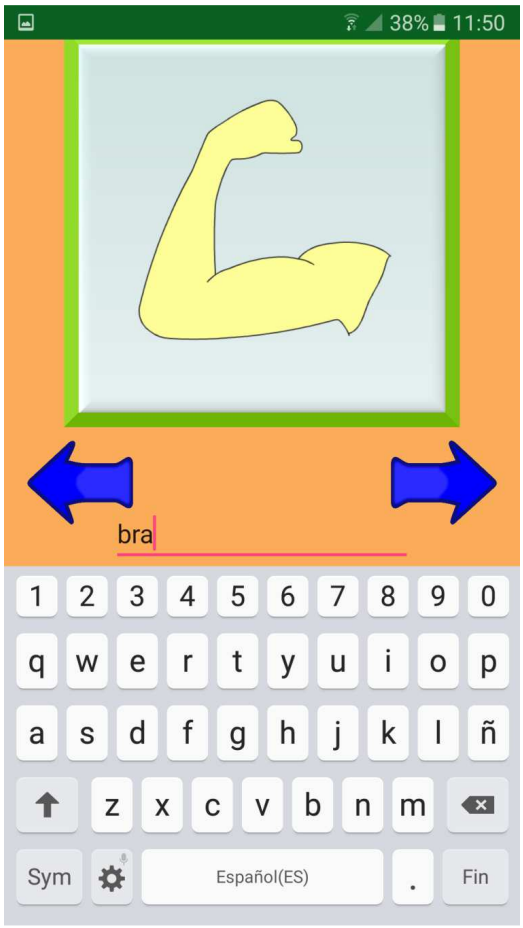
A continuació es mostra el que apareix al pitjar el botó 'micròfon:

4.7 – ESCRIURE PARAULA

Podem escriure la paraula amb el teclat a un EditText. l' és el mateix on també es mostra la paraula al fer servir el reconeixement de veu (mirar funcionalitat anterior):

txtOutput = (EditText) findViewById(R.id.***editText***);

La funcionalitat d'escriure la paraula l'ofereixen directament els EditText, no hem d'implementar res més, només tenir un EditText.



Com podem veure a l'imatge anterior, aquesta funcionalitat es comporta de manera correcta.

4.8 – COMPROVAR SI ÉS CORRECTE

Amb aquesta funcionalitat podem comprovar si el text que hi ha a l''EditText' és correcte o no (si és igual a com s'escriu la paraula en concret) i es mostra un text (un 'Toast') que diu si és correcte o no. També es mostra una imatge (durant uns segons) que diu si es correcte o no i s'escolta una veu que diu si és correcte o no. Per tant, he intentant mostrar si és correcte o no de vàries maneres perquè sigui més atractiu, intuïtiu i més clar pels nens. Tot seguit mostro com he implementat el mètode 'comprovar':

```
public void comprovar(View v){
    final ImageView imageView = (ImageView)
findViewById(R.id.imageDisplay);
    String solucio = this.arxiuanom(images[currImage]);
    EditText editText = (EditText)findViewById(R.id.editText);
    String meu = editText.getText().toString();

    //comparem el text que hi ha al edittext amb el nom de la paraula
corresponent
    if (meu.equalsIgnoreCase(solucio) == true ) {
        //es mostra un text que diu que és correcte
```

```
        Toast.makeText(this, "ÉS CORRECTE!",
Toast.LENGTH_SHORT).show();
        //s'escolta un so que diu que és correcte
        MediaPlayer mp = MediaPlayer.create(P_Principal.this,
R.raw.ok);
        mp.start();
        //es mostra una imatge 'OK' durant uns segons
        new CountDownTimer(3000, 1000) {

            public void onTick(long millisUntilFinished) {
                imageView.setImageResource(R.drawable.ok);
            }

            public void onFinish() {
                setCurrentImage();
            }
        }.start();

    }
    //comprovem si em introduït text a l'edittext
    else if (meu.equalsIgnoreCase("")==true){
        //es mostra un text que diu que no em introduït cap text
        Toast.makeText(this, "NO HAS INTRODUÏT CAP TEXT",
Toast.LENGTH_SHORT).show();
        MediaPlayer mp = MediaPlayer.create(P_Principal.this,
R.raw.notext);
        mp.start();
    }
    else {
        //es mostra un text que diu que no és correcte
        Toast.makeText(this, "NO ÉS CORRECTE. TORNA A PROVAR!",
Toast.LENGTH_SHORT).show();
        //s'escolta un so que diu que no és correcte
        MediaPlayer mp = MediaPlayer.create(P_Principal.this,
R.raw.nook);
        mp.start();
        //es mostra una imatge ' NO OK' durant uns segons
        new CountDownTimer(3000, 1000) {

            public void onTick(long millisUntilFinished) {
                imageView.setImageResource(R.drawable.nook);
                //mTextField.setText("seconds remaining: " +
millisUntilFinished / 1000);
            }

            public void onFinish() {
                setCurrentImage();
            }
        }.start();
        //si no és correcte, el mòbil vibra
        Vibrator vibe = (Vibrator)
getSystemService(this.VIBRATOR_SERVICE);
        vibe.vibrate(200);
    }
    SpannableString spnMsg = colorearString(meu, solucio);
    editText.setText(spnMsg, BufferType.SPANNABLE);
}
```

Com es pot veure, es compara el text que hi ha al EditText amb el nom de la paraula corresponent i es mostra un 'Toast', un àudio i una imatge que seran diferents depenent de si són iguals o no.

