
Autenticació mitjançant paràmetres biomètrics

Ricard Pastor Navarro

12-10-2017

*Treball final de grau en eng. informàtica
Tecnologies de la informació*



Facultat d'Informàtica de Barcelona
Universitat Politècnica de Catalunya
Director: Àlex Pajuelo González

1. Resum

Català

En les pàgines següents s'explicarà el procés seguit per la realització d'aquest treball de final de grau, en el que es pretén substituir, en un programa de gestió de contrasenyes, la contrasenya mestre per paràmetres biomètrics, concretament el reconeixement facial. Per a complir amb aquest objectiu s'utilitzaran alguns algorismes de visió per computador extrets de les llibreries de OpenCV. El sistema final agafarà imatges de l'usuari i farà la comprovació del usuari decidint si ha de donar o impedir l'accés.

Castellano

En las paginas siguientes se explicara el proceso seguido para la realización de este trabajo de final de grado, en el que se pretende sustituir, en un programa de gestión de contraseñas, la contraseña maestra por parametros biométricos, concretamente el reconocimiento facial. Para cumplir este objetivo se utilizaran algunos algoritmos de vision por computador extraidos de las librerías de OpenCV. El sistema final capturara imágenes del usuario y hará la comprobación del usuario decidiendo si tiene que dar o impedir el acceso.

English

The following pages will explain the process followed for the completion of this end-of-degree paper, in which it is intended to replace, in a password manager program, the master password by biometric parameters, namely facial recognition. To achieve this goal some computer vision algorithms extracted from the libraries of OpenCV will be used. The final system will capture images of the user and will check if it has to give or prevent access.

Contents

1	Resum	2
2	Plec de Condicions	5
2.1	Descripció i motivació	5
2.2	Estat actual	5
2.3	Descripció de l'arquitectura/Sistema	6
2.4	Descripció general de les tecnologies potencials	6
2.5	Descripció general de les eines de treball	6
2.6	Descripció dels riscos	6
2.7	Relació del projecte amb les competències tècniques	7
3	Abast del projecte	8
3.1	Context	8
3.1.1	Actors implicats	9
3.2	Estat de l'art	10
3.2.1	Historia	10
3.2.2	Estat actual	10
3.2.3	Altres tecnologies	11
3.3	Formulació del problema.	11
3.4	Abast	12
3.5	Metodologia i rigor	13
4	Planificació	14
4.1	Descripció de les tasques	14
4.1.1	Recursos	14
4.1.2	Planificació temporal	15
4.2	Valoració d'alternatives i pla d'actuació	23
5	Pressupost	24
5.1	Costos directes	24
5.1.1	Recursos de programari	24
5.1.2	Recursos humans	24
5.1.3	Recursos de maquinari	26

5.2	Costos indirectes	26
5.3	Contingència	27
5.4	Imprevistos	27
5.5	Costos totals	27
5.6	Control de gestió	27
6	Eines usades	28
6.1	Descripció de les eines	28
6.1.1	C++	28
6.1.2	OpenCV	28
6.1.3	KeepassX	29
7	Preparació del entorn	30
8	Implementació	32
8.1	Introducció	32
8.2	Codi reconeixement facial	33
8.2.1	Detecció	33
8.2.2	Entrenament o "Training"	34
8.2.3	Reconeixement	37
8.3	KeepassX	38
8.3.1	Modificacions a les interfícies gràfiques i al codi	38
8.3.2	Modificacions en els makefiles	40
8.3.3	Proves	41
9	Resultats	42
9.1	Resultats	42
9.1.1	Creació de la base de dades	42
9.1.2	Carregar la base de dades	43
10	Sostenibilitat	46
10.1	Dimensió Econòmica	46
10.2	Dimensió Social	47
10.3	Dimensió ambiental	47
10.4	Matriu de sostenibilitat	48
11	Conclusions i millores futures	49
11.1	Conclusions de seguretat	49
11.2	Conclusions personals	49
11.3	Possibles millores i treball futur	50
12	Bibliografia	51

2. Plec de Condicions

2.1 Descripció i motivació

Aquest projecte consistirà en crear un *software* que es pugui adaptar per substituir el mètode tradicional d'usuari i contrasenya per un sistema basat en els paràmetres biomètrics, com és el reconeixement facial és a dir que pugui reconèixer a l'usuari mitjançant una càmera. Per fer aquesta funció s'utilitzara una llibreria Open Source extreta d'internet, OpenCV.

La motivació d'aquest projecte és el meu interès en la seguretat informàtica i per poder reduir el creixent nombre de contrasenyes que s'han de memoritzar per poder navegar i utilitzar de forma segura les eines d'internet.

No hi ha un entorn definit, ja que és un TFG modalitat A i el que s'encarregarà de realitzar-lo serà una única persona amb l'ajuda d'un professor que servirà de director, que serà l'encarregat del seguiment del TFG.

Aquest projecte es relaciona amb l'especialitat amb el tema de seguretat informàtica, que és una part intrínseca de les tecnologies de la informació, també amb les eines que s'utilitzaran, ja sigui per agafar les dades, càmera web, emmagatzemar-les de forma segura, encriptació, i per transmetre-les, xarxes, VPNs.

2.2 Estat actual

No hi ha un estat actual, ja que el projecte es realitzarà des de zero. El que sí que hi ha són unes eines *Software*, aquestes eines no seran desenvolupades en el projecte, ja que desenvolupar aquestes eines faria que la duració i complicació del projecte fos més enllà de requerit per un TFG.

