UNIVERSITAT POLITÈCNICA DE VALÈNCIA

etsinf

Escola Tècnica Superior d'Enginyeria **Informàtica**

VERSADOR PER A CANT VALENCIÀ D'ESTIL

TREBALL FI DE GRAU

Aldo Ferrando Esteve

ETSINF

GRAU EN INGENIERÍA INFORMÀTICA

FEBRER 2018

Tutors del projecte:

Ferran Pla

Lluís F. Hurtado

Resum

La finalitat d'aquest projecte ha sigut la d'implementar una aplicació utilitzant algoritmes basats en les cadenes de Markov amb diccionaris invertits que permeta la generació de cançons en valencià.

El resultat final és una *app* que pot ser instal·lada i utilitzada en qualsevol mòbil o tauleta amb OS Android, per la qual cosa, pot ser utilitzada fàcilment per tota classe de públic en general o pels compositors com a ferramenta de generació de cançons.

Paraules clau: canço, poema, Markov, Android.

Abstract

The purpose of this project has been to implement an application using algorithms based on Markov chains with inverted dictionaries that allow the generation of songs in valencià.

The final solution is an app that can be installed and used on any mobile or tablet with Android OS, which can be easily used by all kinds of audiences in general or by composers as a tool for generating songs.

Keywords : song, poem, Markov, Android.

Tabla de contingut

1. Introducció 7
2. Objectius 9
3. Aproximacions 11
3.1 Basada en cadenes de Markov 11
3.2 Basada en gramàtiques 13
4. Tecnologies 23
5. Implementació del programa 27
5.1 Obtenció del corpus lingüístic 27
5.2 Implementació amb l'etiquetatge POS 29
5.3 Implementació amb les cadenes de Markov 31
5.4 Implementació del disseny de l'*app* 33
6. Exemples d'execució 39
7. Conclusions i treballs futurs 47
8. Agraïments 49
9. Glossari de termes utilitzats 51
10. Referencies i bibliografia 55

1. Introducció

La motivació principal per a realitzar aquest projecte ha sigut la del repte de construir una aplicació en la meua llengua materna, el valencià, i també la de dissenyar aquesta aplicació per a l'art de la composició musical.

Aquest programa pretén ser una ferramenta de consulta per als experts en l'art musical o novells, ajudant-los a través de cançons compostes amb aquesta *app*, per poder ser d'ajuda en la tasca de composició.

Altra motivació ha sigut la d'implementar una *app* en Android. Aquest entorn no s'ensenya a la universitat però vaig obtenir tota la informació necessària per a la implementació del programa a través de la documentació disponible en la web Android. Aquesta situació ha sigut un repte i m'ha motivat profundament.

Les cançons i els poemes són coneguts per ser una forma molt avançada de la comunicació lingüística. Són formulacions particulars ben elaborades de certs missatges amb restriccions tant en la seua forma com en continguts.

El desig de volar per a l'home s'utilitza sovint com un exemple de com la pràctica d'enginyeria pot donar lloc a l'emulació dels comportaments observats en la natura. També es utilitzat per il·lustrar la idea que la millor imitació (tal com un avió de reacció) d'un fenomen natural (com el vol de les aus) no necessita sempre el reflex fidel de totes les característiques del fenomen inspirador.

Des del punt de vista de l'enginyeria, la capacitat d'optimitzar la combinació de la forma i el contingut del llenguatge és una característica molt desitjable a ser modelada. L'estudi del fenomen de la composició poètica potencia les fronteres del que els ordinadors poden fer, en termes de tractament del llenguatge humà. Aquest esforç s'enfronta a dos enfocaments possibles (observació i solució), intentar imitar el comportament en el llenguatge dels éssers humans i la tasca de buscar la millor solució possible que les tecnologies disponibles ens podrien brindar.

2. Objectius

Aquest treball es presenta com una aproximació a l'art poètic de la cançó valenciana, de manera general l'objectiu final que s'ha marcat durant la realització d'aquest projecte ha sigut el desenvolupament d'una eina d'entreteniment i inspiració per als artistes compositors i per als aficionats a la poesia.

El present projecte tracta de buscar un mètode de creació de cançons en valencià. Per a aconseguir-ho ha sigut necessari l'obtenció del corpus d'entrenament de poemes en valencià, per al posterior tractament d'aquestes dades amb l'etiquetatge morfosintàctic, finalitzant amb la composició de les oracions necessàries per a crear les cançons desitjades.

El propòsit d'aquest sistema de composició de cançons és proporcionar una imitació de la creació artística i poètica dels compositors mitjançant algorismes matemàtics i la tecnològica de què disposem hui en dia a l'abast de la mà (mòbils i tauletes).

Específicament, els objectius han sigut, primerament obtindré una recopilació de poemes d'on després s'ha aconseguit el corpus lingüístic, a través de la web http://ww.poesia.cat/ aquests poemes s'obtingueren de la web amb una sèrie d'algoritmes en JavaScript, filtrant-se el seu contingut per a obtenir només els poemes en text pla. En segon lloc es van implementar algoritmes en Java per a emmagatzemar en diccionaris enllaçats totes aquestes paraules. En aquests diccionaris les paraules apunten a la següent paraula possible basant-se en el corpus lingüístic. Després s'implementaren mètodes per a compondre una frase correcta gramatical i sintàcticament, amb el fi que les frases obtingudes tinguen sentit per a la creació correcta de cançons valencianes. Finalment, amb Android Studio es dissenyà l'app per a un entorn d'ús que està disponible per a qualsevol persona, tauletes i mòbils que permet a músics i aficionats gaudir d'aquesta aplicació.

3. Aproximacions

En aquesta secció descric les aproximacions utilitzades en aquest projecte per a formar cançons en valencià en l'*app*.

3.1 Basada en cadenes de Markov

Els Processos de Markov formen una classe important dels processos estocàstics i tenen aplicacions en moltes àrees. Les aplicacions dels processos de Markov inclouen l'estudi de sistemes de control de velocitat en els vehicles de motor, les cues o línies de clients que arriben a un aeroport, taxes de canvi de divises, sistemes d'emmagatzematge, i els creixements de població de certes espècies animals. L'algoritme conegut com a *PageRank*, que va ser proposat originalment per al motor de cerca d'Internet Google, es basa en un procés de Markov.

La cadena de Markov és una extensió dels autòmats d'estats finits, aquest autòmat és definit per un conjunt d'estats i unes transicions entre estats que es basen en una entrada d'observacions. Un autòmat d'estat finit ponderat és un simple autòmat finit en el qual a cada arc s'associa una probabilitat que indica quin camí s'ha de seguir. La probabilitat de tots els arcs d'un estat ha de sumar 1.

Representació gràfica de les cadenes de Markov:

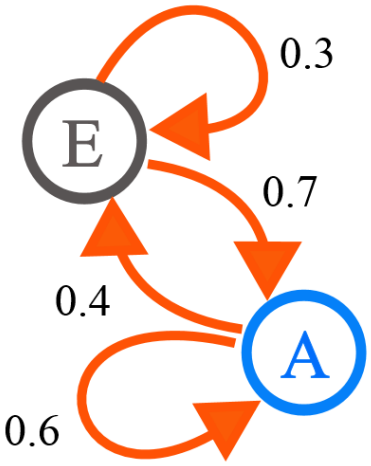


Figura Markov_1

Una cadena de Markov és un cas especial d'un autòmat ponderat, és una sèrie d'esdeveniments, en la qual la probabilitat que passe un esdeveniment depén de l'esdeveniment immediat anterior. Una cadena de Markov no representa problemes inherentment ambigus, per la qual cosa només útil per a assignar probabilitats a seqüències no ambigües.

Una cadena de Markov és un tipus de model gràfic probabilístic. Un simple model de Markov s'especifica com un conjunt d'estats Q, un conjunt de probabilitats de transició A, un estat inicial i un estat final.

Estats: $Q=q_1q_2..q_N$

Probabilitats de transició: $A=a_{01}a_{02}..a_{n1}..a_{nm}$. Cada a_{ij} representa la probabilitat de transició de l'estat i a l'estat j.

Cada a_{ij} representa la probabilitat $p(q_j|q_j)$ per aquest motiu la suma total dels arcs d'un estat donat suma 1.

Una cadena de Markov es útil quan es necessita calcular la probabilitat d'una seqüència d'esdeveniments observables. Les cadenes de Markov poden ser modelades amb màquines d'estat finit, i els recorreguts aleatoris proporcionen un exemple prolífic de la seua utilitat en matemàtiques.

En resum, per al desenvolupament d'aquesta solució s'ha emprat una cadena de Markov per obtenir la següent paraula desitjada en cada iteració.

3.2 Basada en gramàtiques

L'etiquetatge POS és el procés d'assignar a cada una de les paraules d'un text la seua categoria gramatical. L'etiquetatge d'un text és més difícil que simplement tindre una llista de paraules i les seues parts de l'oració, ja que algunes paraules poden tindre diferents parts de l'oració depenen del context. Per la qual cosa, un gran percentatge de paraules són ambigües.

En entorns acadèmics se solen distingir nou categories gramaticals: substantiu, verb, article, adjectiu, preposició, pronom, adverbi, conjunció, interjecció. No obstant això, existeixen moltes més categories i subcategories que augmenten la complexitat de l'etiquetatge morfosintàctic.

Tot açò reflecteix la complexitat de l'etiquetatge POS que s'ha utilitzat en aquest projecte.

FREELING

FreeLing és una biblioteca de C + + que proporciona serveis d'anàlisi lingüística (com ara l'anàlisi morfològica, el reconeixement de dates, l'etiquetatge de POS, etc.)

La versió actual proporciona identificació de l'idioma, tokenització, divisió de les frases, anàlisi morfològic, detecció de NE (entitats anomenades), classificació, reconeixement de les dates, números, magnituds físiques, divises, codificació fonètica, l'anàlisi sintàctica superficial, l'anotació del sentit de les paraules basada en wordnet, desambiguació semàntica i la resolució de la correferència. S'espera que les versions futures incorporaran noves característiques per a millorar el rendiment en les funcionalitats existents.

FreeLing està dissenyat per ser utilitzat com una biblioteca externa de qualsevol aplicació que necessite aquest tipus de serveis. Si l'aplicació que crida està escrita en C ++ natiu, es pot fer una trucada des del programa a la llibreria. Com a alternativa,

es proporcionen API per cridar les principals funcionalitats de *Freeling* des de programes realitzats en Java o Python.

A més a més, FreeLing pot utilitzar-se des de la línia de comandaments, amb moltes opcions de personalització, que permeten a l'usuari analitzar arxius de text sense necessitat de crear programes que criden a les llibreries.

