Màster universitari de seguretat de les TIC

Treball final de màster

"Generació de Claus criptogràfiques a través de la veu"

Estudiant: **Manel Rodríguez Recasens**

Director: **Antoni Martínez Ballesté**

Universitat: **UOC**

Data: **Juny 2019**

Resum

Català

L'objectiu d'aquest projecte és implementar un sistema d'extracció de característiques de la veu amb el propòsit de generar claus i avaluar-lo. Per això, s'ha hagut de realitzar una recerca a partir d'estudis científics de quins mètodes i quines característiques es tenen en consideració per extreure les peculiaritats de la veu. A més, s'ha posat en pràctica aquest mètode per veure quines possibilitats d'us té en un entorn quotidià. Un cop realitzada la fase de recerca i implementació se n'han estudiat la viabilitat i les febleses.

English

This project is about implementing an extraction system of voice characteristics with the aim of acquiring keys. In order to do so, a deep research from scientific studies was conducted to get which methods and what characteristics have to be taken into consideration to extract the voice peculiarities. In addition, the method has been put into practice to see what possibilities it has in our day to day. Once the research and the implementation were done, the viability and weaknesses were studied.

Índex

1. Introducció 6
1.1 Context 6
1.2 Motivació 7
1.3 Planificació 8
1.3.1 Estudi i recerca 8
1.3.2 Disseny 8
1.3.3 Implementació 8
1.3.4 Joc de proves 8
1.4 Decisions preses 9
1.4.1 Mètode escollit 9
1.4.2 "Set-up" 9
2. Marc teòric 10
2.1 Mel Frequency Cepstral Coefficients 10
1.3.2 *Que és?* 10
2.1.2 Com es calcula? 10
2.2 Altres mètodes 14
2.2.1 Linear Prediction Cepstral Coefficients 14
2.2.2 Perceptual Linear Prediction 15
2.2.3 Dynamic Time Warping 15
2.2.4 Hidden Markov Model 16
3. Disseny 17
4. Implementació 22
4.1 Registre 22
4.2 Identificació 25
5. Resultats 27
5.1 Experiment 27
6. Conclusions i treball futur 32
6.1 Conclusions 32
6.2 Treball futur 33
7. Bibliografia 34
8. Taula de continguts 36

8. Annex 37
8.1 Manual d'usuari 37
8.1.2 Obrir el programa 37
8.1.3 Enregistrar 38
8.1.2.4 Accedir 40

1. Introducció

1.1 Context

Actualment vivim en un moment en el que la intel·ligència artificial és un tema a l'ordre del dia. Cada vegada més dispositius que són quotidians incorporen característiques que permeten utilitzar la biometria per poder fer-ne un ús més agilitzat i còmode.

Gràcies a aquest fet, no és necessari haver de recordar una contrasenya diferent per cada accés per cadascun dels dispositius electrònics o aplicacions que utilitzem a diari. No només es facilita el no perdre claus, sinó que, per norma general, s'augmenta molt la seguretat.

Com a exemple, si una persona té com a contrasenya "1689M?" per poder accedir al menú principal de la seva casa intel·ligent a través d'un teclat, a partir d'algorismes de força bruta o, depenent de quina contrasenya s'hagi escollit, amb atacs de diccionari, es podria extreure quines tecles s'han de clicar per a tenir accés. En canvi, si l'usuari utilitza, ja sigui una lectura del seu iris, una gravació de la seva veu, o un sistema de reconeixement facial per poder entrar, cap altra persona ho podrà fer, ja que no pot tenir ni el seu ull, ni la seva veu, ni la seva cara.

Això no vol dir que els sistemes que utilitzin les característiques biomètriques de les persones siguin 100% eficaços i segurs. No podem afirmar que res sigui completament segur, en el cas dels sistemes que utilitzen la biometria, una escletxa per on es podria vulnerar la seguretat seria suplantant l'usuari. Per exemple, amb el reconeixement facial, es podria utilitzar una fotografia de la cara. Amb la veu, una possibilitat seria gravar la veu de la persona amb la frase indicada i utilitzar aquella gravació, de fet com es pot veure a la noticia [1] dos persones van aconseguir amb una mà de cera enganyar un sistema biomètric que el que feia era reconèixer les venes. Tots aquests possibles punts febles, com és esperable, van en funció de com de robust i acurat és el sistema en sí.

Un sistema que utilitza les característiques biomètriques de les persones ha de treballar amb un rang, no pot treballar amb xifres exactes. Això és perquè una persona pot dir "Bon dia", i quan ho torni a dir per molt exacte que ho intenti fer, sonarà, per poc que sigui, diferent, ja sigui el volum per una de les lletres en concret, l'entonació, la velocitat, etc. Hi ha molts factors i no som màquines, això fa que, per norma general, no fem els coses dos cops iguals encara que ho intentem.

El fet que aquests tipus de sistemes no puguin treballar amb valors exactes i ho hagin de fer amb rangs, fa que hi puguin haver el que s'anomenen falsos positius i falsos negatius. És a dir, que una persona que no és l'usuari esperat es doni per vàlid o, en el segon cas, que es cregui que l'usuari esperat és un intrús. És per això que com més acurat sigui el sistema i com més baix sigui el rang d'acceptació menys risc hi haurà de cometre errors a l'hora d'acceptar intrusos o de rebutjar usuaris esperats.

1.2 Motivació

Tornant al que comentàvem prèviament, cada vegada més dispositius que trobem al nostre abast fan ús d'aquestes característiques extretes dels usuaris. Un dels sistemes biomètrics més utilitzats i en augment d'ús són els que estudien la veu de les persones. Per això, aquest treball de final de màster busca estudiar i veure un exemple d'implementació fent recerca dels mètodes utilitzats per les tecnologies de l'actualitat i veure com de fiable i viable pot arribar a ser en un ambient quotidià.

El mètode implementat es posarà en pràctica amb una aplicació d'exemple, per així validar la tècnica. La implementació tindrà com a objectiu utilitzar un sol mètode, sense l'ajuda d'altres algorismes estadístics per el procés de comparació amb la plantilla. Per tal de poder validar-lo de manera individual.

1.3 Planificació

El treball es divideix en 4 punts diferenciats. Aquests, serien els següents:

1.3.1 Estudi i recerca

Un cop fet el primer contacte amb el director, comença la part d'estudi i recerca. En aquesta fase s'han buscat, principalment a la base de dades IEEE-XPLORE, estudis relacionats amb mètodes de tractament del so amb la finalitat de poder crear diferents claus a partir de la biometria de la veu.

Aquest apartat també engloba la recerca feta sobre quin llenguatge de programació es podria utilitzar, llibreries existents que ens poden ser útils, sistema operatiu i què s'utilitzaria per capturar el so.

1.3.2 Disseny

Quan s'ha tingut clar quin mètode s'utilitzaria, s'ha fet el disseny de l'esquelet de la implementació, des de la captura de so fins la l'ús de les característiques extretes a l'aplicació.

1.3.3 Implementació

Una vegada finalitzada l'etapa de disseny, ha començat la implementació. Aquest apartat està descrit extensament al punt quatre del treball.

1.3.4 Joc de proves

En aquest apartat es valida l'eina, estudiant els falsos positius i falsos negatius provant amb usuaris diferents en sexe i edat.

1.4 Decisions preses

En aquest apartat es descriuran les decisions preses per realitzar el treball.

El punt de partida del treball ha estat des de zero, pel que s'ha pogut escollir tot des del començament.

1.4.1 Mètode escollit

Com a mètode d'extracció de característiques de la veu s'ha optat per utilitzar Mel Frecuency Cepstral Coefficients (MFCC) basant-me en estudis com [2], [3] o [4].

1.4.2 "Set-up"

El llenguatge de programació utilitzat ha estat Python 3.6 de 32 bits. Es va triar Python per dos motius. Primerament perquè en cap dels documents científics trobats es feia una implementació en Python i perquè revisant quines llibreries relacionades amb el tractament de fitxers de so i de captura d'àudio es va veure viable. A més, Python és un llenguatge que actualment s'utilitza en quasi bé tots els àmbits, pel que no seria descabellat veure una implementació real fonamentada en aquest llenguatge.

Les llibreries relacionades amb el tractament dels fitxers .wav que s'han utilitzat son: "wave", "pyaudio", "scipy.io", "numpy" i "pydub".

Per l'ús de la tècnica MFCC s'ha utilitzat una llibreria fonamentada en Python anomenada "python_speech_features". El recurs utilitzat per crear la GUI de l'aplicació creada és la lliberia "tkinter". I per diferents moments la "os".