2.3 Descripció de l'arquitectura/Sistema

Aquest projecte és una solució *software* en el llenguatge de programació c++, que estarà dividida en diverses subtasques, per fer més fàcil la implementació. La primera subtasca és la creació d'un programa d'identificació facial, utilitzant OpenCV. Un cop feta aquesta tasca crearé un programa que els executi, compari els seus resultats i basat amb això identifiqui l'usuari. Per últim, com a cas d'us es buscarà un programa gestor de contrasenyes i s'adaptarà el programa perquè substitueixi el sistema d'usuari i contrasenya. Si és possible s'introduiran millores que hagin pogut anar sorgint en la realització del projecte.

2.4 Descripció general de les tecnologies potencials

Els recursos tecnològics seran la llibreria OpenCV, que té integrat una sèrie d'algoritmes de reconeixement d'objectes i facials, tals com l'algoritme Viola-Jones que permet detectar, retallar una cara d'una imatge i identificar-la. Aquestes llibreries utilitzen aprenentatge automàtic per poder fer les identificacions, pel que abans de poder utilitzar-les s'hauran d'entrenar, es a dir afegir informació de l'usuari perquè el programa pugui fer les comparacions. Per a l'emmagatzematge de les dades s'utilitzarà l'algoritme AES o TwoFish perquè no puguin ser utilitzades en cas que fossin robades.

2.5 Descripció general de les eines de treball

Les tecnologies *hardware* utilitzades seran únicament un portàtil Lenovo ideapad 100, amb càmera web integrada, amb un sistema operatiu Windows i un Debian. El llenguatge de programació principal serà C++, ja que les dues llibreries són compatibles amb aquest i com ja s'ha dit s'utilitzarà la llibreria externa OpenCV.

2.6 Descripció dels riscos

Un dels riscos més importants de les tecnologies biomètriques, es que si la informació es robada no pot ser canviada fàcilment, no com una contrasenya. Per aquest motiu totes les dades emmagatzemades seran encriptades i si s'han d'enviar dades sempre serà en una VPN. També s'impedirà que el codi sigui modificat i que l'entrada sigui exclusivament de la càmera web del dispositiu.

2.7 Relació del projecte amb les competències tècniques

Les competències tècniques que s'han escollit per aquest projecte CTI2.3 que es treballara en profunditat. A continuació s'explicara com esta relacionada amb el projecte.

La competència CTI2.3 s'ha escollit per que aquest projecte esta pensat per augmentar i simplificar la seguretat a Internet, ja que esta pensat per reduir la necessitat de contrasenyes a l'hora que utilitza paràmetres i característiques d'un individu.

3. Abast del projecte

3.1 Context

Aquest Treball de Final de Grau (TFG) és part de l'especialitat de Tecnologies de la Informació (TI) de la Facultat Informàtica de Barcelona (FIB) que pertany a la Universitat Politècnica de Barcelona (UPC).

Objectius

L'objectiu principal d'aquest projecte ha canviat de crear un adaptador a modificar un programa de gestió de contrasenyes per que es pugui realitzar la autenticació amb paràmetres biomètrics, en lloc de la contrasenya habitual, això servirà per simplificar el manteniment i memorització de contrasenyes a més de donar una capa de seguretat extra.

Els objectius secundaris son els de ampliar els meus coneixements en seguretat i aprenentatge automàtic i millorar la meva capacitat de treballar amb codis no fets originàriament per mi.

En la primera idea del projecte els paràmetres biomètrics que s'anaven a utilitzar eren el reconeixement facial i la identificació mitjançant la veu, però a causa del fet que no s'han pogut trobar les eines adients per fer la part de veu, s'ha optat només per la part d'imatge.

Conceptes

En els últims temps la preocupació per les nostres dades personals és cada cop més elevat i això ha fet que cada cop sigui més freqüent l'ús de contrasenyes en les nostres aplicacions per evitar que alguna persona aliena pugui accedir-hi, però l'ús de contrasenyes pot ocasionar problemes com que ens oblidem d'ella o que no sigui suficientment segura, entre d'altres. L'objectiu d'aquest treball és intentar reduir o eliminar alguns d'aquests problemes amb l'ajuda de diferents tecnologies

pel reconeixement biomètric.

La tecnologia biomètrica [4] és aquella que aplica tècniques estadístiques i matemàtiques sobre les característiques físiques o conducta de l'usuari per verificar la seva identitat. Algunes de les més característiques són l'empremta dactilar, retina, iris, patrons facials i veu. Els escollits per aquest projecte són els patrons facials.

El reconeixement facial consisteix en que el sistema, a partir d'una imatge o un gravació de vídeo, reconegui el usuari, utilitzant algorismes de extracció d'informació, coneguts com a visio per computador.

Originalment la idea del projecte era aconseguir utilitzar a més del reconeixement facial la identificació del parlant mitjançant la veu però finalment aquesta idea no ha pogut ser portada a terme, ja que no es va trobar cap utilitat amb les funcionalitats desitjades per a la identificació mitjançant la veu i s'ha optat per fer el reconeixement facial i a causa d'això s'ha decidit fer la integració directament en el programa desitjat, sense necessitat d'adaptadors.

3.1.1 Actors implicats

Aquest projecte no té un objectiu d'actors implicat definit, ja que el podran usar tan usuaris independents, que vulguin tenir cura de les seves dades personals o empreses que vulguin afegir una capa extra de seguretat a la seva xarxa.