TAGSET DE FREELING PER AL VALENCIÀ

Part of Speech: `adjective`

Position	Atribute	Values
0	category	**A**:*adjective*
1	type	**O**:*ordinal*; **Q**:*qualificative*; **P**:*possessive*
2	degree	**S**:*superlative*; **V**:*evaluative*
3	gen	**F**:*feminine*; **M**:*masculine*; **C**:*common*
4	num	**S**:*singular*; **P**:*plural*; **N**:*invariable*
5	possessorpers	**1**:*1*; **2**:*2*; **3**:*3*
6	possessornum	**S**:*singular*; **P**:*plural*; **N**:*invariable*

Part of Speech: `conjunction`

Position	Atribute	Values
0	category	**C**:*conjunction*
1	type	**C**:*coordinating*; **S**:*subordinating*

Part of Speech: `determiner`

Position	Atribute	Values
0	category	**D**:*determiner*
1	type	**A**:*article*; **D**:*demonstrative*; **I**:*indefinite*; **P**:*possessive*; **R**:*relative*; **T**:*interrogative*; **E**:*exclamative*
2	person	**1**:*1*; **2**:*2*; **3**:*3*
3	gen	**F**:*feminine*; **M**:*masculine*; **C**:*common*
4	num	**S**:*singular*; **P**:*plural*; **N**:*invariable*
5	possessornum	**S**:*singular*; **P**:*plural*; **N**:*invariable*

Part of Speech: `noun`

Position	Atribute	Values
0	category	**N**:*noun*
1	type	**C**:*common*; **P**:*proper*
2	gen	**F**:*feminine*; **M**:*masculine*; **C**:*common*
3	num	**S**:*singular*; **P**:*plural*; **N**:*invariable*
4	neclass	**S**:*person*; **G**:*location*; **O**:*organization*; **V**:*other*
5	nesubclass	*Not used*
6	degree	**V**:*evaluative*

Part of Speech: `pronoun`

Position	Atribute	Values
0	category	**P**:*pronoun*
1	type	**D**:*demonstrative*; **E**:*exclamative*; **I**:*indefinite*; **P**:*personal*; **R**:*relative*; **T**:*interrogative*
2	person	**1**:*1*; **2**:*2*; **3**:*3*
3	gen	**F**:*feminine*; **M**:*masculine*; **C**:*common*
4	num	**S**:*singular*; **P**:*plural*; **N**:*invariable*
5	case	**N**:*nominative*; **A**:*accusative*; **D**:*dative*; **O**:*oblique*
6	polite	**P**:*yes*

Part of Speech: `adverb`

Position	Atribute	Values
0	category	**R**:*adverb*
1	type	**N**:*negative*; **G**:*general*

Part of Speech: `adposition`

Position	Atribute	Values
0	category	**S**:*adposition*
1	type	**P**:*preposition*

Part of Speech: `verb`

Position	Atribute	Values
0	category	**V**:*verb*
1	type	**M**:*main*; **A**:*auxiliary*; **S**:*semiauxiliary*
2	mood	**I**:*indicative*; **S**:*subjunctive*; **M**:*imperative*; **P**:*participle*; **G**:*gerund*; **N**:*infinitive*
3	tense	**P**:*present*; **I**:*imperfect*; **F**:*future*; **S**:*past*; **C**:*conditional*
4	person	**1**:*1*; **2**:*2*; **3**:*3*
5	num	**S**:*singular*; **P**:*plural*
6	gen	**F**:*feminine*; **M**:*masculine*; **C**:*common*; **N**:*neuter*

Part of Speech: `number`

Position	Atribute	Values
0	category	**Z**:*number*
1	type	**d**:*partitive*; **m**:*currency*; **p**:*percentage*; **u**:*unit*

Part of Speech: `date`

Position	Atribute	Values
0	category	**W**:*date*

Part of Speech: `interjection`

Position	Atribute	Values
0	category	**I**:*interjection*

Part of Speech: `punctuation`

Tag	Attributes
Fd	**pos**:*punctuation*; **type**:*colon*
Fc	**pos**:*punctuation*; **type**:*comma*
Flt	**pos**:*punctuation*; **type**:*curlybracket*; **punctenclose**:*close*
Fla	**pos**:*punctuation*; **type**:*curlybracket*; **punctenclose**:*open*
Fs	**pos**:*punctuation*; **type**:*etc*
Fat	**pos**:*punctuation*; **type**:*exclamationmark*; **punctenclose**:*close*
Faa	**pos**:*punctuation*; **type**:*exclamationmark*; **punctenclose**:*open*
Fg	**pos**:*punctuation*; **type**:*hyphen*
Fz	**pos**:*punctuation*; **type**:*other*
Fpt	**pos**:*punctuation*; **type**:*parenthesis*; **punctenclose**:*close*
Fpa	**pos**:*punctuation*; **type**:*parenthesis*; **punctenclose**:*open*
Ft	**pos**:*punctuation*; **type**:*percentage*
Fp	**pos**:*punctuation*; **type**:*period*
Fit	**pos**:*punctuation*; **type**:*questionmark*; **punctenclose**:*close*
Fia	**pos**:*punctuation*; **type**:*questionmark*; **punctenclose**:*open*
Fe	**pos**:*punctuation*; **type**:*quotation*
Frc	**pos**:*punctuation*; **type**:*quotation*; **punctenclose**:*close*
Fra	**pos**:*punctuation*; **type**:*quotation*; **punctenclose**:*open*
Fx	**pos**:*punctuation*; **type**:*semicolon*

Una vegada tenim els poemes es procedeix a etiquetar-los amb la ferramenta FreeLing per al posterior tractament computacional, és a dir, s'etiqueten les paraules dels quatre mil poemes: adjectius A (amb els seus atributs, de tipus: ordinals O, qualificatius Q, possessius P, per grau: superlatiu S, avaluatiu V, gènere: femení F, masculí M, comú C, per nombre: singular S, plural P, invariable N), de conjuncions C, determinants D, noms N, pronoms P, adverbis R, aposicions S (preposicions P), verbs V, números Z, dates W, interjeccions I i tenint en compte les amb les puntuacions gramaticals (: Fd, , Fc, . Fp, ... Fs, ! Faa, - Fg, ? Fia, ; Fx, altre tipus Fz).

Ací tenim un exemple d'etiquetatge:

món NC -> Nom Comú.

des SP -> Preposició.

d NC -> De vegades l'etiquetatge comet errades. Aquest és un exemple, assigna Nom Comú, quan és en realitat una contracció de la preposició "de".

una DI -> Determinant Indefinit.

altra PI -> Pronom Indefinit.

. Fp -> Punt.

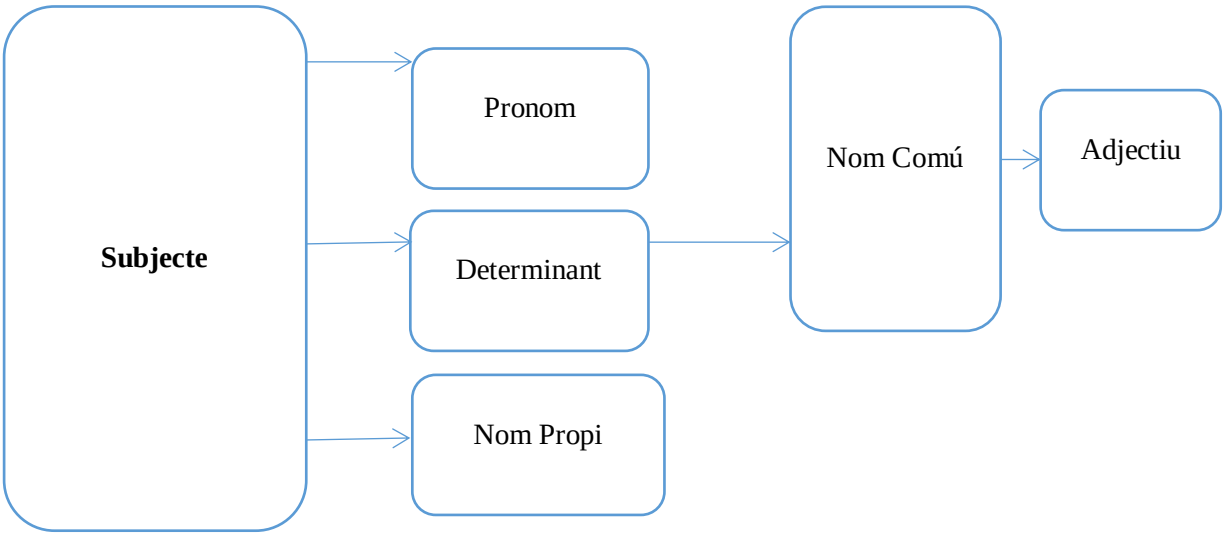
En una primera aproximació del programa, gràcies a l'etiquetatge POS, vaig implementar una sèrie de llistes tipus String en Java per a emmagatzemar les paraules.

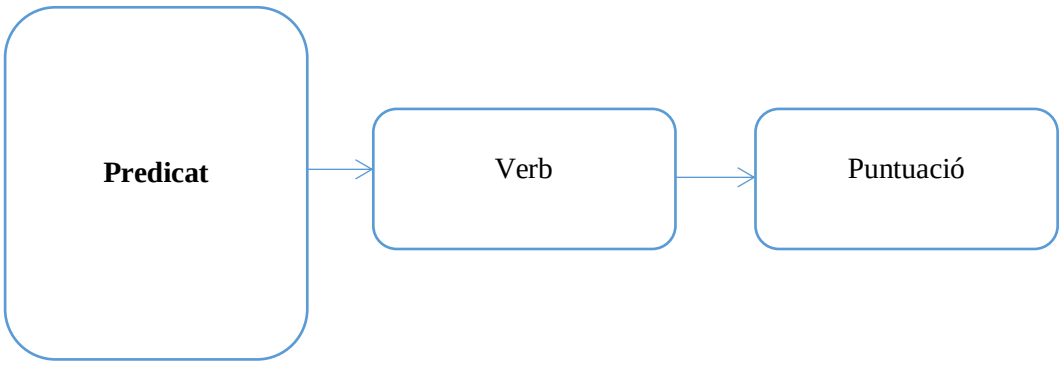
La construcció sintàctica que ordena els elements de la frase, segons la seua funció gramatical és:

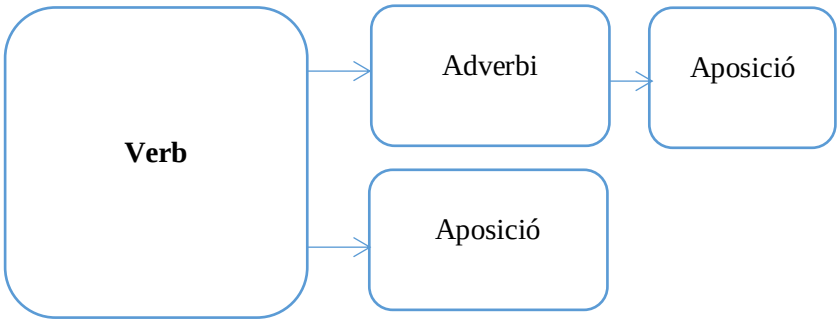
Subjecte - Verb - L'atribut o els complements (directe, indirecte i circumstancial).