Com a micròfon s'ha utilitzat el que tenen incorporat els auriculars Creative HS-810 SB Blaze. Les característiques d'aquest son les següents:

- Resposta de freqüència: 100Hz ~18kHz
- Impedància: <2,2k ohms
- Sensibilitat: -40dBV/Pa

L'ordinador que s'ha utilitzat per realitzar el treball i per fer les proves és un ordinador de sobretaula amb:

- Processador: Intel® Core(TM) i5-6500 CPU @ 3,20GHz
- Ram: 16GB DDR4
- GPU: NVIDIA GeForce GTX 960
- Windows 10

2. Marc teòric

2.1 Mel Frequency Cepstral Coefficients

1.3.2 *Que és?*

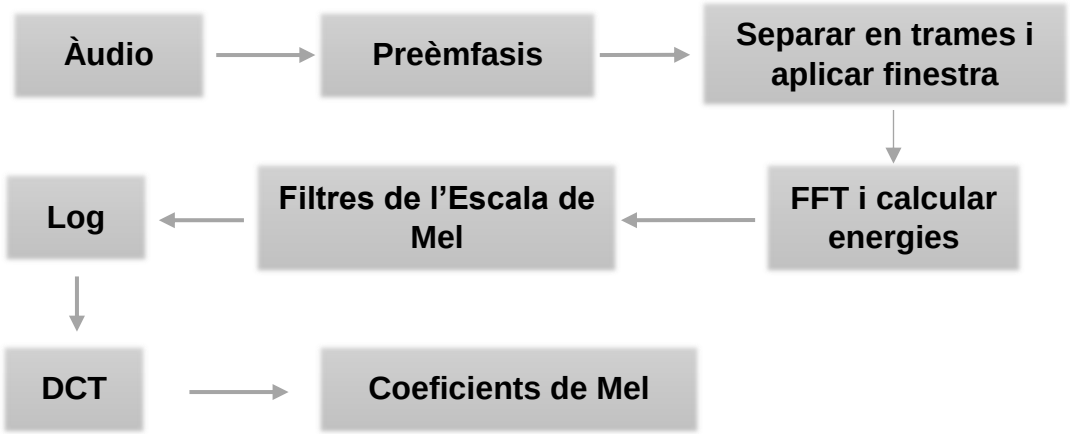
Mel Frequency Cepstral Coefficients (MFCC) és un dels mètodes més utilitzats actualment en el món del reconeixement de la veu per extreure característiques de la parla.

Com a resultat final s'obtenen uns coeficients que ens defineixen la informació essencial, deixant de banda la menys important, d'una senyal de veu.

2.1.2 Com es calcula?

MFCC tracta d'obtenir, a partir de la veu i diferents processats sobre aquesta, els coeficients de Mel, que acaben sent dotze.

MFCC es divideix en diferents punts, un esquema visual seria el que trobem a la "Imatge 1" que es pot veure a la següent pàgina:



Imatge 1: MFCC esquema

A continuació s'explicaran els passos que es poden veure dibuixats a la "Imatge 1".

a) Àudio

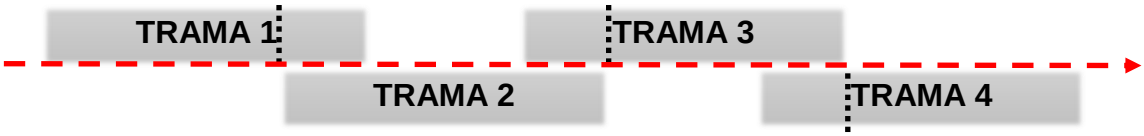
En aquest pas es passarà una senyal d'àudio.

b) Preèmfasis

Aquest pas tracta d'amplificar les freqüències altes per així millorar la relació senyal/soroll

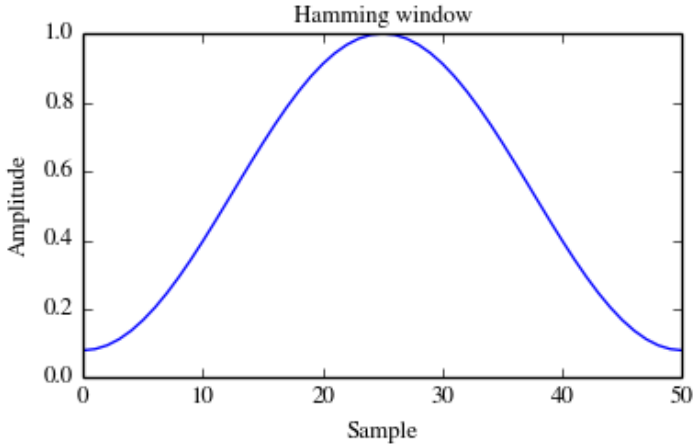
c) Separar en trames

Una senyal d'àudio canvia constantment, és per això que per fer un bon tractament d'aquesta, és necessari dividir-la en trames molt petites, ja que d'aquesta manera es pot considerar que no hi ha hagut gaire modificació en cadascuna de les trames. Normalment, la llargada d'aquestes és entre 20 i 40ms (per exemple a [5] utilitzen una finestra de 25ms). A més, les trames es trepitgen entre elles creant un solapament.



Imatge 2: Separació de la senyal per trames

Ara, per cada trama s'aplica el que és comunament conegut com "windowing". Tracta d'aplicar una finestra. Normalment Hamming és la utilitzada en MFCC. Això és per millorar l'espectre i suavitzar els laterals de la senyal per el següent pas.



Imatge 3: Finestra Hamming

Font 1: [14]

d) FFT i calcular energies

Aquest pas ja es fa per cadascuna de les trames creades anteriorment. Es calcularà la transformada de Fourier ràpida (FFT) per transformar del domini del temps al domini de freqüència. A partir de FFT es poden calcular les energies de cada trama elevant al quadrat el resultat (tant imaginari com real) obtingut. A continuació es normalitza amb una mida de finestra escollida (M), i així s'obté el periodograma.

$$Periodograma = \frac{|FFT(trama)|^2}{M}$$

Equació 1: Càlcul periodograma

e) Filtres de l'escala de MEL

Es tracta d'aplicar de 20 a 40 filtres triangulars sobre l'energia calculada per cada trama del pas anterior segons l'escala de Mel.

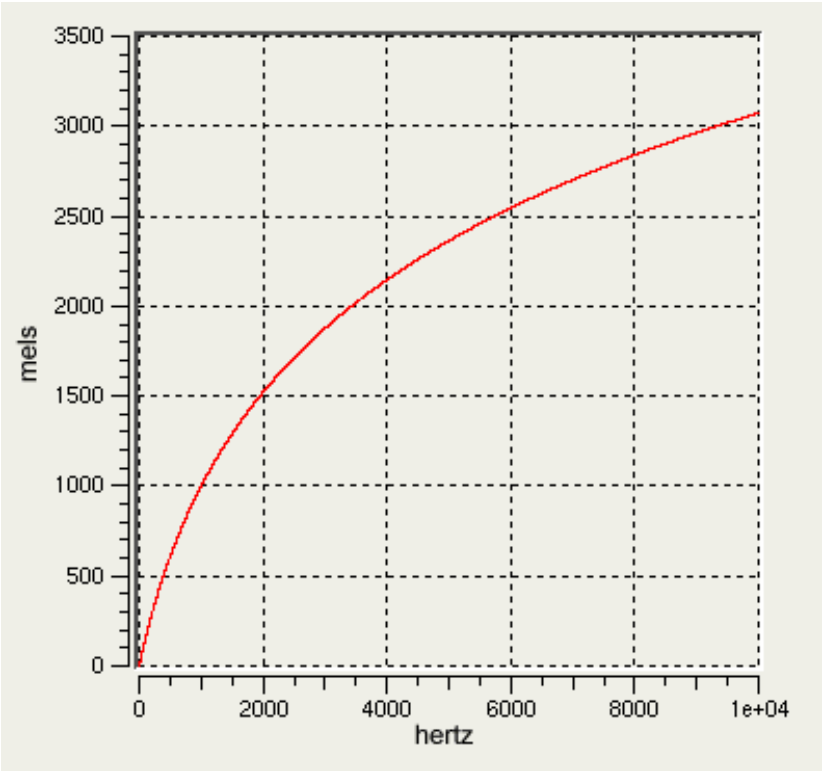
Els filtres de Mel segueixen el funcionament de la percepció del so de l'oïda humana, és a dir, les freqüències baixes tenen més pes que les altes.

L'escala de Mel es defineix com:

$$m = 2595\log 10(1 + \frac{f}{700})$$

Equació 2: Filtres de l'escala de MEL

I gràficament:



Imatge 4: Gràfica de l'escala de Mel

Font 2: [15]

f) Log

Aplicarem el logaritme en base 10 sobre el resultat obtingut a l'anterior pas.

g) DCT

Com últim pas abans d'agafar els 12 coeficients de Mel, s'aplicarà la transformada discreta de cosinus (DCT). Obtindrem 20 DCT dels 20 logaritmes realitzats per cada trama (20 com a número d'exemple però podria ser un altre). DCT serveix per decorrelar

i com a compressor de la senyal per aconseguir dades que ens donin més informació transformant així com es diu a [2] les freqüències mel al domini del temps.