- Usuaris independents: Com s'ha exposat anteriorment cada cop més els usuaris són més conscients dels perills de què les seves dades siguin interceptades per persones amb intencions malignes amb aquest projecte intentaré afegir una capa de seguretat addicional més difícil de traspasar que una contrasenya convencional.
- Empreses: Les empreses han de ser encara més prudents amb les seves dades i amb qui pot accedir als seus arxius i xarxes, amb el resultat d'aquest projecte a més de la capa extra de seguretat també pot millorar la gestió d'usuaris.
- L'autor i el seu professor supervisor: L'autor serà l'encarregat d'aconseguir ajuntar les tecnologies de reconeixement facial i la identificació per veu i intentar aconseguir que tinguin una gran taxa de fiabilitat a més de crear l'adaptador perquè es puguin utilitzar com a contrasenya. Mentre que professor supervisor Álex Pajuelo González s'encarregarà de guiar i ajudar-lo en el procés de disseny i programació.

3.2 Estat de l'art

3.2.1 Historia

Abans de l'època de la informàtica la identificació no era un problema important en l'àmbit informal, podies reconèixer amb qui t'estaves comunicant visualment, o mitjançant mètodes basats en la confiança, no és així amb els sectors militars o privats que necessitaven altres mètodes més fiables, com codis o mitjançant la criptografia.

3.2.2 Estat actual

En el temps de la informàtica, el que estava reservat per àmbits militars o confidencials, s'estén a la gran majoria de la població, ja que la humanitat passa a estar connectada i s'han de buscar formes de decidir que en efecte ens estem comunicant amb el receptor correcte, també amb aquesta connexió permetem que les nostres dades, siguin personals o públiques, siguin accessibles per una quantitat molt més elevada d'individus, per tant s'han de posar eines per evitar que siguin utilitzades per individus amb intencions malicioses, i una forma senzilla són les contrasenyes[6].

La contrasenya és un conjunt de caràcters que només sap l'usuari, que permet que el sistema al qual volem accedir, per exemple una pàgina web, pugui identificar que l'usuari és qui diu ser.

Però les contrasenyes al final només són un conjunt de caràcters alfanumèrics i són tan segures com l'usuari que les utilitza vol que siguin, i si es vol una contrasenya segura, s'han de complir una sèrie de condicions com que tinguin una llargada determinada, intercalar majúscules i minúscules, i moltes més, el que fa és augmentar la complicació i molts cops, per evitar aquesta complicació extra, es reutilitzen o s'utilitzen noms o fets fàcils de recordar, però també fàcils d'esbrinar, com el nom de l'usuari o la data de naixement[3].

Al principi aquesta era una bona condició, ja que no hi havia tants recursos que necessitaven la verificació de l'usuari, però amb el temps això ha anat canviant, cada cop hi ha més informació privada o sensible a la xarxa i cada cop hi ha més recursos que passen a estar a Internet, com ara la banca electrònica i per tant si volem mantenir aquestes dades en privat, s'han de memoritzar cada cop més contrasenyes.

Per evitar aquestes situacions s'han creat eines, com els gestors de contrasenyes, que ens permeten reduir la gran quantitat de contrasenyes a només una la con-

trasenya mestra, però també augmenta el risc, ja que si algú la descobreix, té accés a totes les nostres contrasenyes guardades.

Aquí és on entren les tecnologies biomètriques, que permetran al sistema reconèixer a l'usuari sense necessitat de contrasenyes usant les capacitats de la visió per computador.

Aquestes ja són usades en diferents àmbits, com per exemple el DNI, per identificar amb poc error a l'usuari al qual pertanyen.

Les tecnologies biomètriques més utilitzades són el reconeixement facial, l'empremta dactilar i el reconeixement d'iris.

El problema principal de les tecnologies biomètriques és que si algun individu les intercepta, no es poden canviar. Per aquest motiu s'hauran de tenir especial cura a no enviar cap dada no encriptada i que el sistema sigui tancat i no accepti modificacions de fonts externes.

3.2.3 Altres tecnologies

Com s'ha comentat anteriorment ja hi ha tecnologies utilitzant la biometria per identificar a l'usuari. Una de les més importants és les empremtes dactilars, és a dir utilitzar les formes característiques de la impressió visible o modelada que produeix el contacte de les crestes papil·lars d'un dit de la mà per identificar al seu propietari, això és degut al fet que cada individu té unes diferents. Molts telèfons mòbils actuals porten aquests tipus de sistema, encara que actualment són encara molt poc precisos, també són utilitzats en molts altres llocs. Aquest mètode també és molt utilitzat a les investigacions policials.

Una altra tecnologia biomètrica bastant aplicada és la identificació facial, que ja ha sigut explicada abans, com a exemple es pot posar és l'etiquetatge de Facebook o el nou sistema del bac BBVA en el que pots obrir un compte amb una foto i una videotrucada.

3.3 Formulació del problema.

L'objectiu principal d'aquest projecte ha variat durant la creació d'aquest. Originalment era crear un adaptador per software, que permeti substituir l'ús de contrasenya per l'ús de la pròpia imatge i la veu agafades directament de la càmera del dispositiu en el qual s'utilitzi, adjuntant tecnologies d'identificació facial i d'identificador de l'interlocutor. Però pels motius explicats anteriorment s'ha decidit optar per modificar un programa de gestió de contrasenyes per afegir-li

opcions de reconeixement facial.