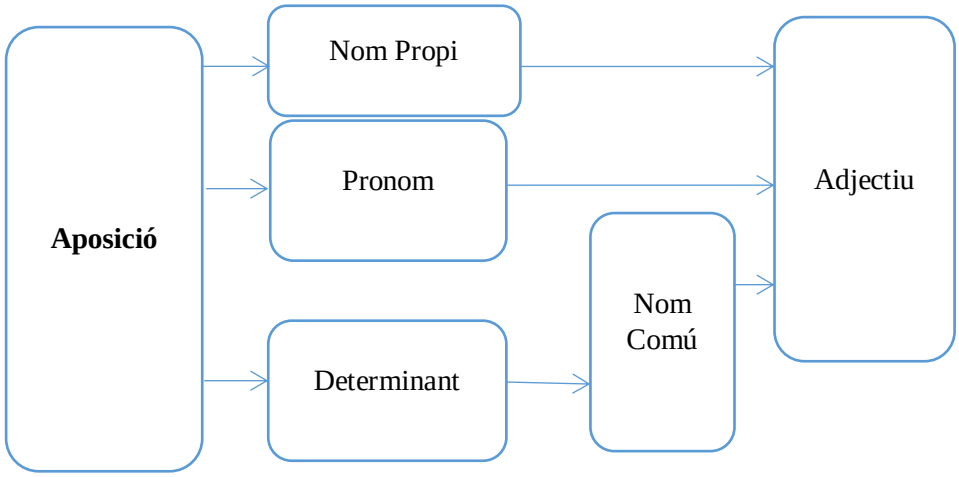
Per a estructurar correctament l'oració de la cançó he emprat un gràfic de creació sintàctica de frases que jo mateix he dissenyat.

Gràfic de creació sintàctica de oracions:

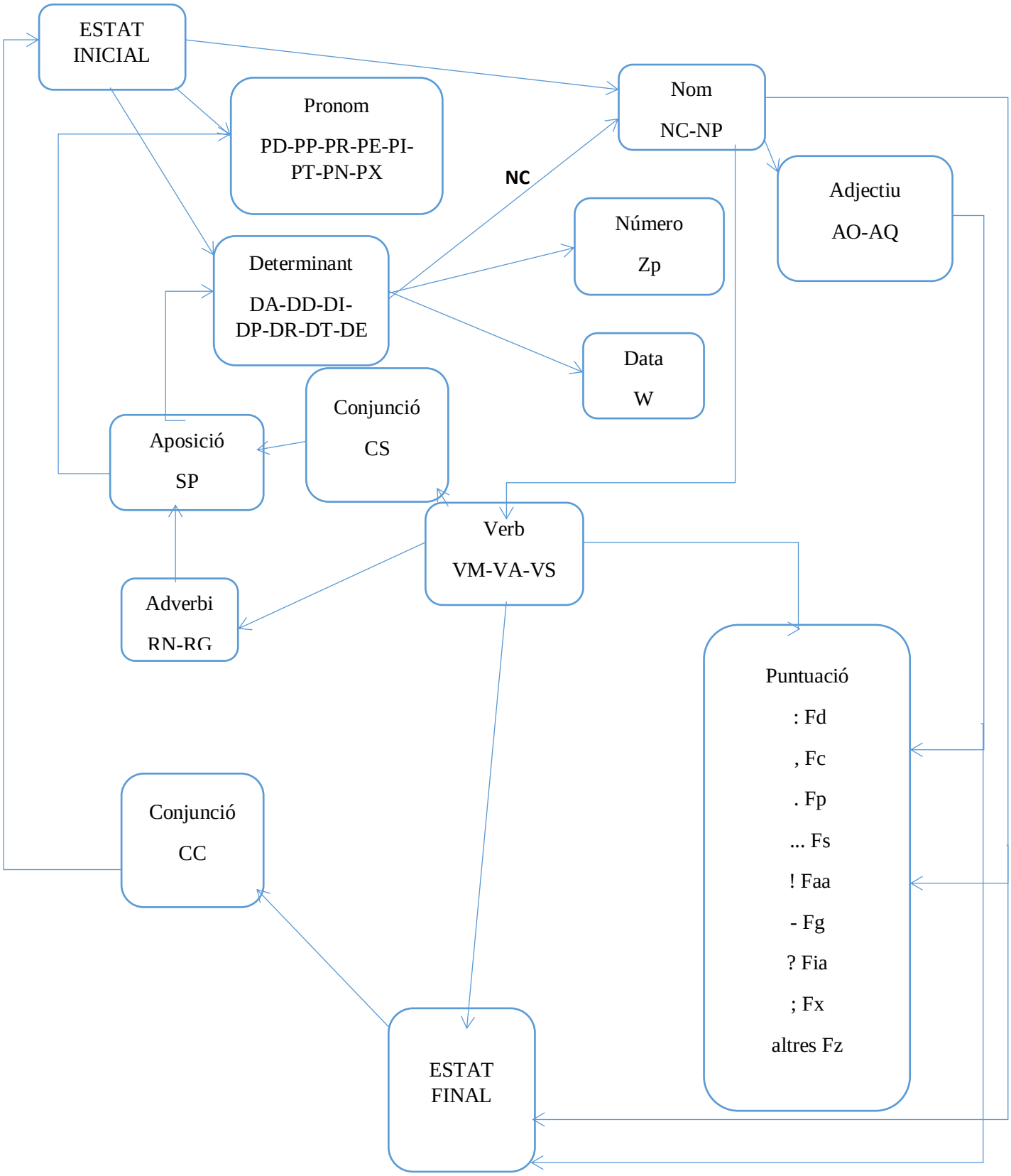








Gràfic de canvi d'estats amb etiquetes:



4. Tecnologies

En aquesta secció descric les tecnologies, llenguatge de programació i entorns de desenvolupament utilitzades en aquest projecte.

Java és un llenguatge de programació que produeix programari per a múltiples plataformes.

Quan un programador escriu una aplicació Java, el codi compilat (conegut com a *byte code*) s'executa en la majoria dels sistemes operatius (OS), inclòs Windows, Linux i Mac OS a través de la màquina virtual Java (*JVM*). Java deriva gran part de la seua sintaxi dels llenguatges de programació C i C ++. Aquest llenguatge de programació va ser desenvolupat a mitjans de la dècada de 1990 per James A. Gosling, un excientífic de Sun Microsystems.

He decidit implementar aquest programa versador de cant en valencià en aquest llenguatge per la seua capacitat de portabilitat i funcionament a tots els sistemes operatius i també per a poder executar-lo en Android.

JavaScript (JS) és un llenguatge Script de *compilació just a temps (o JIT)*. Tot i que és més conegut com a llenguatge de script per a pàgines web, molts entorn de programació també l'utilitzen, com ara NODE.JS i Apache CouchDB.

JS és llenguatge un multiparadigma basat en prototips, és un llenguatge dinàmic orientat a objectes. Per la seua fàcil implementació en pàgines web, he utilitzat aquest llenguatge per obtenir el corpus lingüístic.

Per a la implementació d'aquest programa s'han utilitzat els entorns de desenvolupament BlueJ i Android Studio.

BlueJ és un entorn de desenvolupament de Java dissenyat específicament per a l'ensenyament a un nivell introductori. Va ser dissenyat i implementat per l'equip BlueJ Deakin Universitat de Melbourne, Austràlia, i també a la Universitat de Kent a Canterbury, Regne Unit.

Les seues principals característiques són:

Senzillesa, té una interfície deliberadament més petita i més simple que els entorns professionals com NetBeans o Eclipse. Això permet als estudiants començar a programar de forma més senzilla i intuïtiva.

Dissenyat per l'ensenyament, està dissenyada deliberadament per a la pedagogia. A través de la internet es pot accedir a recursos didàctics sobre aquest entorn, moltes universitats, com la UPV, comencen l'ensenyament de programació amb l'entorn BlueJ.

Interactiu, permet interactuar amb els objectes creats, es pot inspeccionar el seu valor, utilitzar mètodes, de manera visual, de qualsevol objecte creat, passant els paràmetres per a la realització de proves fàcilment. També pot invocar directament expressions Java sense compilar. Per tant BlueJ és una potent consola gràfica per a Java.

Portable, s'executa en Windows, Mac, Linux i altres plataformes que suporten Java. També pot funcionar sense necessitat d'instal·lació des d'una memòria USB.

Madur, té més de quinze anys, però avui dia se segueix actualitzant.

Innovador, té diverses característiques que no es veien abans en altres entorns de desenvolupament. El seu banc d'objectes, l'entorn de codi, i la vivesa de colors són característiques originals BlueJ.

Més informació sobre BlueJ està disponible en http://www.bluej.org.

Android Studio és un entorn de desenvolupament integrat (IDE) de Google que ofereix als desenvolupadors les eines necessàries per a construir aplicacions per a la plataforma Android OS. Android Studio està disponible per a baixar en Windows, Mac i Linux. Es requereix una llicència per a una sola vegada, $ 25 de desenvolupador per publicar aplicacions a Google Play Store. El fonament per Android Studio està basat en IntelliJ IDEA.

L'estudi IDE Android es pot descarregar i utilitzar de forma gratuïta. Té un entorn de treball d'interfície d'usuari ric amb plantilles noves per donar als desenvolupadors una plataforma de llançament en el desenvolupament d'Android. Els programadors trobaran que l'entorn els dóna les eines per construir solucions de telèfon mòbil i de tauleta, així com solucions tecnològiques emergents per Android TV, Android Wear i models contextuals addicionals.

Android Studio ha sigut desenvolupat per a ser utilitzat pels equips de programadors de totes les grandàries. L'estudi IDE Android és compatible amb *GIT* i altres serveis de control de versions similars per a la gestió de treball en equips. Els desenvolupadors d'Android experimentats trobaran eines necessàries per als grans equips de treball, per oferir solucions ràpidament als seus clients. Les solucions Android es poden desenvolupar utilitzant Java o C ++ en Android Studio. El flux de treball per Android Studio es basa en el concepte d'integració contínua. Integració contínua permet als equips posar a prova el seu codi cada vegada que un desenvolupador realitza un canvi en el seu treball. Els problemes poden ser capturats i reportats a l'equip immediatament. El concepte de codi de comprovació contínua proporciona informació processable per als desenvolupadors amb l'objectiu d'alliberar versions d'una solució mòbil més ràpidament a l'aplicació Google Play

Store. Amb aquesta finalitat, hi ha un suport rigorós per a LINT *tools* i per al registre de programes.