També podem veure al codi que, si no hi ha res escrit al 'EditText', es mostra un text (un 'Toast') que posa que no s'ha res introduït text i una veu que diu que s'ha d'introduir text abans de comprovar. Finalment, si no és correcte el mòbil vibra.

El nom de la paraula l'aconseguim gràcies al mètode 'arxiuanom', en el qual obtenim el nom del arxiu (d'imatge) de la paraula corresponent:

```
public String arxiuanom(Integer image){

    //Les 4 següents línies són per obtenir el nom del arxiu de
l'imatge
    CharSequence path = getText(image);
    String[] tokens = path.toString().split("[\\\\|/]");
    String filename = tokens[tokens.length - 1];
    String nom = filename.substring(0,filename.lastIndexOf("."));

    nom = nom.replace('0', 'à');
    nom = nom.replace('1', 'è');
    nom = nom.replace('2', 'é');
    nom = nom.replace('3', 'í');
    nom = nom.replace('4', 'ï');
    nom = nom.replace('5', 'ò');
    nom = nom.replace('6', 'ó');
    nom = nom.replace('7', 'ú');
    nom = nom.replace('8', 'ü');
    nom = nom.replace('9', 'ç');
    nom = nom.replace('_', ' ');

    return nom;
}
```

Un nom d'arxiu no pot tenir accent, etc., només caràcters de 'a-z', '0-9' i '_'. Per tant, faig les substitucions que es veuen a dalt.

A més, les línies:

```
SpannableString spnMsg = colorearString(meu, solucio);
editText.setText(spnMsg, BufferType.SPANNABLE);
```

són perquè es mostren en verd les lletres que em escrit bé i en vermell les que no. Això ho aconseguim amb el mètode 'colorearString':

```
public SpannableString colorearString(String myStr, String soluc)
{
    //si un string és més llarg que l'altra, afegeixen zeros al string
més curt fins que són iguals
    if (myStr.length() > soluc.length()) {
        int dif = myStr.length() - soluc.length();
        for(int i=0; i < dif; i++){
            soluc = soluc.concat("0");
        }
    }
```

```
    SpannableString spnStr=new SpannableString(myStr);
    int strLen=myStr.length();

    for(int i=0; i < strLen; i++)
    {
        if (myStr.toUpperCase().charAt(i) ==
soluc.toUpperCase().charAt(i)) //per a que sigui igual escriure
minúscules o majúscules al edittext(així no distingeix entre
majúscules i minúscules).

// En aquest cas, he passat els 2 strings a majúscules
        {
            spnStr.setSpan(new ForegroundColorSpan(Color.GREEN), i, i
+ 1, 0);
        }
        else
        {
            spnStr.setSpan(new ForegroundColorSpan(Color.RED), i, i +
1, 0);
            spnStr.setSpan(new
android.text.style.StyleSpan(android.graphics.Typeface.BOLD), i, i +
1, 0);
        }
    }
    return spnStr;
}
```

A aquest mètode li passem 2 strings i els compara (en el nostre cas, el text que hi ha al EditText i el nom del arxiu de l'imatge de la paraula corresponent). També tenim en compte si un string és més llarg que l'altra, afegint zeros al string més curt fins tenir els dos string la mateixa llargària. Després recorrem caràcter a caràcter els dos strings i les lletres diferents les 'acoloreix' en vermell i les iguals en verd.

A continuació es mostra una imatge de quan comprovem i és correcte i una altra imatge de quan és incorrecte:


75% 19:55
NENS & CAT
CATEGORIA: COS
CORRECTE ! ! !
CAP
cap
ÉS CORRECTE!
SOLUCIÓ


76% 20:01
NENS & CAT
CATEGORIA: COS
NO ÉS CORRECTE ! ! !
CAP
TORNA A PROVAR!
SOLUCIÓ

4.9 – OCULTAR 'VEURE SOLUCIÓ'

Al menú principal he posat un menú 'overflow' al ActionBar amb un ítem 'checkeable' que permet ocultar o mostrar el CheckBox de la pantalla principal, el qual seleccionem per a veure la paraula. Ho he fet perquè no sigui tan fàcil poder veure les solucions pels nens i per oferir l'opció pels que no vulguin l'ajuda de poder consultar la solució.

Al 'MainActivity.java' he fet:

```
public static Boolean isChecked = false;

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is
present.
    getMenuInflater().inflate(R.menu.menu_main, menu);
    return true;
}

public boolean onPrepareOptionsMenu(Menu menu) {
    MenuItem checkable = menu.findItem(R.id.solucions);
    checkable.setChecked(isChecked);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem ítem) {
switch(item.getItemId()) {

    case R.id.solucions:
        check = !item.isChecked();
        item.setChecked(check);
        if (check==false){
            Toast.makeText(this, R.string.soluc_toast_NOcheck,
Toast.LENGTH_LONG).show();
            isChecked = false;
        }
        if (check==true){
            Toast.makeText(this, R.string.soluc_toast_check,
Toast.LENGTH_LONG).show();
            isChecked = true;
        }
        return(true);
    }
    return(super.onOptionsItemSelected(ítem));
}
```

Si l'ítem del menú no està seleccionat, la variable 'isChecked'=false i si està seleccionat 'isChecked'=true. També es mostra un 'Toast' diferent segons estigui o no seleccionat.