El programa gestor de contrasenyes que s'ha elegit es anomenat KeepasX i es bastant reconegut en àmbits de software lliure de seguretat informàtica.

Per afegir més seguretat i per evitar el robatori d'identitat el software serà l'encarregat de fer el reconeixement facial sense transmetre mai les dades.

Per evitar que el programa pugui ser enganyat quan s'iniciï agafarà les dades directament de la càmera sense permetre que s'introdueixin altres fonts per fer reconeixement, a més totes les comunicacions seran mitjançant una xarxa virtual privada (VPN)[4], que és una tecnologia de xarxa que permet l'extensió segura de la xarxa local (LAN) sobre una xarxa pública o no controlada com Internet.

A causa del canvi en les funcionalitats del software l'objectiu final també ha variat, ja que s'ha optat per adaptar directament un programa de gestió de contrasenyes amb les funcionalitats de reconeixement facial, que seran agafades directament de la càmera del dispositiu.

Finalment el projecte ha sigut dividit en 3 apartats:

1. Instal·lar les llibreries de visió per computador i crear un programa per fer la identificació facial i entrenar-lo perquè identifiqui a l'usuari.
2. Instal·lar i provar el funcionament del programa gestor de contrasenyes KeepassX.
3. Modificar el programa per poder fer que a part de les opcions que tenia anteriorment també permeti la utilització del reconeixement facial com a contrasenya mestra.

3.4 Abast

L'abast d'aquest projecte és aconseguir una opció segura per poder substituir el mètode clàssic d'usuari i contrasenya sense que tingui un gran impacte a l'usuari, encara que com gairebé qualsevol aplicació orientada a la seguretat pot tenir un petit impacte, ja que s'ha de comptar amb el temps tardat comunicar-se amb el servei i el temps que pot tardar aquest a fer el reconeixement.

Els possibles problemes que es pot trobar són:

1. Poden ser relacionats amb les llibreries externes o amb la plataforma de veu, que podran ser substituïts si l'autor no és capaç de fer la integració dels dos serveis. Aquest ha sigut el problema principal amb el que l'autor s'ha trobat que han fet canviar el plantejament del treball.

2. També es pot tenir problemes amb l'entrenament del sistema, ja que perquè sigui precís requereix diferents tipus de proves i diferents usuaris, o
3. Amb la fiabilitat de les eines, ja que si identifiquen malament a l'usuari, poden causar bretxes de seguretat.

3.5 Metodologia i rigor

La metodologia és que l'autor treballarà en el projecte pel seu compte amb l'ajuda del director del projecte. També tindran lloc reunions periòdiques per comentar els progressos o els problemes que puguin sorgir.

En el transcurs d'aquest projecte s'intentarà utilitzant les eines de codi lliure, OpenCV, que és una sèrie de llibreries per la visió per computador.

A més d'utilitzar eines de seguretat informàtica, com l'encriptació per les dades sensibles i les xarxes virtuals privades (VPN) per les comunicacions segures.

En aquest projecte s'ha decidit utilitzar aquestes dues tecnologies biomètriques, ja que són les que més població encabeix, ja que actualment, gairebé qualsevol ordinador portàtil incorpora una càmera de vídeo, a més que les webcams són de fàcil obtenció i barates.

Tot el codi es farà amb C++, ja que és un llenguatge molt conegut, a més que és compatible amb les dues eines. També s'ha utilitzat QT per a la interfície gràfica ja que es la que el programa utilitzava.

Les eines de seguiment seran les reunions amb el director en les quals l'autor, si és possible, farà demostracions de com funciona el programa. A part s'utilitzaran eines com diagrames de Gantt per a la planificació.

La metodologia que es seguirà per aquest treball per la part de GEP es tipus Cascada i per el projecte en si de tipus Àgil

4. Planificació

4.1 Descripció de les tasques

4.1.1 Recursos

En aquesta secció s'analitzen els recursos necessaris per a la realització del projecte. A continuació es detallen els recursos humans, de maquinari i de programari utilitzats.

Recursos humans

El projecte el realitzarà un sol individu, que haurà d'assumir tots rols del projecte des des de cap del projecte fins a *tester*. També es comptarà amb l'ajuda del director del projecte, que assumirà el paper de consultor/supervisor.

Recursos de maquinari

Per la realització del projecte no serà necessari adquirir cap mena de maquinari específic, ja que s'utilitzarà un portàtil amb càmera integrada.

Recursos de programari

Durant la realització del projecte i el curs de GEP, s'utilitzaran diverses eines, llistades a continuació:

Nom	Tipus	Ús
Distribució basada en Debian	Eina de desenvolupament	Execució del programari
Windows	Eina de desenvolupament	Execució del programari
C++	Eina de desenvolupament	Llenguatge de programació
OpenCV	Eina de desenvolupament	Algorismes de VC
QT	Eina de desenvolupament	Framework per interfícies gràfiques
KeepassX	Eina de desenvolupament	Gestor de contrasenyes
Vim	Eina de desenvolupament	Programació del codi
QTCreator	Eina de desenvolupament	IDE per l'edició de QT.
L ^A T _E X	Eina de desenvolupament	Redacció dels documents
Google docs	Eina de desenvolupament	Redacció i emmagatzematge dels documents
Adobe	Eina de desenvolupament	Visualització de pdf
Gantter	Eina de gestió	Creació diagrames de Gantt
Github	Desenvolupament i gestió	Control de versions

Table 4.1: Recursos de programari

4.1.2 Planificació temporal

El treball té una duració aproximada de 4 mesos, des de mitjans de Febrer fins a finals de Juny. La càrrega total serà d'unes 420 hores, corresponents a 18 crèdits ECTS. La dedicació setmanal estimada serà d'unes 25 hores.