Les eines de rendiment proporcionen accés per veure el rendiment de l'arxiu *Apk*. Les eines de rendiment i de perfils ens ensenyen una imatge codificada per colors per a mostrar amb quina freqüència es dibuixa el mateix píxel en una pantalla per reduir la representació de dalt. El processament de GPU mostra l'eficiència de la seua aplicació al manteniment de Google de referència de 16 ms per *frame*. Les eines de memòria de visualització mostren la quantitat de memòria RAM del sistema utilitzada en excés i quan es produeix l'alliberació de les variables no utilitzades (*Java Garbage Collector*), eines d'anàlisi de la bateria presenten la quantitat d'energia que utilitza un dispositiu a l'executar l'*app*.

Comentari de la instal·lació d'Android Studio en Linux:

La configuració d'instal·lació inicial va ser simple, però vaig haver de buscar un *plugin Opengl* per a la correcta visualització de l'*app* en l'ordinador, aquest pas era necessari per a rectificar el disseny i provar la funcionalitat del versador.

Després de trobar el *plugin* en la web d'Android Studio l'adherí a la configuració d'Android Studio, *configuració -> plugins* i finalment amb el reinici del sistema, l'entorn de programari Android Studio tenia una funcionalitat completa .

5. Implementació del programa

En aquesta secció detalle el procés d'implementació del programa, obtenció dels poemes i del disseny de l'*app*.

5.1 Obtenció del corpus lingüístic

Primerament, cal esmentar que per a l'obtenció del corpus de 351.709 paraules s'ha implementat en JavaScript una sèrie de funcions pera a extraure de la web http://www.poesia.cat/ d'un total de quatre mil poemes.

Ací detallaré el funcionament del codi implementat:

A l'inici del *HTML* de la pàgina detectí tots els elements amb l'etiqueta 'a' (és l'espai on estan les adreces dels poemes) i els emmagatzemí en un *array*, d'ací amb l'algoritme *httpGet*() obtinguí la pàgina web d'on s'allotja cada poema, després amb la funció existent *document.documentElement.outer* passí el HTML a text i amb l'ajuda de la funció *download*() emmagatzemí el text en el disc dur.

D'altra banda, per a fusionar els textos en emmagatzemats s'ha utilitzat la funció de CMD (MS-DOS) *copy* *.txt *target*.txt

Finalment per a "netejar" el text vaig utilitzar el programa Sublime Text, que ens permet, a través de regles, seleccionar el text que realment necessitem, evitant marques HTML i altres elements copiats de la web, amb la següent regla finalment he obtingut un text només amb poemes:

* [^<>]* ([^;\n\r]+)*

5.2 Implementació amb l'etiquetatge POS

Primerament, vaig implementar un algoritme que emmagatzema tots els poemes en un BufferReader. Després de recórrer tot el text es guarda cada paraula amb la seua etiqueta en una llista String.

Per a distingir les paraules a través de la seua etiqueta, s'utilitza una sèrie de condicionals, comparant amb les etiquetes POS.

Exemple:

```
if (etiqueta.equals("NP"))        {        nomPropi.add(lletra);        }
```

Fent Split en cada salt de línia " /n" s'emmagatzemen les paraules en llistes String.

Les llistes utilitzades foren aquestes:

```
List<String> nomPropi = new ArrayList<String>();
List<String> nomComu = new ArrayList<String>();
List<String> pronomP = new ArrayList<String>();
List<String> pronomR = new ArrayList<String>();
List<String> determinantA = new ArrayList<String>();
List<String> determinantD = new ArrayList<String>();
List<String> determinantI = new ArrayList<String>();
```

....

Una vegada emmagatzemades les paraules, tenint en compte la grandària de les llistes, vaig formar les frases en l'ordre sintàcticament correcte.

Extracte codi:

```
for (int i=0; i<4; i++)       {
        String randomSentence="";
        Random r = new Random();
        int index1 = r.nextInt(determinantA.size());
        int index2 = r.nextInt(nomComu.size());
        int index3 = r.nextInt(verb.size());
...
```

Les paraules foren obtingudes des de les llistes de manera aleatòria, per la qual cosa, encara que l'ordre sintàctic era adequat, la frase no era correcta.

Extracte codi:

```
randomSentence= determinantA.get(index1)+ " " + nomComu.get(index2) + " " +
verb.get(index3)+ " " + adverbiRG.get(index4)+ " " + adposicio.get(index5)+ " " +
determinantA.get(index6)+ " " + nomComu.get(index7);
```

5.3 Implementació amb les cadenes de Markov

Primerament amb la implementació del mètode addDictionary s'adhereixen les paraules dels poemes recollides pel mètode Split() sempre tenint en compte la puntuació ortogràfica.

Aquest diccionari emmagatzema per cada paraula les paraules consecutives que es relacionaven amb ella en els corpus lingüístic, és a dir, agafa una paraula i totes les que anaven seguides a aquesta, per exemple "la casa" "la cadira", la paraula "la" conté la relació amb les paraules "casa" i "cadira". En aquest exemple la paraula "la" serà la clau i la paraula "casa" seria un dels valors. El mètode dictionary.get(key) és el que obté el valor que correspon a la paraula anterior.

D'aquesta manera, i encara que totes les paraules consecutives són apropiades per a la paraula anterior, aquestes s'obtenen aleatòriament, la qual cosa permet obtindré composicions de cançons diferents cada vegada.

A més a més, he dissenyat el mètode pickRandom per a elegir aleatòriament les paraules a afegir a la frase de la cançó (les paraules clau) i amb el mètode generateSentence anem ajuntant les paraules que es van generant, junt amb les paraules valor que s'han obtingut de manera aleatòria.

Aquest últim algoritme finalitza la frase quan detecta que la següent paraula de la cadena apunta a *null*, és a dir, que la paraula clau no té cap valor per a tornar a la cadena de paraules. En el corpus lingüístic les paraules que solen ser les últimes en la frase seran les que determinen el final de les cançons.

També, l'algoritme puntualitzador(String s) permet, quan ja es forma una frase completa, incloure la coma, a final d'estrofa, o si és el final del vers, aquest mètode introdueix el punt final de la frase.

Tots aquests algoritmes ens permeten compondre frases per a la composició de caçons.

5.4 Implementació del disseny de l'*app*

Per al disseny de la interfície d'usuari he utilitzat l'entorn Android Studio, aquest entorn de programació m'ha permés afegir el disseny que havia estimat necessari per a aquest programa, he tingut en compte els principis d'ergonomia i senzillesa per optimitzar l'*app* i que l'ús de la mateixa puga ser intuïtivament accessible.

A més, a més, s'ha tingut en compte que el patró de colors siga l'adequat per a no fer mal a la vista amb l'ús continuat i resulten agradables a qualsevol persona que utilitze el programa.

Primer model de app:

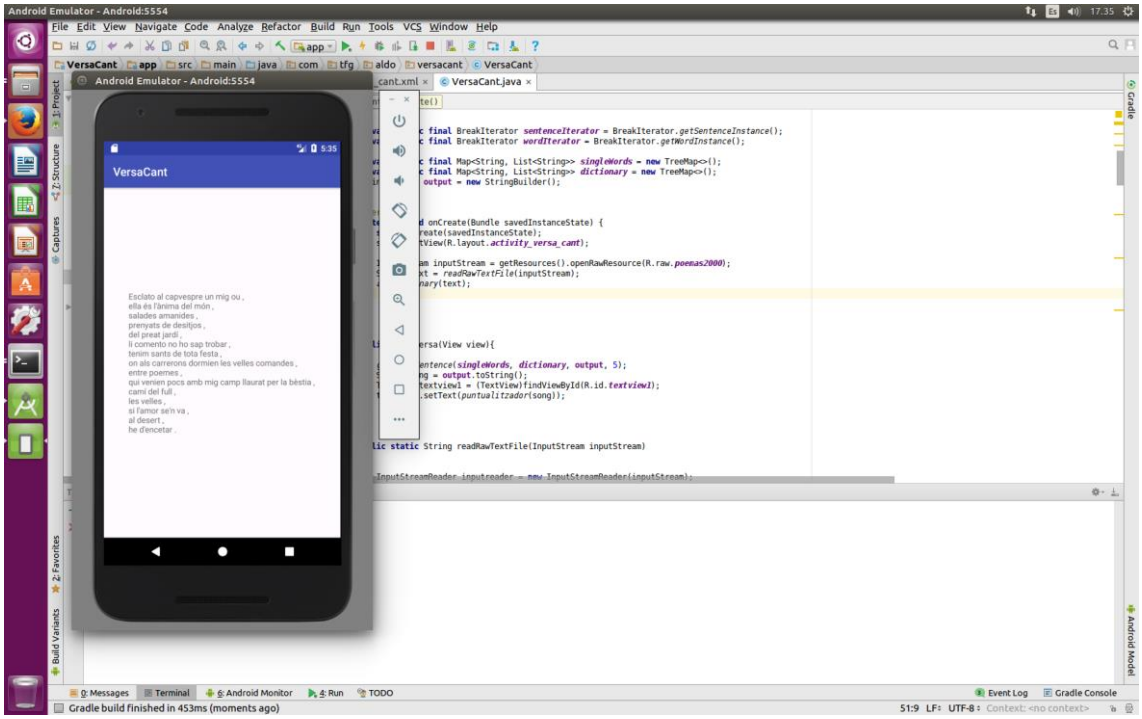


Figura Interficie_1

El primer model era una aproximació molt senzilla per a provar la funcionalitat de l'aplicació. Amb resultats s'observà el correcte funcionament i els algoritmes són capaços de generar cançons de forma molt senzilla.

En l'aplicació no hi havia opcions de confirmació i no tenia cap comprovació de la longitud de les estrofes ni els versos.

Disseny de l'icona:

Figura Icona_1

S'estudiaren diferents dissenys preliminars per a la confecció de la icona de l'*app*, però finalment es decidí optar per la que es mostra en la figura icona_1, ja que representa la tasca d'escriptura de la cançó, amb la representació del llapis he volgut reflectir l'esforç de la creativitat en la creació de les cançons, que de vegades ha de ser escrita i esborrada vàries vegades fins a culminar amb una gran obra.

Icona en el menú del mòbil:

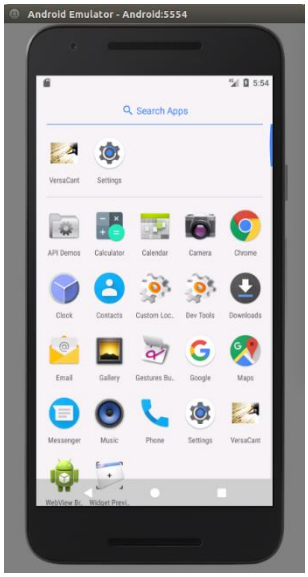


Figura Interficie_2

La icona s'ha integrat en l'entorn del mòbil tenint en compte els colors utilitzats en la icona, la grandària i el nom final de l'*app*, per tal d'aconseguir un ambient agradable de menú, no alterant l'estètica de l'entorn d'usuari del mòbil.