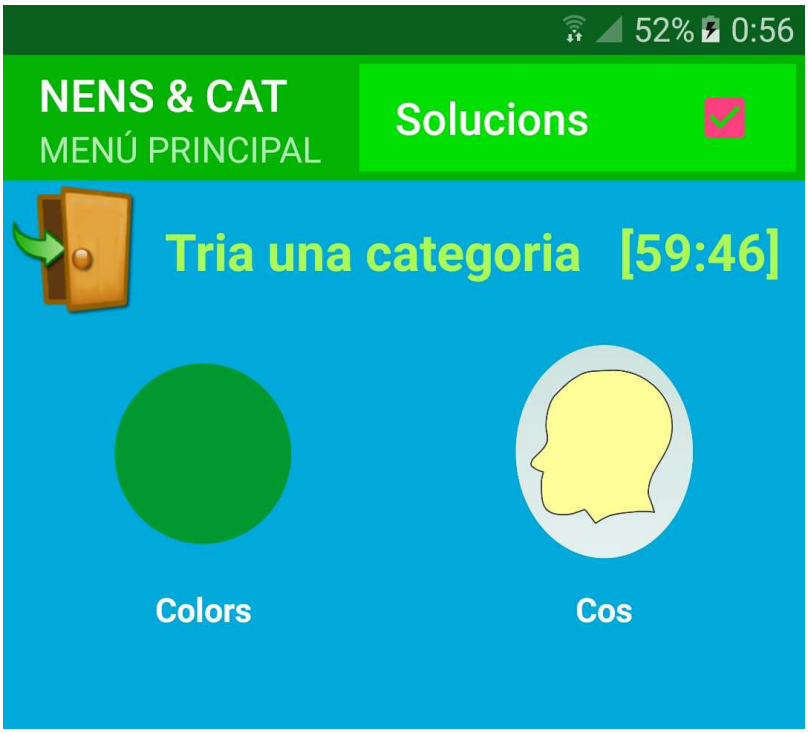
A 'P_Principal.java' he fet:

```
boolean check = MainActivity.isChecked;

if (check==false){
    checkbox.setVisibility(View.INVISIBLE);
```

```
}
if (check==true){
    checkbox.setVisibility(View.VISIBLE);
}
```

Obtinc la variable 'isChecked' de MainActivity i fem visible el CheckBox segons si 'isChecked' es false o true.

A la següent imatge es veu el menú:



4.10 – AVÍS DE TEMPS EXCESSIU AMB EL MÒBIL

Amb aquesta funcionalitat, al cap d'un temps (una hora) d'estar a l'App, surt un avís recordant al nen que no és bo estar molt de temps al mòbil i recomanant-l'hi que faci altres activitats. He fet aquesta funcionalitat perquè em pareix important que els usuaris de l'App (majoritàriament nens) se'n adonin que no han d'estar molt de temps amb el mòbil.

Per a dur a terme aquesta funcionalitat, al 'MainActivity.java' he implementat:

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);


    cT =  new CountDownTimer(3600000, 1000) {

        public void onTick(long millisUntilFinished) {
```

```
            String v = String.format("%02d",
millisUntilFinished/60000);
            int va = (int)( (millisUntilFinished%60000)/1000);
            ((TextView)findViewById(R.id.textView)).setText("Tria una
categoria   "+"[" +v+":"+String.format("%02d",va)+"]");
        }

        public void onFinish() {
            //es comprova si estem a l'App. Si ja em sortit de l'App,
no s'ha de mostrar l'avís
            if (primerPlanol(getPackageName())!=false) {
                ((TextView)
findViewById(R.id.textView)).setText("Convé deixar el mòbil");
                Intent mainIntent = new
Intent().setClass(MainActivity.this, Avis.class);
                startActivity(mainIntent);
            }
        }
    };
    cT.start();

}
```

Quan s'inicia l'activitat s'inicia un cronòmetre de temps enredera (també es mostra un TextView que va mostrant el temps) i, al acabar, s'inicia l'activitat Avís, la qual mostra un avís.

Si estem a l'App, l'avís no s'ha de mostrar. Per a saber si estem a l'App, comprovem si l'App està en primer plànol o no mitjançant el mètode 'primerPlanol'. Tot seguit es mostra el codi comentat:

```
//amb aquest mètode sabem si l'App s'està executant o no (si està en
primer plànol o no)
public boolean primerPlanol(String PackageName){
    // Obtenim l'Activity Manager
    ActivityManager manager = (ActivityManager)
getSystemService(ACTIVITY_SERVICE);

    //Aconseguim una llista de les tasques que corren, només estem
interessats en l'últim (el cim)
    // per tant, donem 1 com a paràmetre  i llavors només obtenim el
més alt
    List< ActivityManager.RunningTaskInfo > task =
manager.getRunningTasks(1);