Es dividirà el projecte en tres grans blocs, descrits a continuació:

Bloc	Descripció	Metodologia	Hores
Bloc 1	Curs de GEP	Cascada	75h
Bloc 2	Desenvolupament del projecte	Àgil	334h
Bloc 3	Preparació de la defensa	-	15h

Table 4.2: Blocs del projecte

Bloc 1: Curs de GEP

Aquest bloc correspon a la realització del curs de GEP, amb inici el dia 20/02/2017 i finalització el 28/03/2017. No té dependències

Durant el curs s'entregaran 6 lliurables, detallats a continuació:

Descripció	Inici	Finalització	Durada	Hores
Definició de l'abast i contextualització	20/02/2017	03/03/2017	11 dies	15h
Planificació temporal	03/02/2017	06/03/2017	3 dies	6h
Gestió econòmica i sostenibilitat	07/03/2017	13/03/2017	7 dies	13h
Presentació preliminar	14/03/2017	21/03/2017	7 dies	13h
Plec de condicions	22/03/2017	28/03/2017	7 dies	13h
Document final i presentació oral	22/03/2017	28/03/2017	7 dies	15h

Table 4.3: Lliurables de GEP

Bloc 2: Projecte

El bloc principal consisteix en el desenvolupament del projecte: buscar informació, implementar el codi de reconeixement facial, entrenar els reconeixements, redactar aquesta memòria, etc.

Aquest bloc no té dependències i es dividirà en 6 subtasques.

Tasca	Inici	Finalització	Durada	Hores
Reconeixement facial	29/03/2017	10/05/2017	42 dies	84h
Rebuig de la idea original	10/05/2017	15/05/2017	5 dies	10h
Cerca de programa i familiarització	15/05/2017	30/05/2017	15 dies	30h
Adaptació	30/05/2017	28/08/2017	60 dies	120h
Millores	29/08/2017	22/09/2017	24 dies	24h
Memòria	29/05/2017	22/09/2017	116 dies	58h

Table 4.4: Tasques desenvolupament

A causa de la complicació del projecte l'autor ha decidit dividir-lo en diferents subprojectes, que seran descrits a continuació.

Reconeixement facial

Instal·lació OpenCV i familiarització

El primer que es farà és instal·lar les llibreries de visió per computador, OpenCV, necessàries per al projecte a més de proves per familiaritzar-se amb l'entorn.

Crear programa reconeixement facial

Aquest apartat consisteix a crear un programa que agafant una imatge o vídeo permeti localitzar les cares i se'l pugui entrenar per reconèixer a usuaris. L'apartat de proves serà per provar la que el programa funciona com està planejat i es farà en cada un dels subprojectes.

Entrenament i proves

En aquest pas s'entrenarà al programa per reconèixer a l'usuari i es faran les proves necessàries.

Rebuig de la idea original

En aquest apartat es comprova que no es pot fer el que es plantejava originalment, al no ser capaç de trobar cap llibreria que detecti i identifiqui a l'usuari en temps real, tot el que es va trobar necessita un preprocessat que no serveix per a les necessitats d'aquest projecte, per tant és necessari modificar el plantejament d'aquest.

Els dos subapartats anteriors no tenen una relació jeràrquica, però ja que és un projecte individual, tenen precedència temporal.

Cerca de programa i familiarització

Buscar programa gestor de contrasenyes

El programa triat es el KeePassX un programa gestor de contrasenyes bastant usat, Open Source i escrit amb el llenguatge C++.

Familiarització

Aquest període de temps es necessari per aprendre com funciona el programa i com poder modificar-lo per que integri el reconeixement facial com a contrasenya.

Adaptació

En aquest apartat es faran els canvis necessaris per poder integrar la funció de reconeixement facial en el programa de gestió de contrasenyes prèviament triat. També es realitzaran les proves necessàries per garantir el correcte funcionament del programa.

Millores (opcional)

Un cop enllestit, en cas de disposar de més temps, es podran fer ampliacions i millores que hagin pogut anar sorgint en el transcurs del projecte.

Redacció de la memòria

La memòria s'anirà redactant a l'hora que es faci el projecte per aquest motiu no hi ha cap dependència.

Bloc 3: Preparació de la defensa

En el bloc final es revisarà i estudiarà la memòria per preparar la presentació final del projecte. Està previst dedicar unes 18 hores, que començarà el dia 16 i acabarà el 22 de Octubre.

La defensa del projecte es durà a terme entre els dies 23 i 27 de Octubre.