Dissenys intermedis:

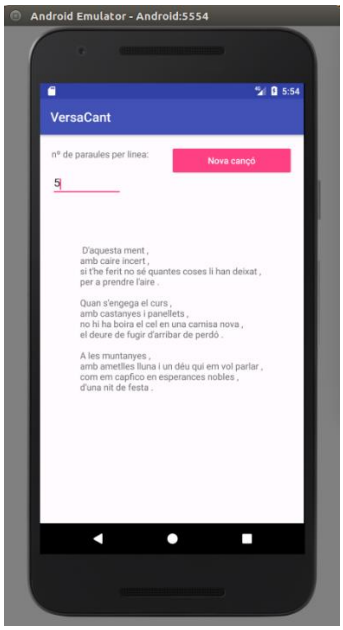


Figura Interficie_3

En primer lloc, es decidí en implementar un entorn amigable mitjançant un botó per a la creació de cada nova cançó.

D'altra banda, es va incloure una opció per a seleccionar el nombre de paraules per línia, però aquesta idea hagué de ser descartada perquè en especificar el nombre exacte de paraules, en alguns casos les frases generades pels algoritmes no tenien de sentit poètic.

Figura Interficie_4

Com s'aprecia en la figura Interficie_4, es va optar per la implementació de camps d'especificació de nombre d'estrofes i versos, d'aquesta manera es pot controlar la mida de les cançons i obtindré el resultat final amb la grandària desitjada.

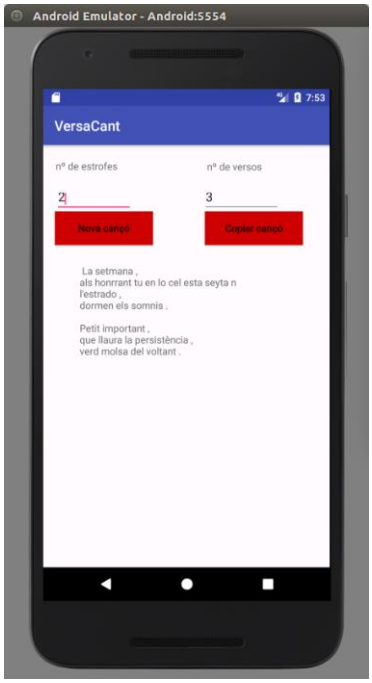


Figura Interficie_5

També, com es pot veure en la figura Interficie_5, s'afegí un botó de copiar amb la mateixa configuració i colors que el botó de nova cançó.

Fent clic en aquest botó s'emmagatzema el text de la cançó a la memòria del telèfon.

Posteriorment podem utilitzar aquest text en qualsevol altra aplicació del mòbil, sent possible millorar la cançó amb un editor de text.

També és possible enviar la cançó generada per correu electrònic o per missatgeria instantània pegant-li directament en aquests tipus d'*app*.

Procés d'implementació final del disseny:

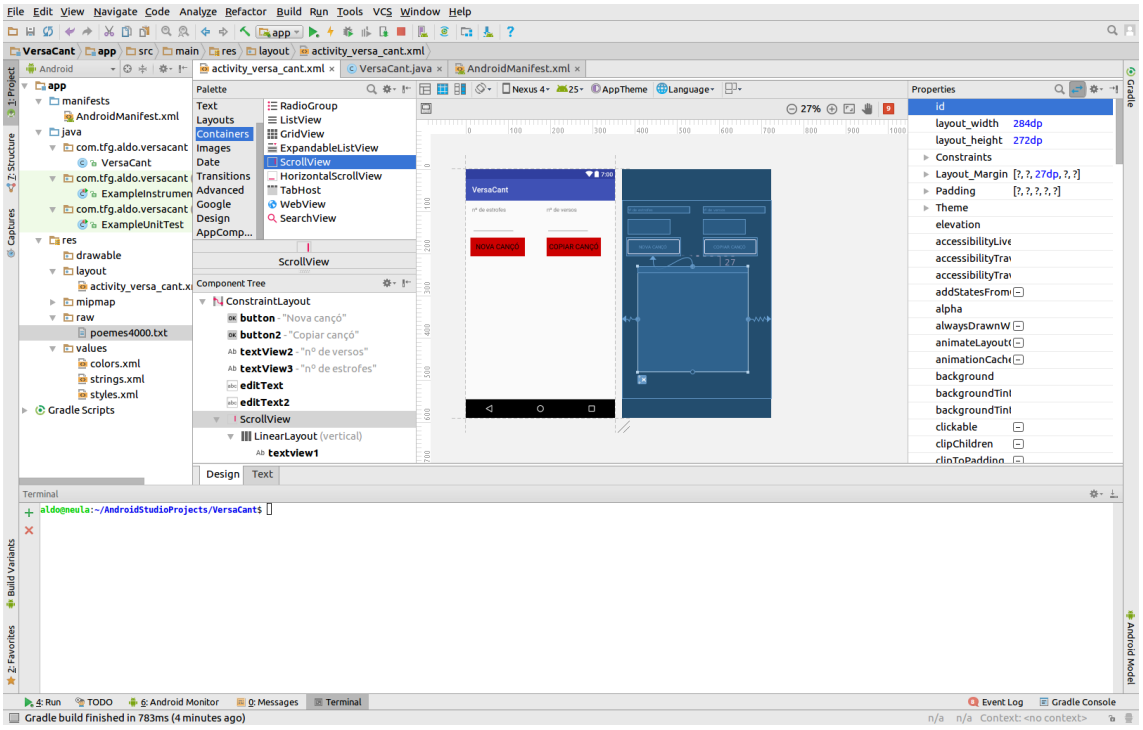


Figura Interficie_6

Finalment, s'ha tractat de tindre en compte, a l'hora de la implementació de l'aplicació, dissenyar una interfície agradable a la vista, amb gran usabilitat i senzillesa.

L'usuari final pot generar cançons fàcilment. També pot com copiar les cançons amb només un clic, la icona utilitzada per a copiar té el format estàndard per a còpia, reconeguda en tots els programes típics d'ordinador i mòbils.

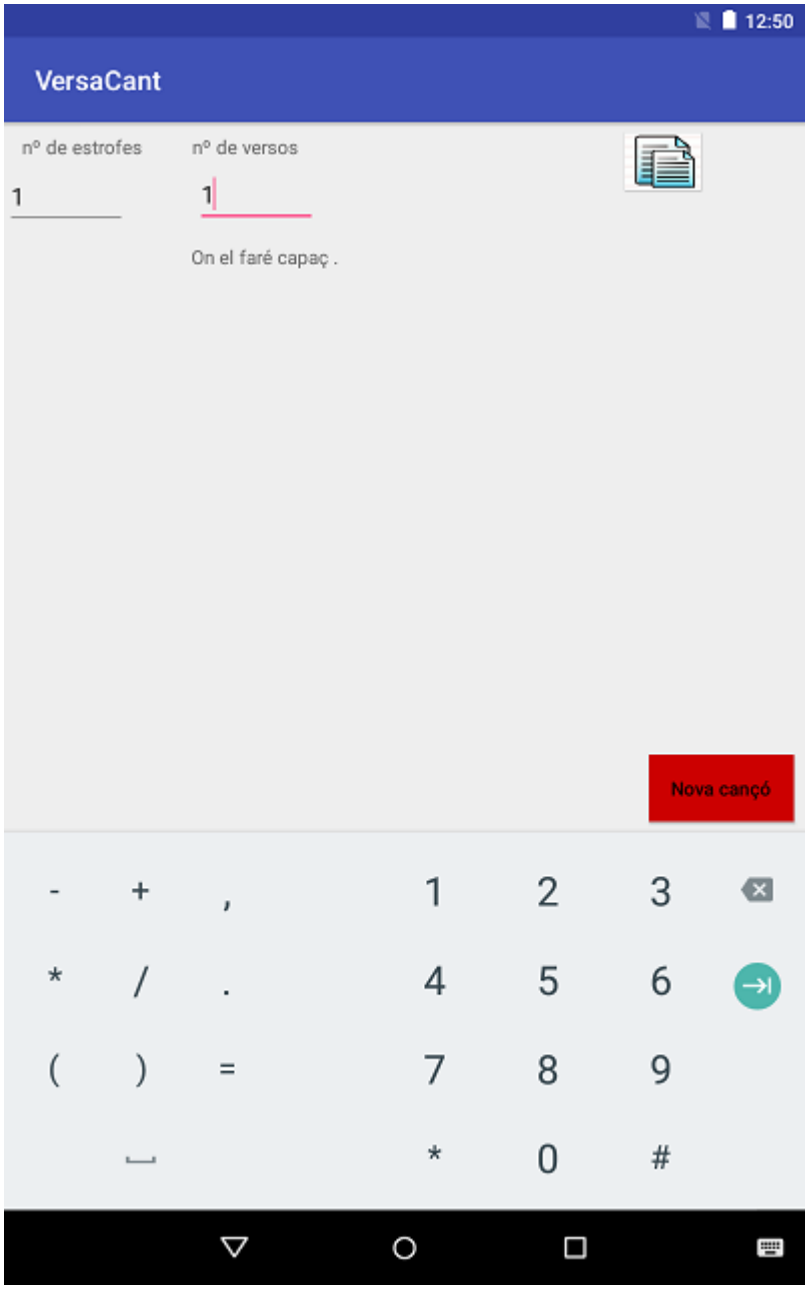
A més, a més el botó nova cançó, situat al final de l'*app*, s'adapta a la grandària de qualsevol mòbil i té fàcil accés.