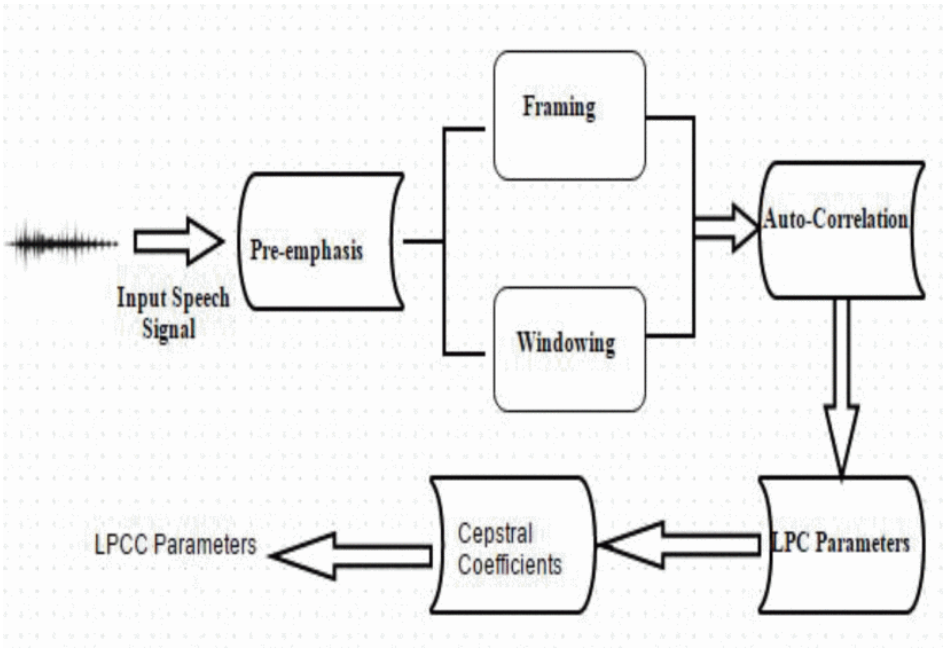
h) Coeficients de Mel

En aquest últim pas ens quedarem 12 coeficients de Mel, però es començarà per el coeficient número 1 en comptes del 0, que es descartarà. Això és perquè els altres valors representen canvis massa grans per a que ens siguin aprofitables.

2.2 Altres mètodes

2.2.1 Linear Prediction Cepstral Coefficients

Aquest mètode és derivat del Linear Predictive Coding (LPC) i l'inici d'aquest és molt semblant a l'explicat al punt 2.1 (MFCC). Comença amb una preèmfasis i continua amb la separació per trames i el "windowing". A partir d'aquí venen les diferències: es porta a terme el mètode d'LP, que es pot veure representat en el diagrama següent:



Imatge 5: Linear Prediction Cepstral Coefficients

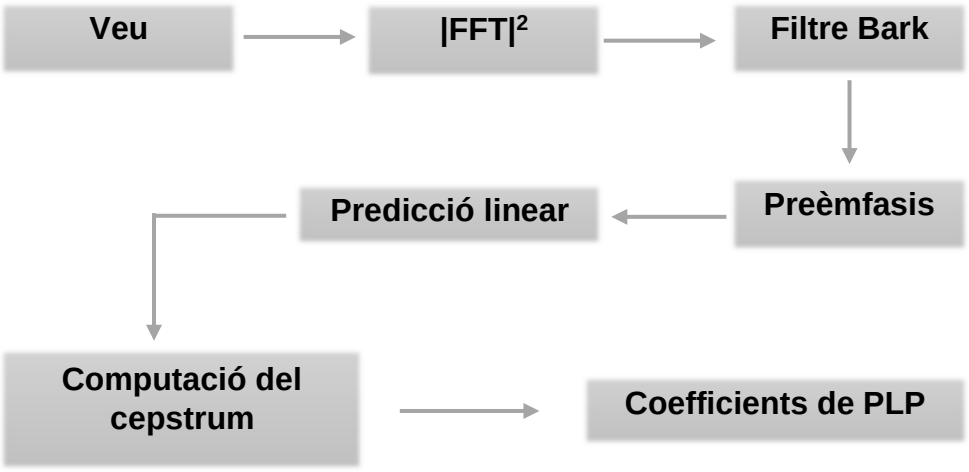
Font 3: [6]

El propòsit del LPC és fer un anàlisi de la senyal en el domini del temps i així cada mostra és calculada com una combinació lineal de les mostres anteriors a aquesta. A partir del pas LPC Parameters per traslladar-nos al mètode LPCC derivem a coeficients cepstrals.

2.2.2 *Perceptual Linear Prediction*

Perceptual Linear Prediction (PLP) és un mètode molt semblant al LPC, però es diferencia en que s'intenta apropar més al sistema auditiu de les persones.

El següent diagrama mostra tots els passos de PLP:



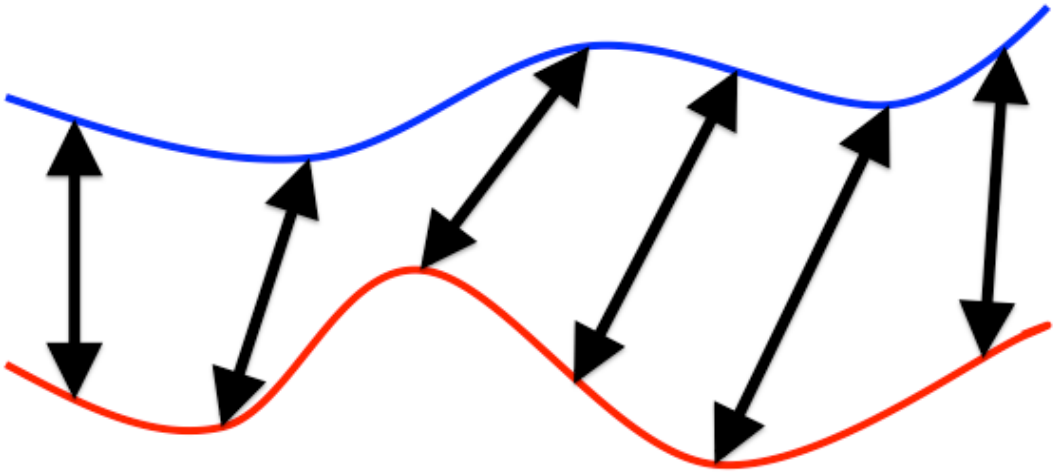
Imatge 6: Perceptual Linear Prediction

Els dos següents algoritmes no s'utilitzen per extreure característiques de la veu sinó per mesurar la similitud.

2.2.3 *Dynamic Time Warping*

Dynamic time warping (DTW) és un algoritme que tracta de veure quina similitud hi ha entre dos seqüències temporals. Com es descriu a [7] serveix "trobar el camí amb menys distorsió entre dos patrons i donar la distancia entre aquests dos patrons". DTW s'ha fet servir molt per reconeixement de veu i sobre tot per contrarestar el problema de que entre la plantilla i la entrada amb la que s'ha de comprovar hi hagin diferents velocitats.

A la imatge 7 es pot observar un exemple visual de DTW:



Imatge 7: Dynamic Time Warping

Font 4: [16]

2.2.4 *Hidden Markov Model*

. Hidden Markov Model (HMM) és un model estadístic que tracta els fonemes de cada paraula. HMM treballa amb màquina d'estats i estadística, aquest model s'ha d'entrenar per així quan se l'hi doni un fragment d'un àudio pugui emparellar els fonemes amb la plantilla i pugui acabar sabent de quina paraula es tracta, endevinant quins són els estats "ocults". S'ha de crear un HMM per cadascuna de les paraules ja que els estats són diferents entre paraules. El problema que té és que necessita que la fase d'entrenament sigui molt exhaustiva per a que els resultats siguin correctes.

3. Disseny

En aquest apartat s'explicarà a alt nivell el disseny de la nostra implementació:

Dels mètodes vistos a anteriorment finalment es va escollir implementar d'entre els tres primers MFCC. Aquesta decisió va ser fonamentada per dos motius:

- El primer va ser basat en l'estudi [8] ja que es treu com a conclusió que per el reconeixement de veu PLP i MFCC donen uns millors resultats, que no pas LPC.
- I com a segon, per decidir si utilitzar PLP o MFCC; es va veure que MFCC va ser estandarditzat per la European Telecomunications Standards Institute (ETSI) al [10] per la qual cosa es va creure que seria una bona opció per posar en pràctica en el nostre entorn, sense l'ajuda de cap algorisme ni estadístic ni que ajudi a normalitzar la identificació amb la plantilla.

Per tant a grans trets la nostra implementació serà:

Imatge 8: Aplicació a alt nivell

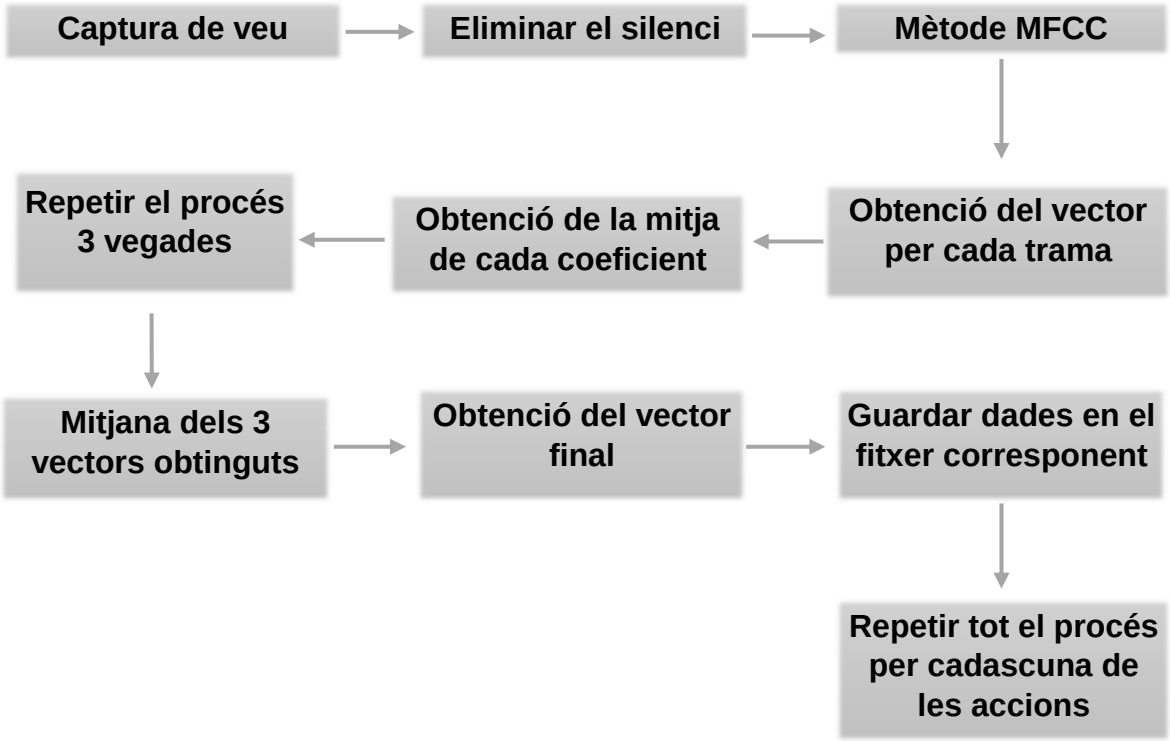
El disseny inicial arribava fins a la identificació entre diferents usuaris però es va investigar per ampliar aquest disseny inicial i així poder treure més conclusions per validar el sistema. Es va plantejar que es pogués encriptar i desencriptar un fitxer de text utilitzant el vector extret per cada persona com a clau per encriptar i desencriptar. D'aquesta manera s'hauria millorat la seguretat ja que no hagués calgut emmagatzemar la clau per cada persona com a plantilla, sinó que s'hauria comparat que el fitxer resultant era el correcte. El problema que es va trobar va ser que, com s'ha dit prèviament, una persona no repetirà sense cap petita modificació més d'un cop una

frase, i no només això, sinó que l'ambient en el que s'ha desenvolupat el treball és un ambient quotidià per la qual cosa el micròfon no és de gama professional i agafa molts sorolls externs que amb altres recursos tecnològics es podrien eliminar. Això fa que sigui necessari treballar amb un llindar i no amb xifres exactes per lo que aquesta idea es va descartar.

Després es va plantejar la idea que finalment es va implantar: utilitzar un submenú per validar la distinció entre diferents frases d'un mateix usuari. Tracta d'un menú que simula el que podria ser una casa domòtica.

Per tant, més específicament, el disseny del sistema està dividit en dos fases: la fase de registre i la fase d'identificació, que estan descrites a continuació:

- **Registre**



Imatge 9: Fase de registre

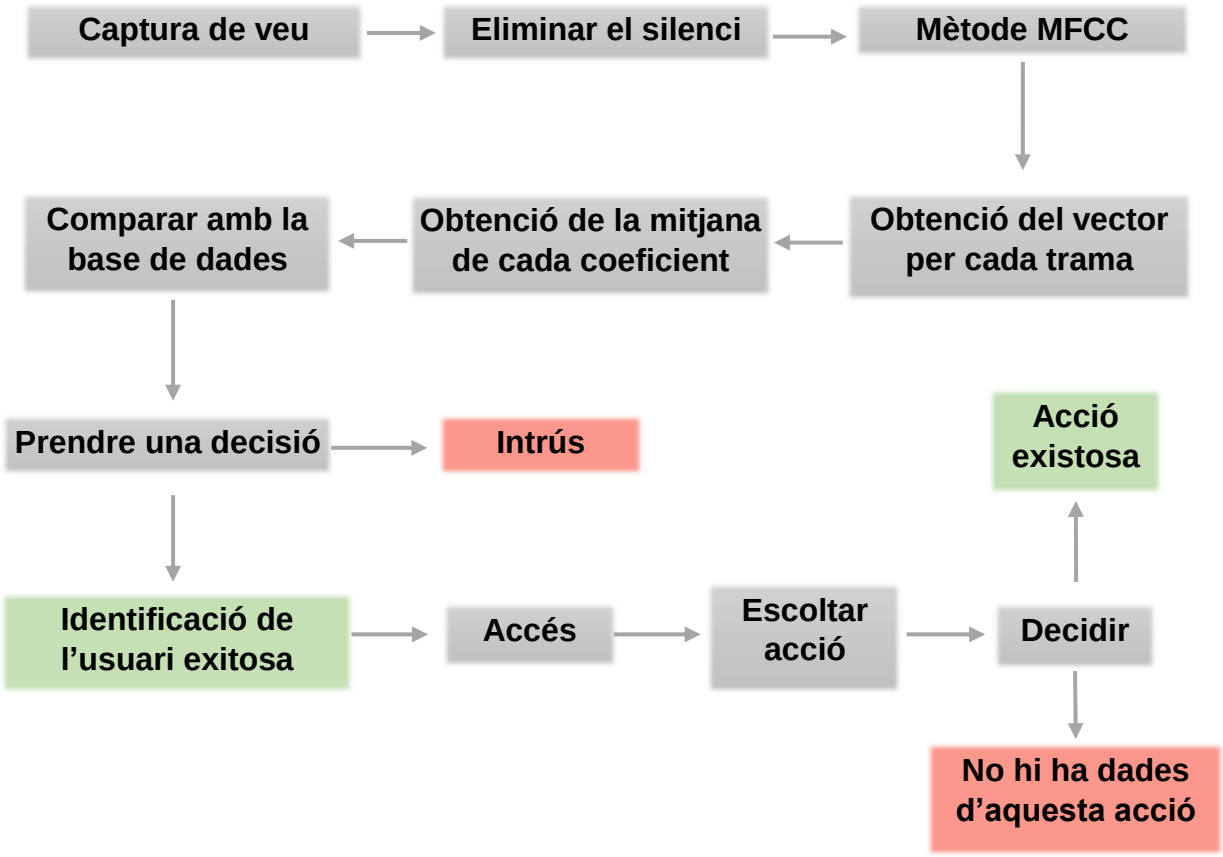
Primerament es capturarà la veu. Un cop aquesta s'hagi guardat, s'ha d'eliminar el que a nosaltres no ens serveix: el silenci. La decisió de si hi haurà o no silenci serà a partir d'un llindar que se li donarà per a que es tracti o no aquell fragment. Un cop es té guardada la veu es procedirà a realitzar el mètode pròpiament dit d'MFCC descrit en el punt 2.1 del treball. Com a resultat d'aquest mètode s'obtindrà un vector de 12 coeficients per cadascuna de les trames. Aquí és on hi ha les principals diferències amb la majoria de les implementacions de sistemes de reconeixement de veu que utilitzen el Mel Frequency Cesptral Coefficients. Davant el problema que ens sorgiria al comparar trama per trama a la identificació en el cas que aquestes no anessin exactament igual podria passar que la trama 20 de la mostra agafada per identificar, pot sense cap mena de problema, no ser la 20, sinó que tranquil·lament pot ser la 25 ja que és molt fàcil que es varïi la velocitat en la que es repetirà la frase clau i/o l'allargada temporal de la clau. Com a solució, encara que no tant eficient com utilitzar per exemple el DTW explicat al punt 2.2.3 (com per exemple es fa al estudi [11]) es farà una mitjana de cada un dels 12 coeficients per cada trama, és a dir que es sumarà cada Xi coeficient de cada trama i es dividirà per el nombre total de trames creades a partir de l'àudio d'entrada sense el silenci. D'aquesta manera s'obtindrà un vector resultant de 12 valors per tot el fragment analitzat.

A continuació, es repetiran els anteriors passos i es farà una mitjana de cada coeficient del vector resultant de les tres vegades que s'ha realitzat el procediment. Això es fa perquè una persona no repetirà mai dos cops exactament de la mateixa manera una frase clau, per lo que es vol agafar una mitjana de com es diu normalment per a que hi hagi més possibilitats de que el sistema la reconegui a la fase d'identificació.

Finalment es guardarà la clau d'entrada per aquella persona al fitxer corresponent.

Un cop tenim les claus per cada individu guardades es farà el mateix procediment per les tres accions que es poden guardar per persona i s'emmagatzemarà en un fitxer individual per cada usuari la clau per cada acció. És a dir, que guardarem la plantilla per cada possible acció per cada individu per així comparar-ho a la fase d'identificació.

- **Identificació**



Imatge 10: Fase d'identificació

En aquesta fase es compararà un nou fragment d'entrada amb les plantilles guardades.

Una part d'aquesta fase comparteix passos amb l'anterior. Aquestes fases són el capturar la veu de la persona a partir d'un micròfon, l'eliminació del que considerem silenci, l'aplicació del mètode MFCC per obtenir el vector resultant per cada trama i el realitzar la mitjana per cada coeficient. És en aquest punt que comença a diferir el procés.

Arribats en aquest punt, s'agafarà el vector resultant que és la clau que s'utilitza com entrada al sistema i es compararà amb les claus guardades a la base de dades. Es mirarà a partir de la comparació quina clau no sobrepassa els límits que considerem com un intrús i en cas que hi hagués èxit amb més d'una clau es mirarà amb quina hi ha menys diferència. S'ha d'intentar que el llindar límit seleccionat sigui l'idoni per a que aquesta última situació no es doni ja que vol dir que hi hauria possibilitats de tenir falsos positius.