    // Obtenim l'informació que necessitem per a la comparació.
    ComponentName componentInfo = task.get(0).topActivity;

    //Comprovem si coincideix amb el nostre PackageName
    if(componentInfo.getPackageName().equals(PackageName)) return
true;

    // Si no, la nostra aplicació no està en primer plànol
    return false;
}
```

A continuació es mostra un imatge de l'Avís:

4.11 – ALTRES ASPECTES

També he fet que es pugui sortir de les diferents pantalles pitjant un botó ('Home', 'Sortir'...) i també amb el botó 'back' de telèfon mòbil.

Per exemple, per sortir d'App, al 'MainActivity.java' he implementat els dos mètodes següents:

```
//Al pitjar al botó 'back' de mòbil
@Override
public boolean onKeyDown(int keyCode, KeyEvent event)
{
    if ((keyCode == KeyEvent.KEYCODE_BACK))
    {
        //mostrem el quadre de diàleg
        dialogo();
    }
    return super.onKeyDown(keyCode, event);
}
```

```
//al pitjar el botó 'sortir'
public void cerrar(View view) {
    //mostrem el quadre de diàleg
    dialogo();
}
```

Per anar de la pantalla principal al menú principal, a 'P_Principal.java' he fet servir aquests dos mateixos mètodes però, en vers de cridar al mètode 'dialogo()', poso el valor '0' a SharedPreferences (el codi es pot veure a l'apartat '4.3 – Veure Categoria'). A més, al mètode 'cerrar', faig 'finish()' per a finalitzar l'activitat. Per anar de la categoria a la pantalla principal també faig servir el mètode 'cerrar':

```
public void cerrar(View view) {
   finish();
}
```

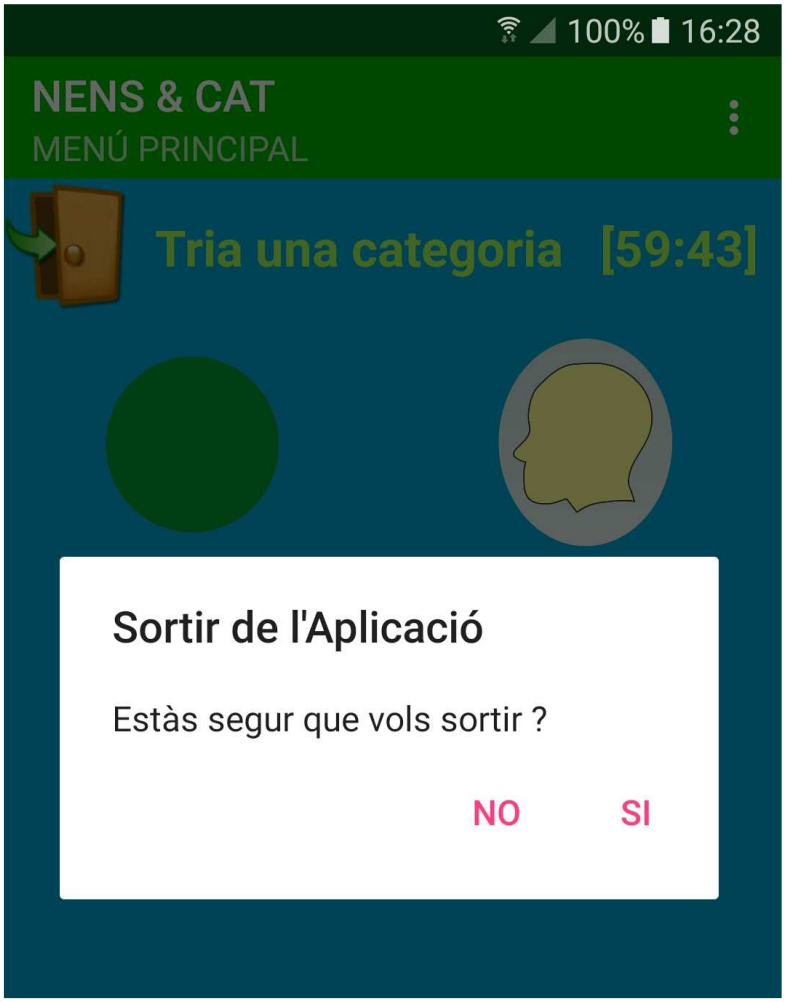
Com he dit més amunt, cridem al mètode 'dialogo()' Aquest mètode mostra un quadre de diàleg (que demana si estem segurs de sortir de l'App) al pitjar el botó 'sortir' o al pitjar a el botó 'back' de telèfon mòbil quan estem al menú principal, per a evitar que es surti de l'App al pitjar sense voler sense voler un d'aquests botons.

El codi del mètode 'dialogo()' és el següent:

```
//mètode que mostra un quadre de diàleg
public void dialogo (){
    AlertDialog.Builder dialogo1 = new AlertDialog.Builder(this);
    dialogo1.setTitle("Sortir de l'Aplicació");
    dialogo1.setMessage("Estàs segur que vols sortir ?");
    dialogo1.setCancelable(false);
    dialogo1.setPositiveButton("SI", new
DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialogo1, int id) {
            //aturem el cronòmetre
            cT.cancel();
            //finalitzem l'activitat
            finish();
            //el checkBox del menú overflow no ha d'estar seleccionat
al iniciar l'App, per tant, ho desseleccionem al sortir
            isChecked = false;
        }
    });
    dialogo1.setNegativeButton("NO", new
DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialogo1, int id) {
            //tanquem el quadre de diàleg
            dialogo1.cancel();
        }
    });
    dialogo1.show();
}
```

Com es pot veure al codi, si seleccionem 'SI' s'atura el cronòmetre, finalitza l'activitat i desseleccionem el checkBox del menú de l'ActionBar del Menú Principal. En canvi, si seleccionem 'NO', simplement es tanca el quadre de diàleg.

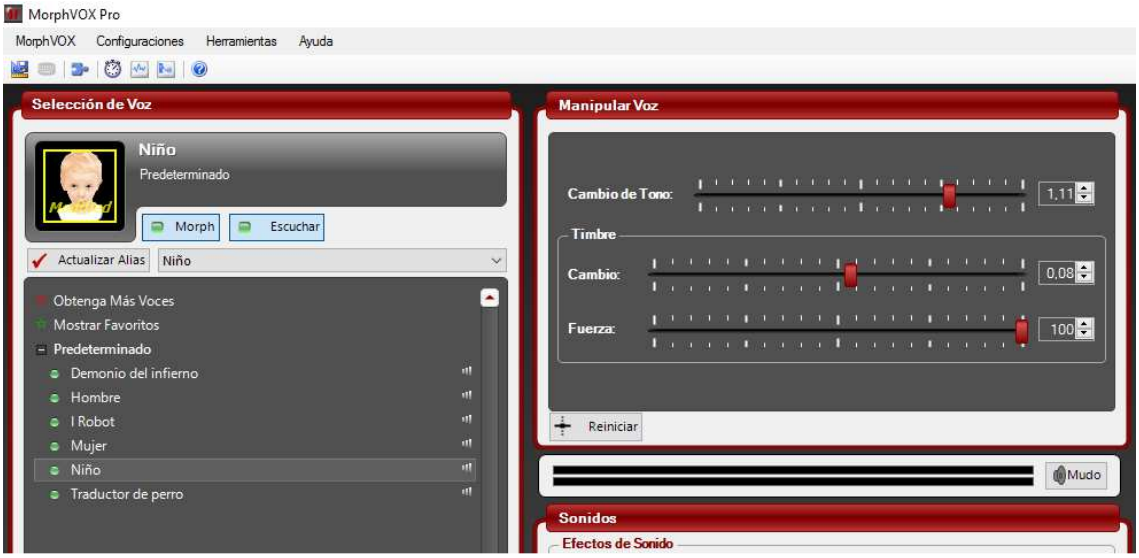
A continuació es mostra l'imatge del quadre de diàleg:



Per tant, he mirat de fer que les funcionalitats detallades fins ara es puguin fer de vàries maneres (pitjant botó de l'App o botó 'back' del mòbil, pitjant botó de l'App o amb els dits, etc.) o es puguin mostrar les coses de vàries maneres (mostrar 'toast', mostrar imatge, mostrar, escoltar-se una veu, etc.). Això ho he fet perquè, d'aquesta manera, es millora l'usabilitat i experiència dels usuaris (nens).

Per altra banda, dir que he gravat un arxiu d'àudio per a cadascuna de les paraules. Text-to-Speech de Google ('Síntesis de voz de Google') no hi és en català i a altres opcions, la veu és de poca qualitat, no és clara. A més, considero que és millor gravar jo mateix la veu ja que, utilitzant sintetitzadors de veu, puc fer que la meva veu soni com la d'un nen i quedarà millor a una App infantil.

He utilitzat el programa MorphVOX Pro per a gravar la meva veu i transformar-la:



He afegit una pantalla de benvinguda (Splash Screen), amb el logotip de l'App. La pantalla de benvinguda surt durant un temps determinat (simulant que es carrega l'aplicació, 'Loading...') abans d'iniciar-se l'aplicació. Al desaparèixer aquesta pantalla, es mostra el menú principal.

A continuació es mostra el codi ('Emergent.java') que he implementat per a fer la pantalla de benvinguda:

```
// Establim la durada de la pantalla de benvinguda
private static final long SPLASH_SCREEN_DELAY = 3000;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    // Establim orientació vertical
    setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
    // Ocultem la barra de títol
    requestWindowFeature(Window.FEATURE_NO_TITLE);

    setContentView(R.layout.emergent);

    TimerTask task = new TimerTask() {
        @Override
        public void run() {

            // Iniciem la següent activitat
            Intent mainIntent = new Intent().setClass(
                    Emergent.this, MainActivity.class);
            startActivity(mainIntent);

            // Taquem l'activitat perquè l'usuari no pugui tornar a
aquesta activitat prement el botó Enrere
            finish();
        }
    };

    // Simulem un procés de càrrega ('Loading...') abans de l'inici de
l'aplicació
```

```
    Timer timer = new Timer();
    timer.schedule(task, SPLASH_SCREEN_DELAY);
}
```

La classe 'Emergent.java' la 'incloem' al 'AndroidManifest.xml'de la següent manera:

```
<activity
    android:name="tfm.nenscat.Emergent"
    android:screenOrientation="portrait">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```

És a dir, és la classe que es llança al principi.