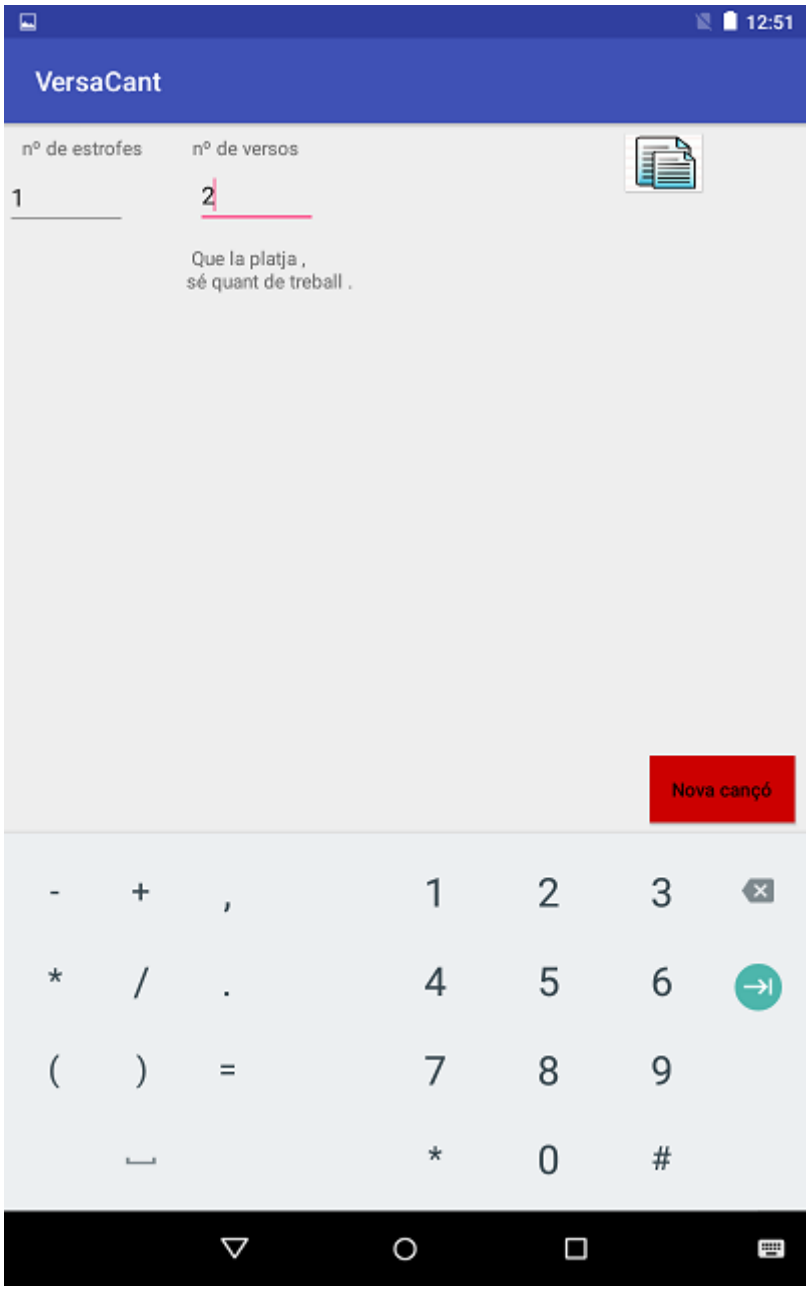
Disseny final de l'app:

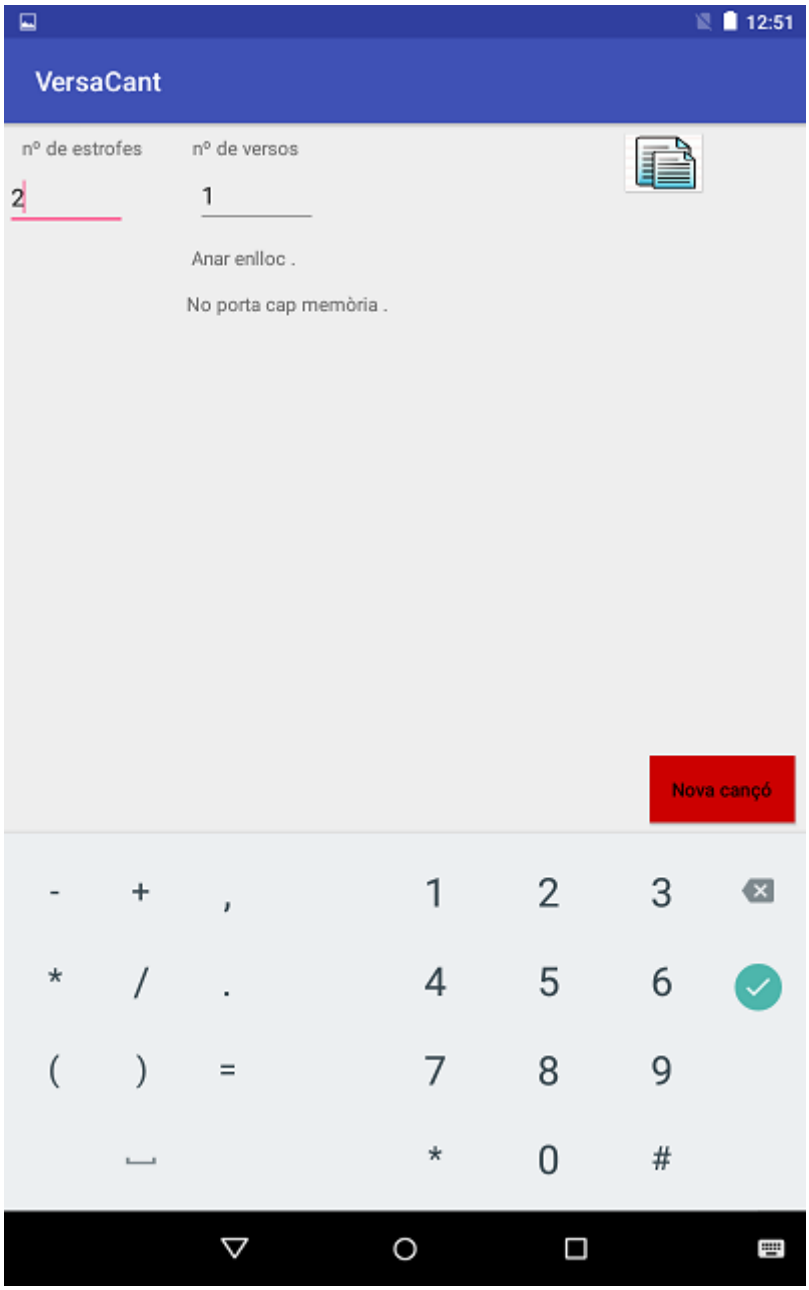
Figura Interficie_7

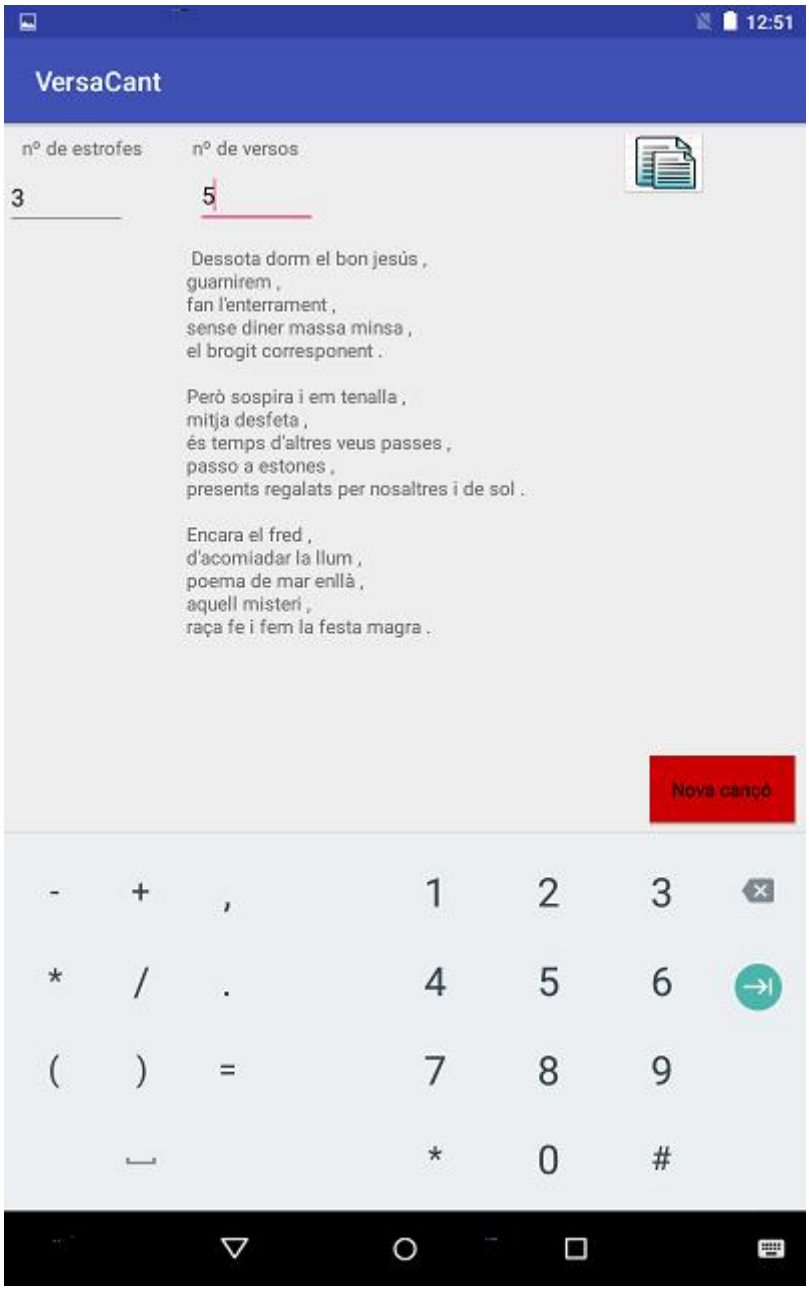
Per a concloure, cal esmentar que la combinació de colors barra blava, fons blanc i botó roig respecten tant la visual de l'usuari com la simbologia cromàtica. Aquesta combinació de colors és un clàssic per excel·lència.