A partir de l'anterior pas s'extraurà una decisió; és intrús o és un usuari guardat? En cas que sigui intrús es negarà l'entrada a aquest individu i si finalment la decisió és positiva se li permetrà l'accés. Un cop dins, es tindrà la opció de fer que l'aplicació escolti una acció de l'usuari. Per tant, quan l'usuari digui què vol fer, el sistema a partir de la senyal d'àudio rebuda extraurà una clau i la compararà amb les claus guardades per aquell usuari per veure si alguna d'aquestes coincideix dintre del llindar especificat. Amb el mateix procediment que per l'accés principal al sistema, en cas que més d'una acció estigui dintre del llindar acceptable es mirarà quina acció té les distàncies més petites amb la clau actual.

Com a sumari del que es vol aconseguir amb aquest disseny podem trobar els següents punts:

- Els usuaris podran enregistrar la seva veu com a clau d'accés al sistema.
- Podran crear claus biomètriques a partir de la veu en relació a diferents accions que tindran la possibilitat de dur a terme quan hagin iniciat sessió en el sistema.
- Podran accedir a l'escenari creat repetint la frase d'enregistrament
- I finalment podran activar diferents accions que s'han guardat a la fase de registre.
- Les accions disponibles seran: "Porta garatge", "Música ambiental" i "Aire condicionat"

4. Implementació

En aquest apartat es descriurà en detall la implementació del nostre sistema:

4.1 Registre

Començarem per la fase d'enregistrament:

El que es detallarà a continuació, tal com s'ha mencionat a l'apartat de disseny, es realitzarà tres vegades.

Primer capturarem la veu i ho guardarem en un fitxer anomenat "Voice.wav". El sampleRate escollit ha sigut 16000. S'ha cregut que era una bona opció que la durada de la gravació fos de cinc segons.

Un cop es té la veu enregistrada, s'ha d'eliminar el silenci, i això ho farem de la següent manera:

Utilitzant la llibreria Pydub per tractar el fitxer començarem a llegir el fitxer des del principi i analitzarem si la intensitat està per sota d'un mínim que se li ha assignat. En aquest cas, després de fer varies proves es va veure que un threshold adient seria -30. Per tant anirem guardant el fragment d'àudio que no superi aquest llindar especificat.

Farem el mateix però pel final, per eliminar el silenci que hi ha quan l'usuari deixa de parlar. Això ho farem utilitzant aquesta llibreria per donar la volta al fragment d'àudio i començar a llegir pel final.

```
if (len(audio) != silence_begining) and len(audio) > silence_begining:
    all_silence = False
    silence_end = self.silence(audio.reverse())
```

Fragment codi 1: Audio reverse

I realitzarem el mateix pas que hem fet amb el silenci del inici. Un cop fet això guardarem un fitxer d'àudio en el que només conservarem la part en que l'individu parla.

```
final_audio = audio[silence_begining:start_silence_end]

final_audio.export("final_file.wav", format = "wav")
```

Fragment codi 2: Guardar fragment sense silenci

La forma d'extreure el silenci va ser basada en [12].

Es va creure que no era necessari eliminar el silenci que hi hagués en alguna pausa provocada per la parla del individu, perquè això podria formar part de com s'expressa la persona.

Un cop hem retallat l'entrada per micròfon per obtenir quan la persona està parlant, començarem a analitzar-ho.

Llegirem el fragment que ens interessa i cridarem el mètode MFCC amb aquest:

```
(rate, sig) = wavfile.read('final_file.wav')
mfcc_feat = mfcc(sig,rate, winfunc=numpy.hamming, nfilt = 32);
```

Fragment codi 3: Crida MFCC

Com es pot veure direm que es vol utilitzar la finestra Hamming. Tornant a l'explicació de l'apartat 2.1, la nostra mida de finestra serà 25ms i un solapament de 10ms. Aquests valors han estat basats en la implementació [5].

Seguint els consells de la implementació utilitzada a [3] s'anava a utilitzar una longitud de 256 per la finestra de FFT però en el nostre cas tenim més trames per la qual cosa ha estat necessari utilitzar una longitud de 512 mostres.

Per el número de filtres Mel utilitzats s'han seguit els resultats obtinguts en [13] en el que s'arriba a la conclusió que 32 filtres Mel triangulars són els que donen millor resultat conjuntament utilitzant la finestra Hamming.

Cada vegada que s'acaba el procés MFCC es guardarà el vector resultant i s'eliminaran els dos fitxers d'àudio creats, com a precaució per no deixar més informació sensible de la necessària:

```
os.remove("voice.wav")
os.remove("final_file.wav")
```

Fragment codi 4: Eliminar fitxer sensibles

Un cop tinguem els tres vectors resultants guardats repetint com s'ha dit el procés tres vegades farem la mitjana per així aconseguir el nostre vector dels coeficients de Mel final:

```
for v in range(0, len(vector)):
    for c in range(0, len(vector[v])):
        final_vector[c] += vector[v][c]

for i in range(0, len(final_vector)):
    final_vector[i] = final_vector[i]/len(vector)
```

Fragment codi 5: Obtenció del vector final

I es guardarà al fitxer "/private/24üestions_H.txt". La informació estarà separada per ":" i aquesta serà "ID d'usuari:Nom d'usuari:Clau Biomètrica". La clau estarà separant els coeficients amb ";", això ens facilitarà el procés quan llegim el fitxer. Un exemple seria:

```
1:pere:-24.43456031896163;-26.388924794185467;21.146471967604114;-21.67646....
```

Imatge 11: Fragment fitxer d'entrada

Automàticament es procedirà a mostrar una nova pantalla on l'usuari haurà d'enregistrar la clau per cadascuna de les accions que es contempla al sistema. Aquestes són "Porta garatge", "Música ambiental" i "Aire condicionat", i es realitzarà el mateix procediment descrit abans en el procés de creació de la clau d'entrada al sistema.

La diferència, però, serà que per cada usuari hi haurà un fitxer específic per les claus biomètriques de la seva pròpia veu per les diferents accions.

Aquest fitxer tindrà per nom l'ID de l'usuari i l'interior serà "Acció:Clau Biomètrica". Com es veu, les dos parts estan separades per ":". Cada acció estarà en una línia diferent del fitxer de text.

```
Música ambiental:-20.739677546996884;-18.77840950636625;21.4043...
Aire condicionat:-19.484261310623328;-26.39184301040036;22.73453...
```

Imatge 12: Fragment fitxer d'accions

El valor ID de l'usuari acaba sent una enumeració dels usuaris guardats en el nostre sistema, és a dir, el primer usuari que s'enregistri tindrà assignat l'ID "0". El nom, en canvi, l'escull cada individu escrivint-lo en el procés d'enregistrament de la veu per l'accés al sistema.

És en aquest punt que finalitza el procés d'enregistrament.

4.2 Identificació

El procés d'identificació, a l'inici, té moltes semblances amb el que seria l'esquelet del registre.

El procés d'enregistrament de la veu i extracció del vector final no difereix de l'explicat en l'apartat anterior, simplement que aquest procediment es realitzarà únicament un cop i no tres. Ara no cal entrenar el sistema ja que no es tracta de crear la plantilla si no que s'ha d'extreure una única empremta per així comparar amb la clau guardada.

Un cop tenim el vector extret s'ha de comparar amb la nostra base de dades. Primer es compararà amb tots els usuaris guardats en la nostra base de dades i es mirarà que la diferència entre cada coeficient no sigui major que el llindar fixat, en cas que la resta entre dos coeficients superi el threshold aquesta plantilla es descartarà. Si finalment tots els coeficients estan dins del llindar, voldrà dir que no es tractarà com un intrús i es guardarà un vector que contindrà les diferències entre cadascun dels valors.

```
for v in range(0, len(vectors)):
    valid = True
    vector_dif = []
    for i in range(0, len(vectors[v][key_word])):
        vector = vectors[v][key_word]
        dif = float(vector[i]) - vector1[i]
        vector_dif.append(abs(dif))
        if (abs(dif) > threshold_enter):
            valid = False
    if valid == True:
        valid_list[v] = vector_dif
```

Fragment codi 6: Comparació amb plantilles guardades

Ara s'ha de veure d'entre les plantilles que s'han catalogat de vàlides quina té les distàncies més curtes. Així que a continuació es mirarà quina és la plantilla més adient i es retornarà com la escollida, juntament amb l'usuari.

```
for valid_v in valid_list:
    actual_dif = 0
    for difference in valid_list[valid_v]:
        actual_dif += difference
    if best[1] > actual_dif:
        best[1] = actual_dif
        best[0] = valid_v
        valid = True
return valid, best[0]
```

Fragment codi 7: Plantilla amb les distàncies més curtes

Si el resultat ha estat negatiu s'informarà que no hi ha ningú enregistrat amb aquesta clau biomètrica i en cas que sigui afirmatiu l'usuari tindrà accés a l'interior del sistema.

En aquest punt ve la segona part de la identificació: escoltar l'acció.

El sistema escoltarà quina acció vol realitzar l'usuari i es repetiran els passos descrits per extreure la clau biomètrica d'aquesta. El procediment seguit a continuació és molt semblant al que es realitza per entrar. La diferència, però, és que la comparació no és entre les claus d'accés de tots els usuaris sinó que és individual amb el fitxer d'aquell individu, comparant així les accions enregistrades.