Diagrames

	Nombre	Inicio	Fin
1	☐ GEP	20/02/2017	28/03/2017
2	Lliurable 1	20/02/2017	03/03/2017
3	Lliurable 2	03/03/2017	06/03/2017
4	Lliurable 3	07/03/2017	13/03/2017
5	Lliurable 4	14/03/2017	21/03/2017
6	Lliurable 5	22/03/2017	28/03/2017
7	Lliurable 6	22/03/2017	28/03/2017
8	☐ Projecte	20/02/2017	09/06/2017
9	☐ Reconeixement facial	20/02/2017	20/03/2017
10	Instal·lació OpenCV i familiarització	20/02/2017	20/02/2017
11	Crear programa reconeixement facial	21/02/2017	13/03/2017
12	Entrenament i proves	14/03/2017	20/03/2017
13	☐ Identificació de l'interlocutor	21/03/2017	13/04/2017
14	Instal·lar Alizé i familiarització	21/03/2017	21/03/2017
15	Creació del programa d'identificació de l'interlocutor	22/03/2017	06/04/2017
16	Entrenament i proves	07/04/2017	13/04/2017
17	☐ Integració	14/04/2017	09/05/2017
18	Creació del programa d'integració	14/04/2017	01/05/2017
19	Proves	02/05/2017	09/05/2017
20	☐ Adaptació	10/05/2017	01/06/2017
21	Adaptació del programa	10/05/2017	24/05/2017
22	Proves	25/05/2017	01/06/2017
23	Millores (opcional)	02/06/2017	09/06/2017
24	Memòria	02/03/2017	09/06/2017
25	Preparació de la defensa	12/06/2017	16/06/2017

Figure 4.1: Descripció de tasques Gantt final Figura 4.2

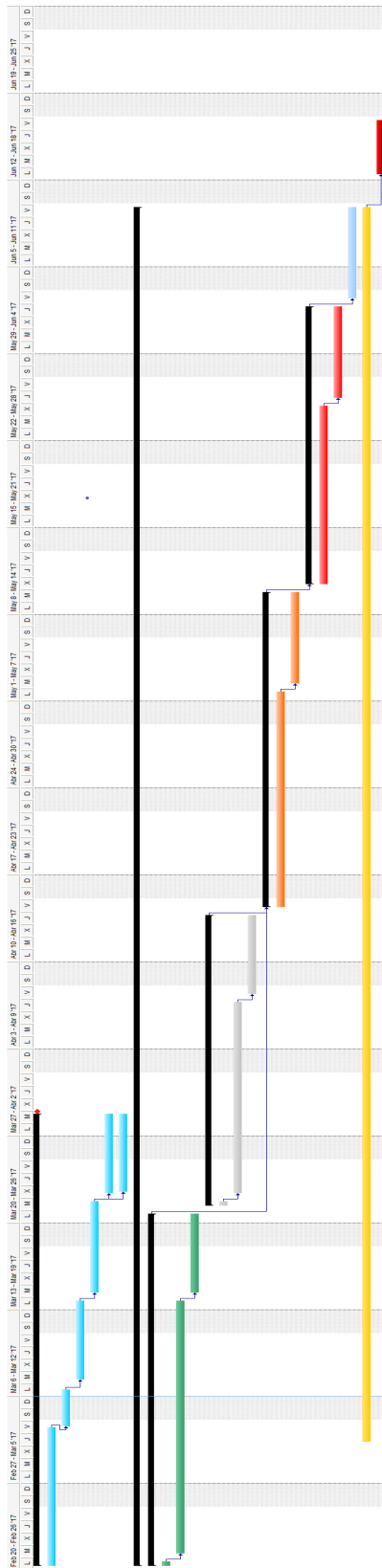


Figure 4.2: Gantt original del projecte

	Nombre	Duració	Inici	Fin	Predecessoras	Recursos
	GEP	9d?	20/02/2017	28/03/2017		
	Lliurable 1	3.17d?	20/02/2017	03/03/2017		
	Lliurable 2	0.33d?	03/03/2017	06/03/2017	2	
	Lliurable 3	1.67d?	07/03/2017	13/03/2017	3	
	Lliurable 4	2d?	14/03/2017	21/03/2017	4	
	Lliurable 5	1.67d	22/03/2017	28/03/2017	5	
	Lliurable 6	1.54d?	22/03/2017	28/03/2017	5	
	Projecte	47.67d?	29/03/2017	13/10/2017		
	Reconeixement facial	10.17d?	29/03/2017	10/05/2017		
	Instal·lació OpenCV i familiarització	0.33d?	29/03/2017	29/03/2017		
	Crear programa reconeixement facial	8.92d?	30/03/2017	05/05/2017	10	
	Entrenament i proves	0.92d?	05/05/2017	10/05/2017	11	
	Rebuig de la idea original	1.17d?	10/05/2017	15/05/2017		
	Cerca de programa i familiarització	3.79d?	15/05/2017	30/05/2017	9,13	
	Buscar programa gestor de contrasenyes	0.79d?	15/05/2017	17/05/2017		
	Familiarització	3d?	17/05/2017	30/05/2017	15	
	Adaptació	21.33d?	31/05/2017	28/08/2017	14	
	Adaptació del programa	12.67d?	31/05/2017	21/07/2017		
	Proves	2d?	21/08/2017	28/08/2017	18	
	Millores (opcional)	6.33d?	29/08/2017	22/09/2017	17	
	Memòria	11.67d?	28/08/2017	13/10/2017		
	Preparació de la defensa	1.67d?	16/10/2017	23/10/2017	21	