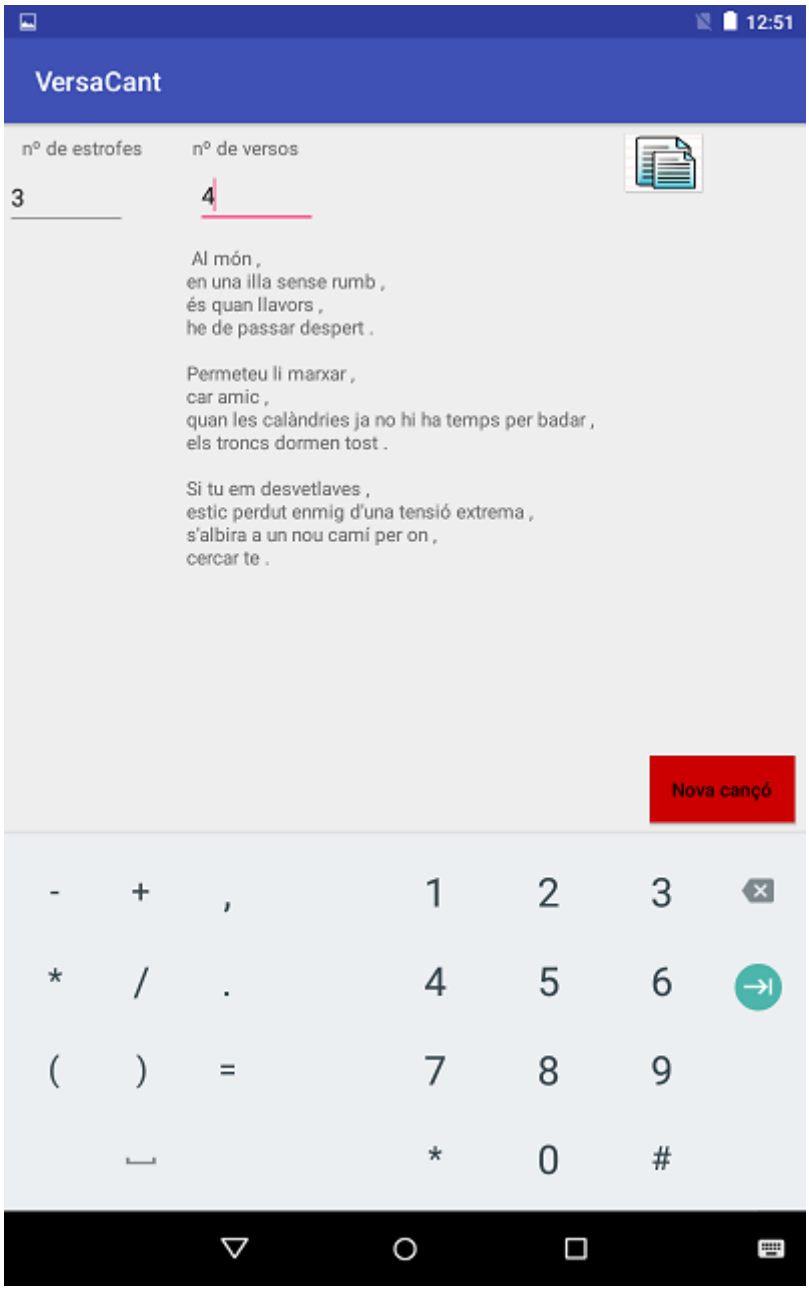
6. Exemples d'execució

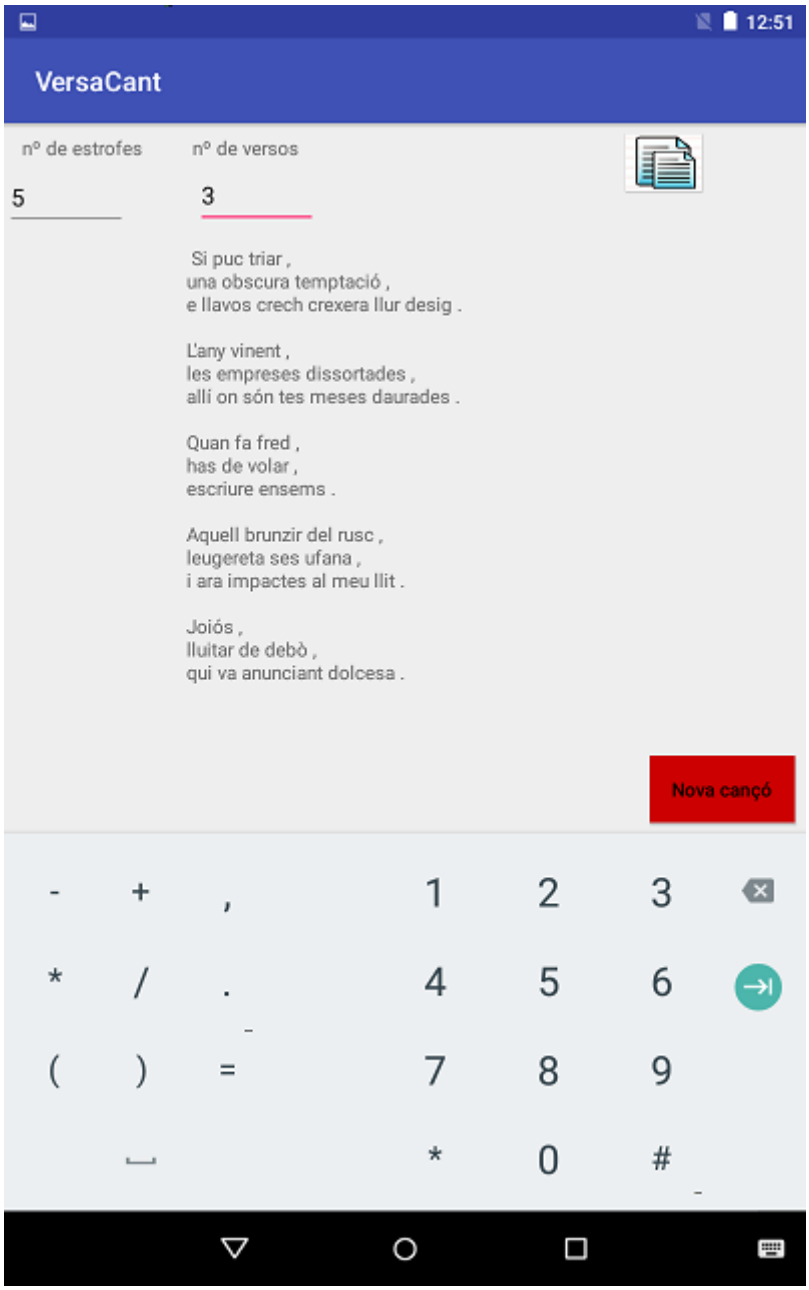


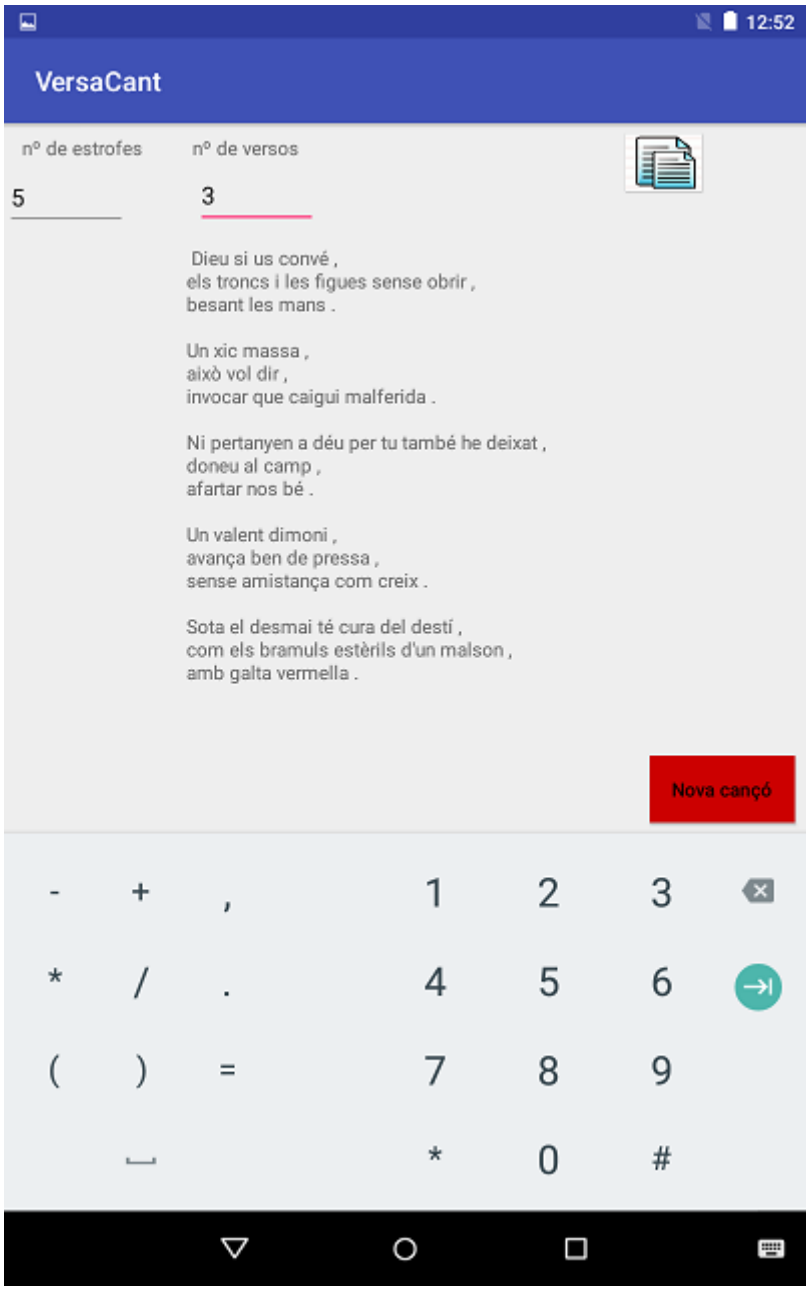

12:51
VersaCant
nº de estrofes
nº de versos
1
2
Que la platja ,
sé quant de treball .
Nova cançó
- + , 1 2 3
* / . 4 5 6
( ) = 7 8 9
* 0 #


12:51
VersaCant
nº de estrofes
2
nº de versos
1
Anar enlloc .
No porta cap memòria .
Nova cançó
- + , 1 2 3
* / . 4 5 6
( ) = 7 8 9
* 0 #


12:51
VersaCant
nº de estrofes
nº de versos
3
5
Dessota dorm el bon jesús ,
guarnirem ,
fan l'enterrament ,
sense diner massa minsa ,
el brogit corresponent .
Però sospira i em tenalla ,
mitja desfeta ,
és temps d'altres veus passes ,
passo a estones ,
presents regalats per nosaltres i de sol .
Encara el fred ,
d'acomiadar la llum ,
poema de mar enllà ,
aquell misteri ,
raça fe i fem la festa magra .
Nova cançó


12:51
VersaCant
nº de estrofes
nº de versos
3
4
Al món ,
en una illa sense rumb ,
és quan llavors ,
he de passar despert .
Permeteu li marxar ,
car amic ,
quan les calàndries ja no hi ha temps per badar ,
els troncs dormen tost .
Si tu em desvetlaves ,
estic perdut enmig d'una tensió extrema ,
s'albira a un nou camí per on ,
cercar te .
Nova cançó


12:51
VersaCant
nº de estrofes
nº de versos
5
3
Si puc triar ,
una obscura temptació ,
e llavos crech crexera llur desig .
L'any vinent ,
les empreses dissortades ,
allí on són tes meses daurades .
Quan fa fred ,
has de volar ,
escriure ensems .
Aquell brunzir del rusc ,
leugereta ses ufana ,
i ara impactes al meu llit .
Joiós ,
lluitar de debò ,
qui va anunciant dolcesa .
Nova cançó


12:52
VersaCant
nº de estrofes
nº de versos
5
3
Dieu si us convé ,
els troncs i les figues sense obrir ,
besant les mans .
Un xic massa ,
això vol dir ,
invocar que caigui malferida .
Ni pertanyen a déu per tu també he deixat ,
doneu al camp ,
afartar nos bé .
Un valent dimoni ,
avança ben de pressa ,
sense amistança com creix .
Sota el desmai té cura del destí ,
com els bramuls estèrils d'un malson ,
amb galta vermella .
Nova cançó

7. Conclusions i treballs futurs

En aquest treball s'han proposat dos models per a la generació de cançons en valencià. El primer model proposat inicialment utilitzà etiquetatge POS. El segon utilitza el formalisme de les cadenes de Markov amb diccionaris invertits per a l'obtenció de les cançons amb una certa concordança sintàctica.