Com es pot veure, ocultem la barra de títol i establim l'orientació vertical de la pantalla. A més, fem servir un Timer. Quan acaba el Timer, iniciem l'activitat 'MainActivity'. A continuació es mostra l'imatge de la pantalla de benvinguda:

En aquesta App, he emprat 'ImageButtons' amb imatges molt reconeixibles en lloc de 'Buttons' amb un nom concret. He fet això perquè crec que és millor pels nens usar icones o imatges que no botons amb text, ja que és més intuïtiu per ells.

Com que faig servir 'ImageButtons', he fet el següent (a cada element del GridView del menú principal i a cada 'ImageButton' de l'App) per tal de simular l'efecte click al menú principal, pantalla principal i botó 'Enredera' de 'categoria.xml'. Per exemple, per al botó 'Enredera' de 'categoria.xml':

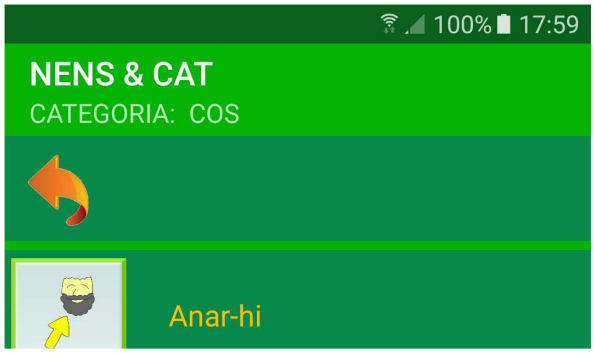
('gradient_cat.xml')
```
<?xml version="1.0" encoding="utf-8"?>
<layer-list xmlns:android="http://schemas.android.com/apk/res/android"
>
    <item>
        <bitmap android:src="@drawable/enrera"/>
    </item>
    <item>
        <shape
xmlns:android="http://schemas.android.com/apk/res/android">
            <gradient android:angle="90" android:startColor="#01DF3A"
android:centerColor="#04B431" android:endColor="#01DF3A"/>
        </shape>
    </item>
</layer-list>
```

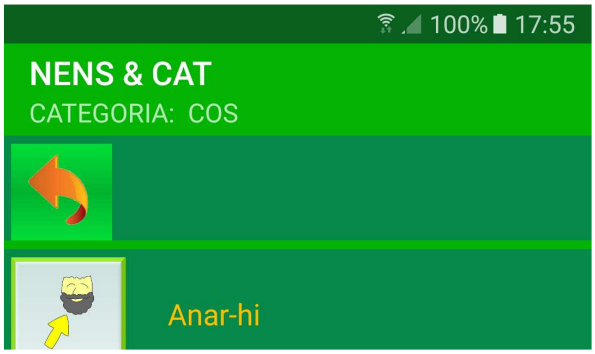
('effectbutton_cat.xml')
```
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:state_focused="true" android:state_pressed="false"
android:drawable="@drawable/enrera" />
    <item android:state_focused="true" android:state_pressed="true"
android:drawable="@drawable/gradient_cat" />
    <item android:state_focused="false" android:state_pressed="true"
android:drawable="@drawable/gradient_cat" />
    <item android:drawable="@drawable/enrera" />
</selector>
```

Finalment, a les propietats del 'ImageButton':

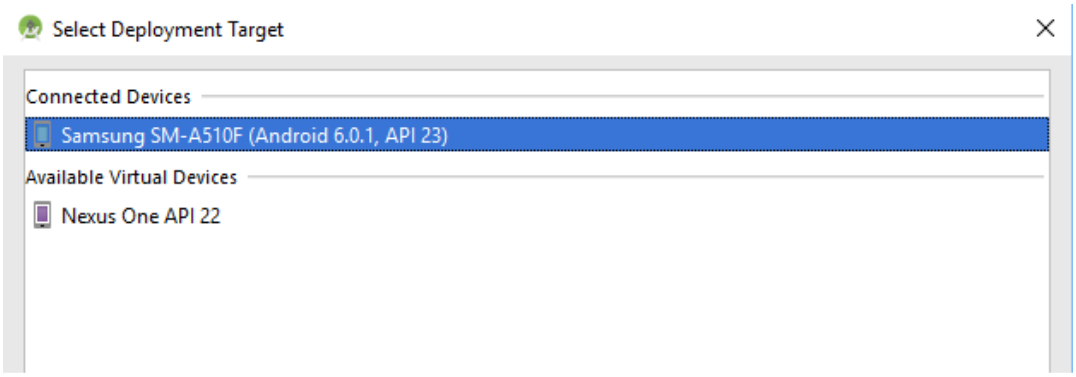
```
android:background="@drawable/effectbutton_cat"
```