Figure 4.3: Descripció de tasques Gantt final Figura 4.4

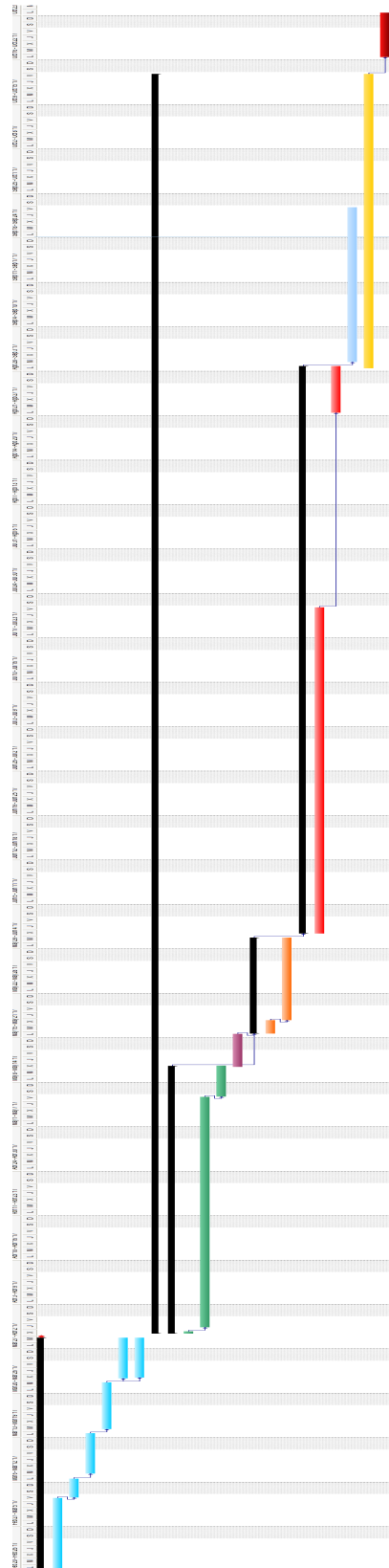


Figure 4.4: Gantt final del projecte

4.2 Valoració d'alternatives i pla d'actuació

Mala planificació [Impacte: baix]

Es faran reunions amb el professor per evitar-ho. També, si fos necessari, es podrien utilitzar el temps planejat per les millores per solucionar aquest problema. En cas que la planificació falli molt, es pot ampliar el temps diari dedicat al projecte.

Fallada de hardware [Impacte: mitja]

En cas de fallades en portàtil, s'hauria d'adquirir una webcam el que faria augmentaria el temps necessari. No hi hauria una pèrdua de dades important, ja que es treballa amb redundància en línia.

Incompatibilitat de les eines triades [Impacte: mitja]

En cas d'incompatibilitat de les dues eines triades s'haurà d'escollir un altre per fer la substitució

5. Pressupost

5.1 Costos directes

5.1.1 Recursos de programari

Tot el *software* utilitzat en aquest projecte és gratuït i de codi obert, menys el sistema operatiu Windows que ja venia amb la màquina. Per tant, el programari no suposarà cap despesa.

Podeu trobar el llistat del programari utilitzat a la taula Recursos de programari de l'apartat anterior.

5.1.2 Recursos humans

Com ja s'ha explicat anteriorment, aquest projecte serà realitzat per una sola persona, que s'encarregarà de tots els rols.

Tenint en compte les tasques descrites a la secció de planificació, les hores de treball queden repartides de la següent manera:

Tasca	Cap	Analista	Programador	Cost
Curs de GEP	75h			1.500 €
Lliurament 1	15h			300 €
Lliurament 2	6h			120 €
Lliurament 3	13h			260 €
Lliurament 4	13h			260 €
Lliurament 5	13h			260 €
Lliurament 6	15h			300 €
Projecte		55h	213h	2.955 €
Reconeixement facial		10h	74h	890 €
Rebuig idea original		10h		150 €
Cerca programa i familiarització		15h	15h	375 €
Adaptació		20h	100h	1.300 €
Millores			24h	240 €
Redacció de la memòria	58h			1.160 €
Preparació defensa	15h			300 €
Total	148h	55h	213h	5.915 €

Table 5.1: Recursos humans (hores i costos relacionats)

S'ha separat la redacció de la memòria del projecte en si, ja que les hores assignades a la memòria seran assignades al cap de projecte.

Suposem uns costos de 20 €/h pel cap de projecte, 15 €/h per l'analista i 10 €/h pel programador/*tester*.

Rol	Hores	Cost/hora	Cost total
Cap de projecte	148h	20€/h	2.960 €
Analista	55h	15€/h	825 €
Programador	213h	10€/h	2.130 €
Total			5.915 €

Table 5.2: Recursos humans (costos)

5.1.3 Recursos de maquinari

El *hardware* ja va ser descrit en l'apartat de recursos i consisteix en un ordinador portàtil amb càmera.

Producte	Preu	Ús	Vida útil	Amortització
Ordinador personal	300€	4 mesos	5 anys	23,4 €
Total				23,4 €

Table 5.3: Costos de maquinari

5.2 Costos indirectes

En costos indirectes s'inclouran els més importants: la connexió a Internet i el consum elèctric. Aquest seran fàcils d'amortitzar, ja que en tractar-se d'un domicili particular no només s'utilitzen en el desenvolupament del projecte.

Tipus	Temps	Cost	Cost total
Connexió a Internet	4 mesos	35 €/m	140 €
Electricitat	4 mesos	75 €/m	300 €
Total			440€

Table 5.4: Costos indirectes

5.3 Contingència

Com a mesura de contingència, s'estableix un marge del 10%.

5.4 Imprevistos

Es podria donar el cas que el projecte ocupi més temps de l'esperat, pel que es considerarà d'hores de treball, que recaurien en el programador. Per aquest motiu s'augmenta el pressupost en 800 € que es podrien no utilitzar.