Per a desenvolupar aquest programa s'ha utilitzat un corpus lingüístic de poemes en valencià, obtinguts a través d'un script via web. Els poemes generats atenen a creacions artístiques ja existents.

Tot i que la implementació real era molt complexa, s'ha demostrat la viabilitat del projecte amb la utilització de diferents tecnologies fins a trobar la forma final proposada, que obté uns resultats inicials per a la solució del problema.

S'han identificat una sèrie de possibles extensions per a la millora de l'aplicació, com la introducció d'algorismes per a la producció semàntica de textos poètics més eficients, l'emmagatzematge de paraules a diccionaris invertits, que amb la utilització de probabilitats obtinguen les seqüències més coherents de paraules i també el desenvolupament de mecanismes més complexos de control al llarg de les línies de les frases compostes.

Aquest sistema extrau dades de més de quatre mil poemes per a obtenir, amb l'entrenament d'aquestes dades, els recursos lingüístics necessaris per a la composició de frases que es poden utilitzar tant en el camp de la música com en la poesia. Aquest entrenament es basa en la inclusió de les paraules clau i els seus possibles valors en diccionaris, per a l'extracció aleatòria de seqüències de paraules.

Respecte al procés basat en etiquetatge POS, tot i que es formen frases gramaticalment correctes, la implementació no pogué formar cançons correctes.

Amb el sistema de cadenes de Markov, més tard implementat, podem generar frases amb millor eficiència.

Gràcies a la implementació en Android Studio, s'ha obtingut una agradable interfície per a mòbils i tauletes.

Finalment, les funcionalitats de l'aplicació són la generació aleatòria de cançons en valencià, també podem seleccionar el nombre de versos i estrofes mitjançant els camps implementats en l'*app* i a més a més podem guardar cada cançó nova en la memòria del mòbil amb el botó de còpia.

Aquesta *app* ens permet aprofitar les cançons que podem compondre per a compartir-les o d'inspiració per a la composició de cançons pròpies. Aquestes són les funcions principals del programa versador per a cant valencià d'estil.

8. Agraïments

M'agradaria donar les gràcies a totes les persones (familiars, mestres i amics) que m'han animat a mantenir la motivació sobre la informàtica i acabar el Grau en Enginyeria Informàtica. En concret, m'agradaria donar les gràcies a Lluís Felip Hurtado Oliver, i a Ferran Pla Santamaría, per acceptar-me per treballar en aquest projecte i per tota l'ajuda que he rebut d'ells, així com els consells que m'han donat els dos per a la execució d'aquest projecte.

També m'agradaria donar les gràcies als meus companys de grup 1G per la càlida recepció que vaig rebre a l'ingressar en la universitat, al grup de la rama de computació que compartí experiències amb algunes assignatures i al grup de sistemes d`informació al que finalment pertany ara. A més a més també agrair a les meues companyes de pràctiques de la rama de sistemes Bea i Chelo per tots els bons moments a classe.

Per finalitzar, vull agrair als companys i professors de la universitat ITT Dublín per l'experiència Erasmus a Irlanda i als companys de SAP Ireland Limited que m'ha ajudat a comprendre altres formes de treball i d'estudi en altres cultures.

9. Glossari de termes utilitzats

A	B	C
Apache CouchDB: Programari de base de dades de codi obert. Apk: *Android Package Kit* és el format d'arxiu de paquet que utilitza el sistema operatiu Android per a la distribució i instal·lació d'aplicacions mòbils i middleware. App: Programari d'aplicació dissenyat per executar-se en dispositius mòbils Android Studio: Entorn oficial de desenvolupament integrat (IDE) per a la plataforma Android.	Baum–Welch algorithm: Algoritme utilitzat per a encontrar els paràmetres d'un HMM. Utilitza l'algoritme forward-backward i el seu nom prové dels matemàtics Leonard E. Baum and Lloyd R. Welch.	Cadena de Markov: Sèrie d'esdeveniments, en la qual la probabilitat que passe un esdeveniment depén de l'esdeveniment immediat anterior. Char: caràcter, lletra. Compilació just a temps (o JIT): Tècnica en la qual el codi del programa és compilat a codi de màquina nadiu en temps d'execució.
D	E	F
Deep learning: Tècnica d'extracció i transformació de noves característiques del processament de la informació, les quals poden ser de forma supervisada o no.		Freeling: Kit d'eines de processament de llenguatge natural de codi obert. Frame: Component adaptable en una línia de muntatge automatitzada de programari.
G	H	I
GIT: Sistema de control de versions per al seguiment dels canvis en els arxius d'ordinador i coordinar el treball en aquests arxius entre diverses persones. S'utilitza sobretot per al desenvolupament de programari, però es pot utilitzar per realitzar un seguiment dels canvis en els arxius.		IntelliJ IDEA: Entorn de desenvolupament Java integrat (IDE) per al desenvolupament de programes informàtics. Interface: Dispositiu o programa que permet a l'usuari comunicar-se amb un ordinador.

Java Garbage Collector: mecanisme implícit de gestió de memòria implementat en alguns compiladors i intèrprets de llenguatges de programació		
J	K	L
Java: Llenguatge de programació. JavaScript: Llenguatge script web. JVM: És un conjunt de programes d'ordinador i estructures de dades que implementen un model específic de màquina virtual. JDK: El kit de desenvolupament de Java és una implementació de la plataforma Java edició estàndard publicat per l'empresa Oracle en forma d'un producte binari dirigit als desenvolupadors de Java en Solaris, Linux, Mac OS X o Windows.		LINT tools: Utilitat d'Unix que les banderes algunes construccions sospitoses i no portàtils (que podrien tenir errors) en C codi font del llenguatge.
M	N	O
Markov: Matemàtic rus Andrei Markov (1856-1922).	NODE.JS: Codi obert multiplataforma JavaScript entorn d'execució per al desenvolupament d'una àmplia varietat d'eines de servidor i aplicacions. N-grama: Donada una seqüència, anomenem n-grama a una subseqüència de n elements.	
P	Q	R
POS "part-of-speech tagging (POS tagging)": consisteix a obtenir la categoria gramatical de cadascuna de les paraules		

que formen un text, eliminant l'ambigüitat que puguen tindre determinades paraules. Parsing: Procés d'analitzar una cadena de símbols. PageRank: Algoritme utilitzat per Google Search per classificar els llocs web en els seus resultats de cerca. Procés Poisson: En probabilitat, és un tipus d'objecte matemàtic aleatori que consisteix en punts situats aleatòriament en un espai matemàtic. Procés de Wiener: En matemàtiques, és un procés estocàstic de temps continu nomenat en honor de Norbert Wiener. Python: Llenguatge de programació d'alt nivell utilitzat per a la programació de propòsit general, creat per Guido van Rossum i llançat per primera vegada en 1991.		
S	T	U
Split: Divisió d'una cadena. String: Cadena de lletres.	Tree-adjoining grammar: Formalisme de gramàtica definida per Aravind Joshi. Tokenització: Procés de demarcació i, possiblement, la classificació de les seccions d'una cadena de caràcters d'entrada.	Unix: família de sistemes operatius d'ordinadors multitasca i usuaris múltiples que es deriven de l'original d'AT & T Unix. UX: L'experiència de l'usuari (UX) es refereix a les emocions i actituds d'una persona sobre l'ús d'un producte, sistema o servei en particular.
V	W	X

10. Referencies i bibliografia

- App versaCant disponible en:

https://sites.google.com/view/aldoapp/aldoapp Consulta: 1 Febrer 2018

- Android Developers. Google, https://developer.android.com/about/versions/android-5.0.html Consulta: 1 Maig 2017
- Android Developers. Google, https://developer.android.com/studio/intro/update.html Consulta: 1 Maig 2017
- Agnar Aamodt and Enric Plaza (1994). "*Case-based reasoning; foundational issues, methodological variations, and system approaches*", AI COMMUNICATIONS, Vol 7:1, pp 39–59 Consulta: 1 Juny 2017
- Asher Levin, David (2009). *Markov chains and mixing times*. ISBN 978-0-8218-4739-8.
- BlueJ , https://www.bluej.org/doc/documentation.html Consulta: 1 Febrer 2018
- Duffy, Scott (2003). *How to do Everything with JavaScript. Osborne.* ISBN 0-07-222887-3.
- DSIC – UPV: Octubre, 2016 https://www.prhlt.upv.es/~evidal/students/sin/Bloque2/sinB2T5/sinB2T5-2p.pdf Consulta: 1 Abril 2017
- Garside, R. (1996). *The robust tagging of unrestricted text: the BNC experience.* In J. Thomas and M. Short (eds) Using corpora for language research: Studies in the Honour of Geoffrey Leech Longman, London, pp 167-180. Versió PDF
- GitBook.com, https://talp-upc.gitbooks.io/freeling-user-manual/content/tagsets.html Consulta: 1 Febrer 2018
- J. R. Norris (1998) Markov *Chains Cambridge Series in Statistical and Probabilistic Mathematics* (No. 2) ISBN-13: 9780521633963 | ISBN-10: 0521633966.
- Martínez Jiménez, José Antonio; Muñoz Marquina, Francisco; Sarrión Mora, Miguel Ángel (2011). «Análisis morfológico. La formación de palabras: Derivación y composición». Lengua Castellana y Literatura (Akal edición). Madrid: Akal Sociedad Anónima. ISBN 9788446033677.

- Mike Sharples (June 1999). *How We Write: Writing As Creative Design*, *Routledge*. ISBN 10: 0415185874.
- M. Boden (1990). *Creative Mind: Myths and Mechanisms*, Weidenfeld & Nicholson, London. ISBN 978-0-19-875155-7.
- Noam Chomsky, (1957) *Syntactic Structures*, Mouton, The Hague. ISBN 978-3-11-021832-9.
- ThoughtCo Color Sysmbolism for Graphic Artists https://www.thoughtco.com/g00/color-symbolism-information-1073947?i10c.referrer=https%3A%2F%2Fwww.google.es%2F Consulta: 1 Juny 2017
- Poesia.cat 2017, http://www.poesia.cat/ Consulta: 1 Febrer 2018
- Real acadèmia espanyola,http://dle.rae.es/?id=DgIqVCc Consulta: 1 Febrer 2018
- WordReference.com LLC 2017, http://www.wordreference.com/es/translation.asp Consulta: 1 Febrer 2018