De la mateixa manera que abans, es guardaran els vectors de diferències d'aquelles accions que es donin per vàlides (les que cap distància entre coeficients de la plantilla i l'entrada donada superi un llindar especificat). A continuació, es mirarà quina és la que

té més semblances amb el fragment d'àudio donat i s'informarà quina acció s'ha realitzat en cas que se n'hagi portat a terme alguna.

5. Resultats

5.1 Experiment

Les proves realitzades s'han fet amb 6 usuaris. Les dades dels usuaris son les següents:

USUARI	EDAT	SEXE
0	25	Home
1	58	Home
2	30	Home
3	57	Dona
4	29	Dona
5	22	Dona

Taula 1: Usuaris experiment

Com podem veure s'ha volgut que hi hagués el mateix nombre d'homes com dones i que hi hagués la presència d'individus de diferents edats.

La metodologia realitzada ha estat registrar les claus per cada usuari tant per l'accés al sistema com per les accions un cop dins. Les claus extretes queden detallades a la Taula 2 que apareix a la següent pàgina.

Entrada al sistema per cada usuari					
0	**1**	**2**	**3**	**4**	**5**
-16,394	-14,137	-22,715	-18,243	-22,427	-16,020
-17,101	-15,461	20,088	-8,034	-12,486	-21,155
24,038	18,433	23,148	-0,448	17,51	17,024
-16,095	-16,457	-32,797	-11,653	-29,062	-38,0315
-14,140	-11,779	1,169	-11,176	-3,215	-5,052
-1,0438	14,111	9,622	-6,438	-3,483	-8,825
-10,680	-14,401	-20,196	0,885	-9,401	-20,031
-4,501	-7,754	2,610	-11,019	-4,232	-4,44
-7,910	1,486	-2,461	4,698	2,907	-6,983
-1,951	-10,142	2,192	-0,757	-3,082	-3,372
5,385	0,059	-3,889	-5,597	-6,074	-15,373
-10,297	-8,705	-4,007	3,371	3,023	6,811

Taula 2: Clau entrada dels usuaris

A continuació es mostraran les claus de les accions per cadascun dels usuaris:

PG és "Porta garatge", MA és "Música ambiental" i AC és "Aire condicionat"

Usuari 0			Usuari 1		
PG	**MA**	**AC**	**PG**	**MA**	**AC**
-12,708	-17,444	-19,691	-15,174	-17,970	-17,385
-26,463	-18,053	-19,889	-31,295	-17,379	-16,579
0,995	18,602	21,028	10,295	24,461	22,438
-21,378	-12,955	-13,954	-17,945	-16,182	-10,687
-3,569	-6,663	-3,015	3,147	-10,472	-8,912
-1,257	-6,585	-6,034	21,546	13,296	7,445
2,308	-3,206	-1,018	-10,984	-9,940	-5,112
0,966	2,245	10,279	0,764	-0,995	6,405
16,715	-0,901	0,574	20,991	11,582	8,510
-5,074	1,497	0,719	-9,284	-10,686	-6,871
-2,935	7,451	9,222	5,268	4,045	5,729
-11,042	-10,310	-8,401	-11,142	-6,060	-2,581

Taula 3: Claus d'accions usuaris 0 i 1

Usuari 2			Usuari 3		
PG	**MA**	**AC**	**PG**	**MA**	**AC**
-17,459	-21,784	-22,460	-17,934	-11,443	-12,512
-32,411	-21,581	-21,860	-7,738	-0,825	-3,908
19,377	26,688	28,666	-2,102	3,635	5,276
-36,244	-33,580	-29,763	-6,698	-2,903	-5,999
0,402	-0,397	-5,589	-4,244	-6,432	-7,561
7,355	-3,079	-2,309	-1,516	3,021	0,734
-18,658	-16,305	-12,903	8,141	-0,854	1,620
17,876	11,491	10,404	3,186	-1,536	-4,092
7,536	-4,557	-6,113	2,941	3,813	0,545
2,999	7,868	9,849	-3,121	-7,887	-7,659
-11,050	-2,603	2,759	1,071	-0,400	-0,106
-3,641	-7,256	-3,023	3,864	3,809	3,995

Taula 4: Claus d'accions usuaris 2 i 3

Usuari 4			Usuari 5		
PG	**MA**	**AC**	**PG**	**MA**	**AC**
-16.919	-20.539	-22.597	-15,950	-14,541	-19,811
-14.283	-14.436	-20.295	-32,534	-24,370	-18,645
5.788	3.188	13.634	4,102	10,663	12,249
-8.762	-15.729	-25.310	-27,769	-31,731	-25,052
5.156	4.385	-4.410	3,769	2,515	1,383
-2.103	-7.632	-5.362	-7,354	-11,250	-7,831
3.274	-5.180	-0.660	-5,505	-15,999	-7,870
10.707	4.222	0.620	3,844	-1,459	-3,270
2.666	2.754	-1.349	-8,310	-9,491	-9,433
-2.838	-5.870	-5.914	-11,101	-8,653	-4,773
-2.496	-5.582	-5.805	-4,176	-10,550	-7,371
3.026	1.362	-0.116	8,450	11,415	5,788

Taula 5: Claus d'accions usuaris 4 i 5

Un cop registrats s'ha fet que cada usuari intenti entrar al sistema cinc vegades. El llindar fixat per aquesta proba ha estat de 10 punts per sobre i per sota de la plantilla de cada usuari. En aquesta prova hi ha hagut un 100% d'èxit: cada usuari ha entrat a la primera i no ha estat confós amb cap altre usuari.

Suposant el pitjor cas, que seria que una persona que no està enregistrada intentés entrar, en alguns casos ens hem trobat que sí que podria.

A continuació es procedeix a analitzar-ho:

Tenint en compte les plantilles extretes a la fase d'entrenament, ningú pot entrar en nom de ningú a la sessió de l'altra persona. Però com hem dit, normalment, sense cap post tractament la biometria no treballa amb números exactes, sinó que per poca variació que hi hagi cada vegada es reben diferents valors. És per això que si alguna de les claus està molt a prop una de l'altra en distància de números, pot donar-se el cas que en alguna ocasió un altre usuari surti de la seva mitjana i hi hagi una variació suficient com per a que pugui donar-se la situació en que aquella empremta sigui erròniament identificada com un altre usuari.

En el nostre cas, hem tingut un 100% d'encerts al entrar, això és perquè com s'ha explicat, primer es mira tota la base de dades i es mira quines plantilles tenen una diferència dintre del llindar acceptat. Un cop tenim les que poden ser correctes, es fa una segona ronda per veure quina té més semblança amb la actual i és llavors que s'escull la definitiva.

Investigant els resultats s'ha trobat que hi ha hagut zero falsos negatius, però si que s'han trobat falsos positius.

En el procés d'entrada a la plataforma s'ha trobat un fals positiu, això ha succeït quan l'usuari 1 ha intentat entrar, s'ha identificat correctament, però si aquest no hagués estat enregistrat s'hauria identificat com usuari 0.

En cas que s'hagués fet amb un threshold de 8, els resultats que s'haurien obtingut és que hi haurien zero falsos positius. En canvi, però, ens trobem que hi ha dos falsos negatius, aquests han estat amb l'usuari 4.

Threshold = 10		**Threshold = 8**	
Fals positiu	Fals negatiu	Fals positiu	Fals negatiu
3,33%	0%	0%	6,66%

Taula 6: Comparació de dos thresholds a l'entrada

Per tant encara que tinguem una probabilitat de més encerts amb el llindar limitat a deu. És preferible tenir pocs falsos negatius que no pas tenir falsos positius perquè és millor haver de repetir l'entrada, que no que una persona es pugui fer passar per una altra.

Per l'anàlisi de la identificació de diferents accions donades per un mateix usuari, cadascun dels sis individus han intentat activar tres vegades "Porta garatge", tres vegades "Música ambiental" i tres més "Aire condicionat". En aquest cas el threshold ha estat més restrictiu perquè hi havia menys variacions entre les diferents accions. Aquest ha estat fixat a 8 en un principi però s'ha vist que amb tot i això el llindar es trepitjava entre accions en cinc ocasions de les 18. I en cas que es baixés el threshold a 7 s'ha vist que passava en dues ocasions. Com que els valors que s'han rebut no han estat suficientment exactes com per baixar-ho més per a que els llindars no es solapin, s'ha volgut fer l'experiment així i veure quins resultats s'obtenien, perquè sinó els falsos negatius haurien crescut massa.

Els resultats han estat els següents:

Threshold = 7	
Falsos positius	Falsos negatius
22,22%	20,37%

Taula 7: Threshold 7 a l'entrada

Com es pot veure els resultats han estat molt pitjor que per l'accés.

6. Conclusions i treball futur

6.1 Conclusions

A l'inici del treball els objectius que ens plantejàvem eren els següents:

1- D'una banda fer una recerca dels diferents mètodes utilitzats per l'extracció de característiques de la veu.
2- Realitzar una implementació utilitzant un d'aquests mètodes en un ambient quotidià i,
3- Comprovar la seva validesa.

Durant la realització del treball es van dividir els escenaris d'ús del mètode triat en dos:

- Per una banda, la identificació d'usuaris entre altres individus.
- Per una altra banda, la diferenciació de frases pronunciades per un mateix usuari.