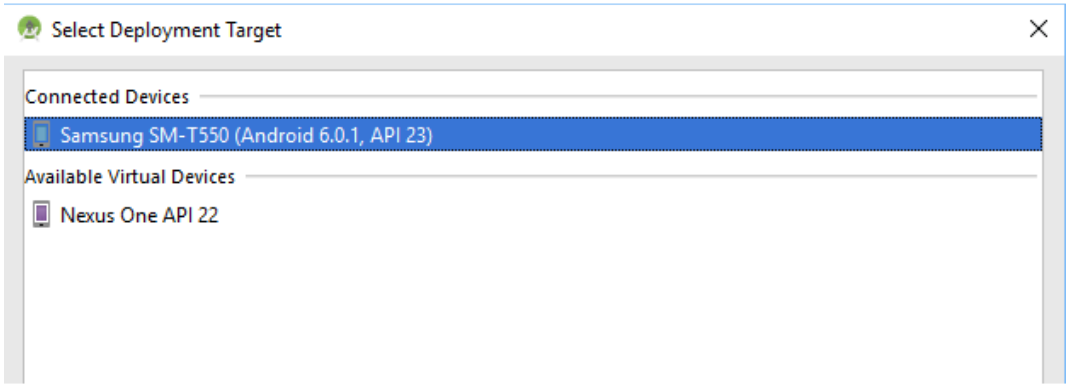
Es pot veure que al pitjar al botó el fons canvia:





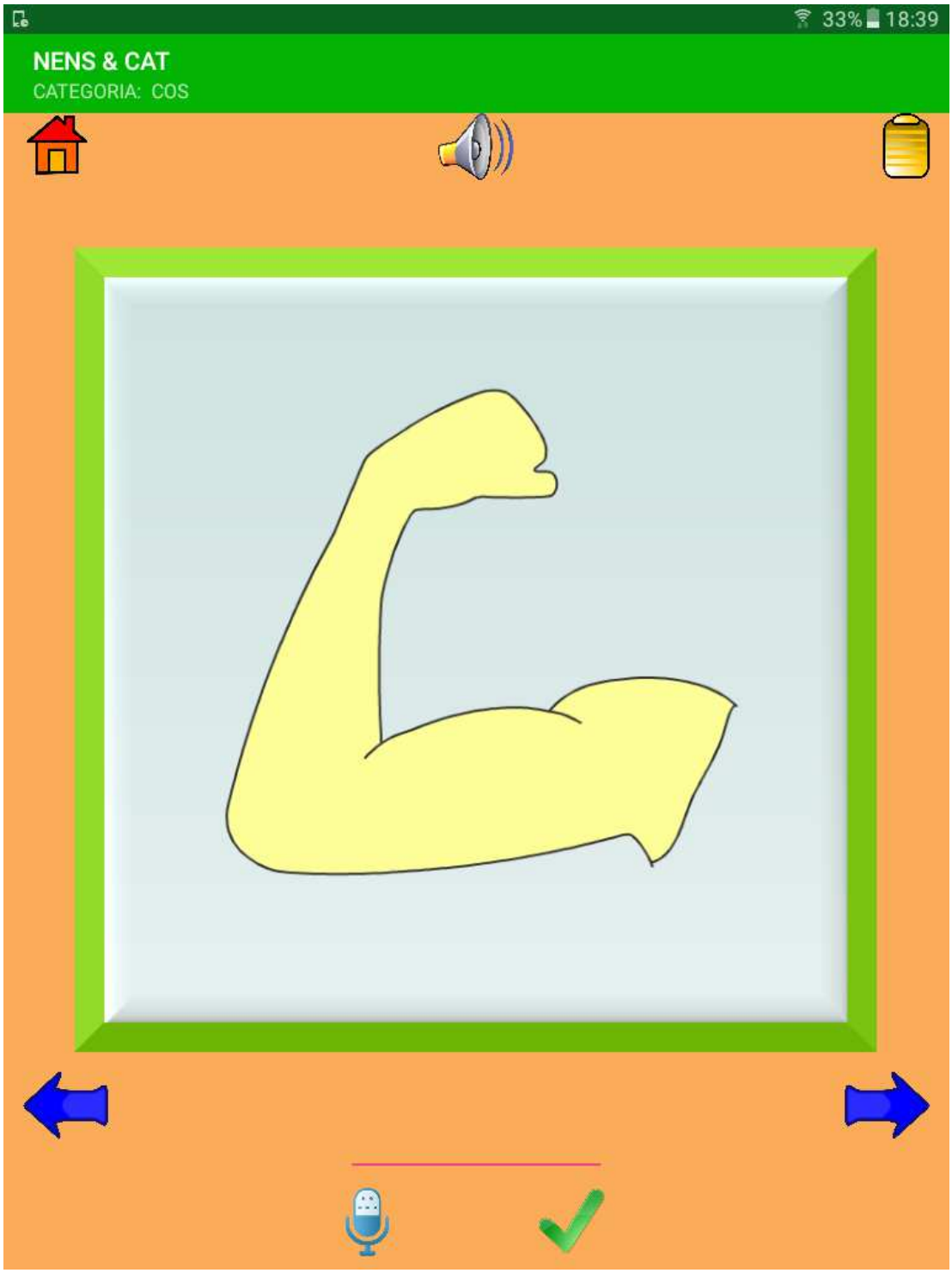
L'App l'he anada provant al meu telèfon mòbil i a la tableta:





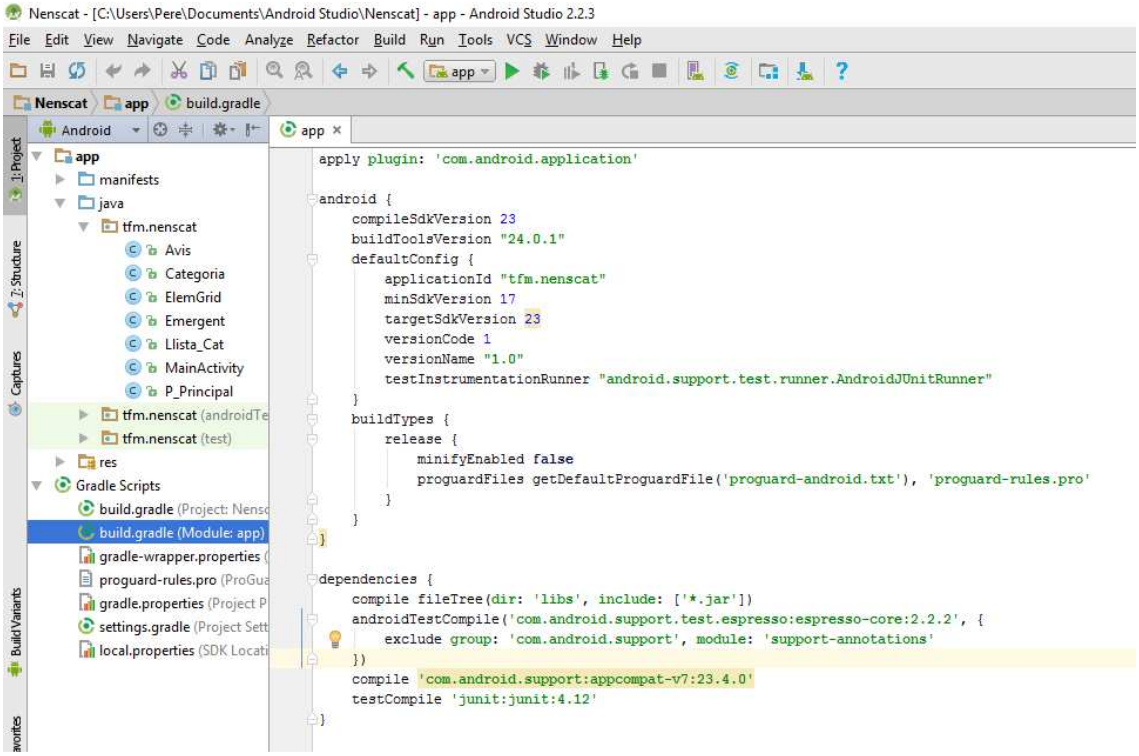
He provat de fer al mòbil i a la tableta cadascuna de les funcionalitats que he citat anteriorment. A les imatges anteriors d'aquesta memòria, es pot observar com funcionen algunes de les funcionalitats.

A continuació es mostra una imatge de l'App a la tauleta (fins al moment, havia mostrat imatges de l'aplicació al telèfon mòbil):



Com es pot observar, a la tableta també funciona i es veu bé.

Les propietats de compilació, etc. de l'aplicació són:



5 – CONCLUSIONS

Aquest TFM m'ha servit molt per a 'aficionar-me' al desenvolupament Android. Abans de començar aquest treball, no disposava quasi de coneixements en Android ni havia emprat encara l'Android Studio.

Gràcies a haver fet aquest treball, he après a emprar l'Android Studio i he après també moltes de les característiques de Android i a implementar-ne APIS, etc. També m'ha servit per a refrescar la programació en Java.

Per altra banda, estic content del resultat final de l'App. Tenint en compte que és la primera aplicació que he implementat i, tenint en compte que és una aplicació senzilla e infantil, l'aplicació té una aparença bona i realitza correctament les funcionalitats plantejades d'un principi.

En resum, ha estat una experiència molt positiva amb un resultat que crec ha estat bo.

Futures Millores:

- La possibilitat de definir perfils per a que més d'un nen pugui utilitzar l'aplicació en el mateix telèfon (o tauleta). Que el nen es pugui fer una foto amb la seva càmera per identificar el seu perfil.

- L'assoliment de "medalles" quan el nen aconsegueix arribar a certs objectius (resoldre correctament un cert nombre de paraules, connectar-se a l'App un cert nombre de vegades, etc.)

- També m'agradaria desenvolupar l'App per a què funcioni en dispositius iOS i també fer que funcioni a través de la web i, d'aquesta manera, poder consultar l'App a través del ordinador.

6 – FONTS D'INFORMACIÓ

A continuació es mostren les fonts d'informació que he fet servir per a dur a terme aquest projecte:

- StackOverflow '**http://stackoverflow.com'**

- Desarrolladores de Android **'https://developer.android.com/index.html'**

- Wikipedia **'https://es.wikipedia.org/'**

- Google **'https://www.google.es/'** (per a cercar imatges, etc.)

- Wiki UOC Desenvolupament d'aplicacions per a dispositius Android
'http://cv.uoc.edu/webapps/xwiki/wiki/matm0552ca/'