Les conclusions a les que podem arribar un cop finalitzat el treball serien les següents:

- La recerca realitzada al voltant dels diferents mètodes disponibles per l'extracció de les característiques de la veu va donar com a resultat l'elecció del MFCC.
- Finalment s'ha aconseguit un escenari quotidià en el que s'ha utilitzat el mètode ja esmentat per la creació i validació de claus biomètriques extretes de la veu.
- S'ha posat en pràctica per poder treure conclusions relacionades amb l'eficàcia del mètode en els dos escenaris plantejats anteriorment.

Les conclusions extretes un cop hem validat el mètode amb l'ajuda de l'eina creada son les següents:

- En l'escenari d'identificació de diferents usuaris, l'ús del MFCC ha donat uns resultats positius. Pel que podríem dir, que per aquest cas, utilitzar aquest únic mètode seria un camí adequat.
- Per contrapartida, a la fase d'identificació de frases pronunciades per un mateix usuari, es va concloure que utilitzar l'MFCC com a únic mètode per tractar les

característiques de la veu no era una bona decisió ja que la presència d'excessius falsos positius i falsos negatius no ens permetia arribar al nivell de precisió que considerem necessari per a un bon funcionament.

6.2 Treball futur

Assolits els objectius del present treball només ens queda plantejar-nos quines serien les següents accions que es podrien realitzar en investigacions futures:

- Primerament hauríem de tractar l'aspecte de la seguretat. El que es vol dir amb això és que les claus extretes es podrien emmagatzemar en una base de dades externa al sistema. Una altra opció seria fer ús d'algun algorisme estadístic i entrenar el sistema per poder guardar aquestes claus de forma encriptada i mirar d'implementar un codi corrector d'errors com podria ser Viterbi.

- Tanmateix es podria realitzar la mateixa implementació però evitant tractar tots els coeficients a la vegada, afegint l'ús de DTW i així comparar els resultats que s'obtindrien seguint aquesta metodologia amb els obtinguts en aquest treball.

- Finalment, des del punt de vista dels recursos materials, es podrien comparar els resultats obtinguts amb diferents games de micròfons, veient així si aquest ha tingut un impacte significatiu en els resultats extrets.

7. Bibliografia

[1]https://www.20minutos.es/noticia/3527730/0/hackeo-reconocimiento-biometrico-venas-mano/

[2] L.P. Garcia-Perera, J.C. Mex-Perera, J.A. Nolazco-Flores (13 Juny 2005). "Multi-speaker Voice cryptographic key generation".

https://0-ieeexplore-ieee-org.cataleg.uoc.edu/document/1387084

[3] Atik Charisma, M. Reza Hidayat, Yuda Bakti Zainal (08 Febrer 2018) "Speaker recognition using mel-frequency cepstrum coeficients and sum square error";

https://0-ieeexplore-ieee-org.cataleg.uoc.edu/document/8284159

[4] Hongyu Xu, Xia Zhang, Liang Jia (25 Juny 2012) "The extraction and simulation of Mel frequency cepstrum speech paràmetres"

https://0-ieeexplore-ieee-org.cataleg.uoc.edu/document/6223385

[5]Juan A. Nolazco-Flores ; Roberto A. Aceves L. ; L. Paola Garcia-Perera (07 Octubre 2010) "Speech Magnitude-Spectrum Information-Entropy (MSIE) for Automatic Speech Recognition in Noisy Environments"

https://0-ieeexplore-ieee-org.cataleg.uoc.edu/document/5597871

[6]Harshita Gupta ; Divya Gupta. (11 Juliol 2016). "LPC and LPCC method of feature extraction in Speech Recognition System"

https://0-ieeexplore-ieee-org.cataleg.uoc.edu/document/7508171

[7] Nishanth Ulhas Nair ; T.V. Sreenivas. (27 Gener 2009). "Multi Pattern Dynamic Time Warping for 34üestions speech recognition"

https://0-ieeexplore-ieee-org.cataleg.uoc.edu/document/4766617

[8] Usha Sharma ; Sushila Maheshkar ; A. N. Mishra. (13 Juliol 2015). "Study of robust feature extraction techniques for speech recognition"

https://0-ieeexplore-ieee-org.cataleg.uoc.edu/document/7154944

[9] ETSI ES 202 050 V1.1.5 (2007-01)

[10]ETSI ES 201 108 V1.1.3 (2003-09)

[11] Bhadragiri Jagan Mohan ; Ramesh Babu N. (19 Juny 2014). "Speech recognition using MFCC and DTW"

https://0-ieeexplore-ieee-org.cataleg.uoc.edu/document/6838564

[12] https://stackoverflow.com/questions/29547218/remove-silence-at-the-beginning-and-at-the-end-of-wave-files-with-pydub

[13] Kritagya Bhattarai ; P. W. C. Prasad ; Abeer Alsadoon ; L. Pham ; A. Elchouemi. (15 Maig 2017). "Experiments on the MFCC application in speaker recognition using Matlab"

https://0-ieeexplore-ieee-org.cataleg.uoc.edu/document/7926796

[14]https://docs.scipy.org/doc/scipy0.15.1/reference/generated/scipy.signal.hamming.html

[15] https://es.wikipedia.org/wiki/Escala_Mel

[16]https://www.mathworks.com/matlabcentral/fileexchange/43156-dynamic-time-warping-dtw

8. Taula de continguts

Imatge 1: MFCC esquema 11
Imatge 2: Separació de la senyal per trames 11
Imatge 3: Finestra Hamming 12
Equació 1: Càlcul periodograma 12
Equació 2: Filtres de l'escala de MEL 13
Imatge 4: Gràfica de l'escala de Mel 13
Imatge 5: Linear Prediction Cepstral Coefficients 14
Imatge 6: Perceptual Linear Prediction 15
Imatge 7: Dynamic Time Warping 16
Imatge 8: Aplicació a alt nivell 17
Imatge 9: Fase de registre 18
Imatge 10: Fase d'identificació 20
Fragment codi 1: Audio reverse 22
Fragment codi 2: Guardar fragment sense silenci 23
Fragment codi 3: Crida MFCC 23
Fragment codi 4: Eliminar fitxer sensibles 24
Fragment codi 5: Obtenció del vector final 24
Imatge 11: Fragment fitxer d'entrada 24
Imatge 12: Fragment fitxer d'accions 25
Fragment codi 6: Comparació amb plantilles guardades 26
Fragment codi 7: Plantilla amb les distàncies més curtes 26
Taula 1: Usuaris experiment 27
Taula 2: Clau entrada dels usuaris 28
Taula 3: Claus d'accions usuaris 0 i 1 28
Taula 4: Claus d'accions usuaris 2 i 3 29
Taula 5: Claus d'accions usuaris 4 i 5 29
Taula 6: Comparació de dos thresholds a l'entrada 30
Taula 7: Threshold 7 a l'entrada 31
Imatge 13 : Comanda per executar l'eina 37
Imatge 14 : Menú principal 38
Imatge 15 : Menú principal inhabilitat per la fase d'enregistrament 38
Imatge 16: Menú d'enregistrament 39
Imatge 17: Menú d'enregistrament d'accions 39
Imatge 18: Finestra d'enregistrament de l'acció "Música ambiental" 40
Imatge 19 : L'usuari no està enregistrat 40
Imatge 20: Interior del sistema 41
Imatge 21: Acció realitzada amb èxit 41
Imatge 22: Acció no trobada 41

8. Annex

8.1 Manual d'usuari

En aquest apartat s'explicarà com utilitzar l'eina creada. Tant la implementació com la proba d'ús s'ha realitzat en el sistema operatiu Windows 10. Totes les gravacions duren cinc segons. Primer de tot ens hem d'assegurar de que tenim un micròfon connectat.

Com a requisits per fer ús del sistema tenim que s'ha implementat amb Python 3.6 de 32 bits. I, les llibreries utilitzades han estat: "tkinter", "os", "wave", "pyaudio", "python_speech_features", "scipy.io", "pydub" i "numpy".

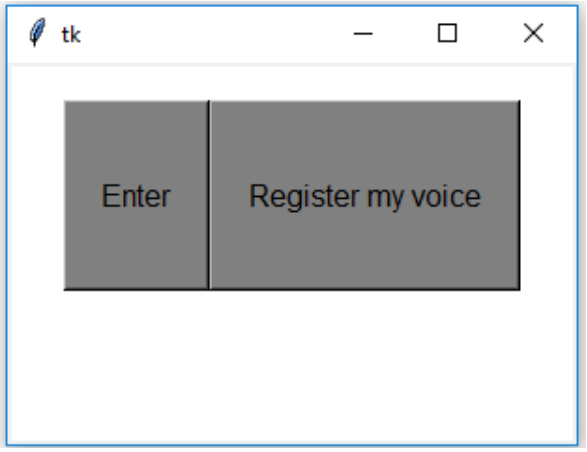
8.1.2 Obrir el programa

Per obrir el programa, s'ha de col·locar amb la línia de comandes al directori on està ubicada l'eina i s'executarà d'aquesta manera:

```
python main.py
```

Imatge 13 : Comanda per executar l'eina

Automàticament s'obrirà el menú principal que es mostra amb la imatge 14, present en la següent pàgina.

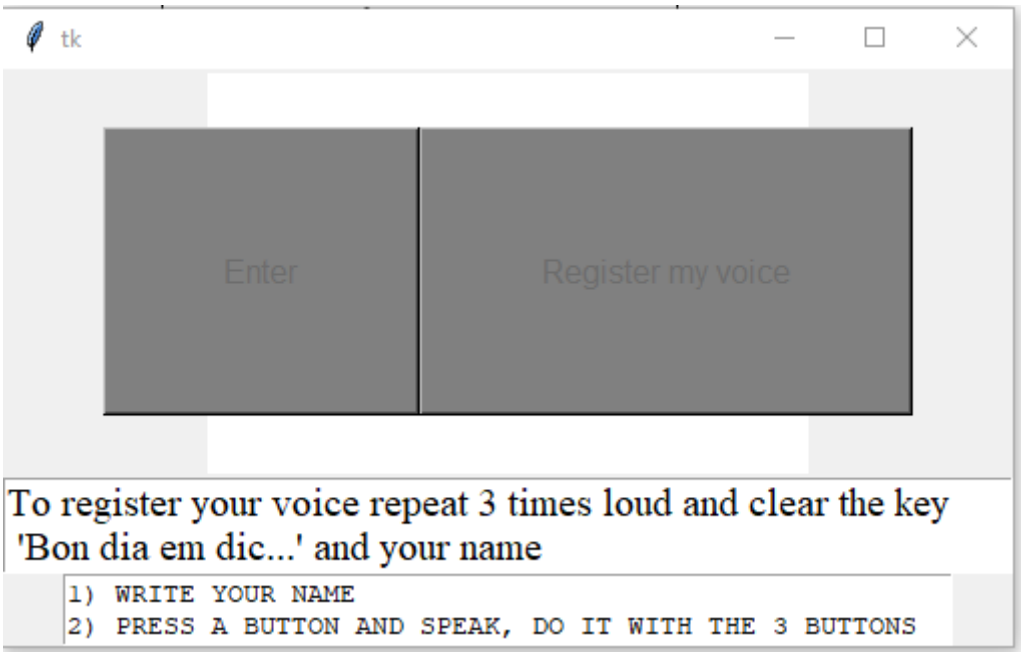


Imatge 14 : Menú principal

8.1.3 Enregistrar

Per començar s'haurà d'enregistrar la veu fent click al botó "Register my Voice".

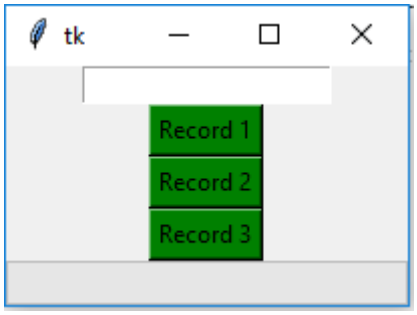
La següent imatge que ens trobarem és aquesta:



Imatge 15 : Menú principal inhabilitat per la fase d'enregistrament

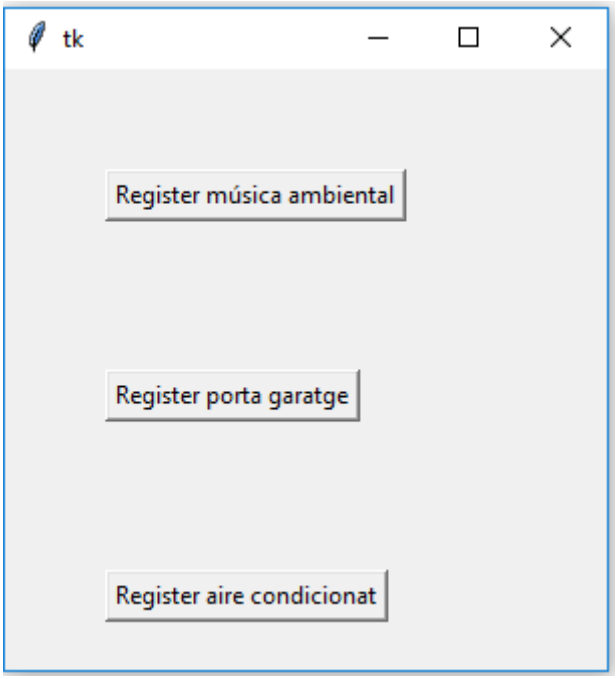
Com veiem el menú principal està inhabilitat i tenim unes instruccions per seguir amb l'enregistrament; Haurem de dir tres vegades alt i clar "Bon dia em dic.." seguit del

nostre nom. A més, haurem d'escriure el nostre nom. Això ho farem en aquesta finestra secundària que també s'ha obert:



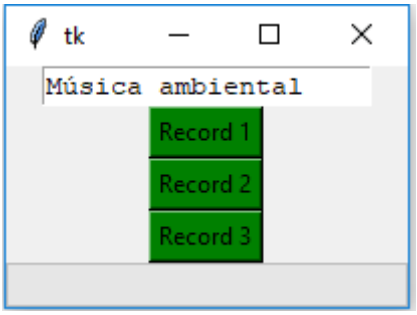
Imatge 16: Menú d'enregistrament

Al quadre blanc escriurem el nom i anirem clicant els tres botons. En cadascun d'aquests repetirem la frase clau descrita abans. Veurem com va creixent una barra de progrés i que els botons que anem clicant es tornen de color vermell per indicar que ja s'han utilitzat. Quan es finalitza aquest procés, aquesta finestra es tancarà automàticament i se n'obrirà una altra, la qual té com a finalitat enregistrar les claus per cada acció.



Imatge 17: Menú d'enregistrament d'accions

Per cada botó apareixerà una finestra com la utilitzada per guardar la clau d'entrada al sistema, però aquest cop no s'haurà d'escriure el nom, ja que ja està emmagatzemat.



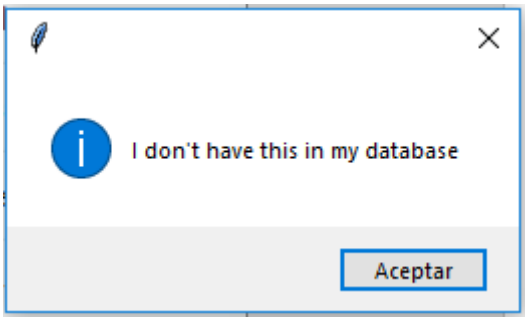
Imatge 18: Finestra d'enregistrament de l'acció "Música ambiental"

Com es pot veure on abans s'havia d'escriure el nom, ara hi ha l'acció que s'ha de repetir per enregistrar la clau i així en un futur utilitzar-la.

Quan hem guardat tres vegades l'acció i haguem repetit el mateix procés per les dos restants tornarem al menú principal i podrem intentar entrar com un usuari enregistrat.

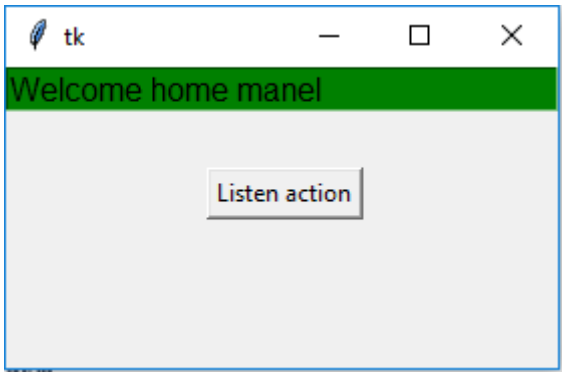
8.1.2.4 Accedir

Per això, haurem de clicar el botó "Enter". Un cop cliquem, l'aplicació escoltarà el que l'usuari diu i si la persona no està enregistrada es mostrarà aquest missatge per pantalla:



Imatge 19 : L'usuari no està enregistrat

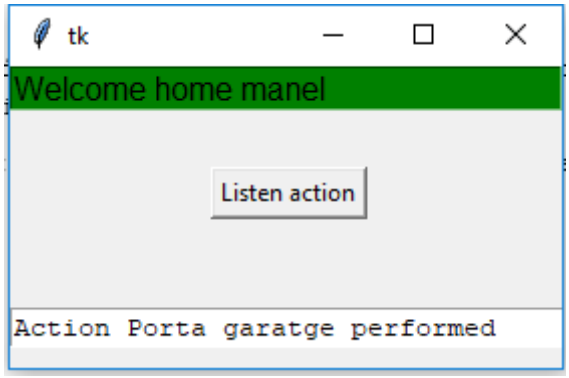
Si la comparació amb la base de dades ha sigut satisfactòria, podrem veure la següent imatge. En aquesta rebem una benvinguda i tenim la opció de clicar al botó de "Listen action" per indicar quina acció volem portar a terme.



Imatge 20: Interior del sistema

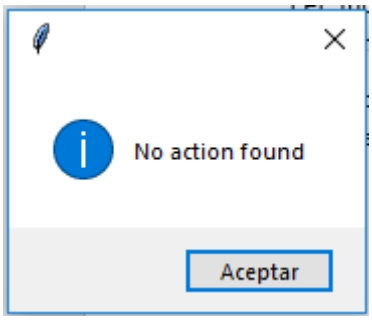
Per indicar quan acció volem, clicarem a "Listen action" i automàticament parlarem indicant una de les accions enregistrades abans.

Si l'acció que hem dit està guardada rebrem aquest missatge a la part inferior de la finestra.



Imatge 21: Acció realitzada amb èxit

En el cas que l'acció no s'hagi trobat rebrem aquesta alerta:



Imatge 22: Acció no trobada