També per possibles si l'ordinador principal falla es reserven 20 € per comprar una càmera web.

5.5 Costos totals

Tipus	Cost estimat
Costos Directes	5.938.5 €
Recursos de programari	0 €
Recursos humans	5.915 €
Recursos de maquinari	23,4 €
Costos indirectes	440€
Imprevistos	820€
Contingència (10%)	736,55 €
Total	7935,05 €

Table 5.5: Costos totals

5.6 Control de gestió

A l'acabar cada tasca és farà una comprovació de si és compleixen les estimacions temporals i de pressupost, en cas que no es compleixin es prendran mesures per arreglar aquests desviaments.

Es calcularà la desviació en mà d'obra, programari, maquinari i altres costos (cost estimat - cost real) * hores reals.

6. Eines usades

6.1 Descripció de les eines

En aquesta secció es farà una breu descripció de les eines que s'han utilitzat per a la realització del projecte.

6.1.1 C++

Com s'ha comentat anteriorment en aquest TFG estarà escrit en el llenguatge de programació C++, ja que és àmpliament conegut i compatible amb les eines que és volent utilitzar. C++ va ser creat l'any 1983 i està estandarditzat per l'ISO l'any 1998 i la seva última estandardització data del 2014, sent actualment la seva última versió, les seves característiques principals són que està orientat a objectes, que és un llenguatge imperatiu i genèric. Per la part visual el que s'usarà és QT que és un framework multiplataforma que s'utilitza per crear interfícies gràfiques, també de codi lliure. QT va ser creat l'any 1991, per Trolltech, més endavant va ser comprat per Nokia, fent-se de codi totalment obert l'any 2012.

6.1.2 OpenCV

Com ja s'ha dit anteriorment OpenCV és una llibreria de més de 2500 algoritmes optimitzats d'aprenentatge automàtic o "Machine learning" i de Visió per computador en temps real. Aquesta llibreria té una llicència de codi lliure BSD, tant per finalitats educatives com comercials i una comunitat de més de 47 mil d'usuaris. Compatible amb C++, que és el que ens interessa en aquest projecte, Python i Java. Les funcionalitats que s'han utilitzat d'aquestes llibreries són principalment la de reconeixement facial que consta de:

1. Detecció de la cara
2. Extracció de les característiques facials
3. Fer la comparació amb les característiques de l'usuari que hauran sigut prèviament extretes i guardades

També s'usaran altres característiques d'aquesta llibreria com les d'interactuar amb la càmera per agafar la imatge OpenCV va ser llençat l'any 1999, sent inicialment un projecte de recerca d'Intel i la seva alpha va ser publicada a la IEE l'any 2000, millorant des de llavors, i l'any 2012 una organització sense ànim de lucre va encarregar-se del projecte.

6.1.3 KeepassX

[1] Com s'ha comentat anteriorment el programa de gestió de contrasenyes escollit ha sigut el KeepassX, aquest programa té nombrosos avantatges, la primera sent que es tracta d'un programa de codi lliure "GNU General Public License", el que ens permet agafar el codi font per tal de poder modificar-lo com ens sigui convenient, un altre dels avantatges que té, és que està escrit en C++, el que permet la fàcil integració amb les noves característiques i llibreries que s'han utilitzat en el projecte, i una tercera és que és un programa bastant conegut i usat, el que indica que és una bona opció en el tema de seguretat. KeepassX va començar com a Keepass/L un port per a Linux d'una aplicació per Windows anomenada "Keepass Password Safe" però més endavant, en convertir-se en multiplataforma va canviar el nom. Respecte a l'encriptació utilitza AES (Rijndael) o Twofish de 256 bits

7. Preparació del entorn

OpenCV

Com que el projecte està desenvolupat utilitzant la biblioteca OpenCV, serà necessari instal·lar-la. Per a la correcta instal·lació d'aquestes llibreries son necessaris unes eines i llibreries, que seran les que s'instal·lin primer.

```
$ sudo apt-get install cmake
$ sudo apt-get install libjpeg-dev libtiff5-dev
libjasper-dev libpng12-dev
$ sudo apt-get install libatlas-base-dev gfortran
```

La versió a instal·lar és la 3.2, que ens podem baixar des del repositori oficial d'OpenCV al GitHub.

```
$ wget -O opencv.zip https://github.com/Itseez/
opencv/archive/3.2.0.zip
$ unzip opencv.zip
```

A part també hem de descarregar uns altres algorismes que no forment part de OpenCV, ja que no tenen una API estable o no estan encara totalment testejats.

```
$ wget -O opencv_contrib.zip https://github.com/
Itseez/opencv_contrib/archive/3.2.0.zip
$ unzip opencv_contrib.zip
```

Un cop obtinguts els arxius necessaris, ja podem preparar, compilar i instal·lar OpenCV al nostre sistema.

```

$ cd ~/opencv-3.2.0/
$ mkdir build
$ cd build
$ cmake -D CMAKE_BUILD_TYPE=RELEASE \
-D CMAKE_INSTALL_PREFIX=/usr/local \
-D INSTALL_C_EXAMPLES=ON \
-D INSTALL_PYTHON_EXAMPLES=ON \
-D OPENCV_EXTRA_MODULES_PATH=
~/opencv_contrib-3.2.0/modules \
-D BUILD_EXAMPLES=ON ..
$ make -j4
$ sudo make install
$ sudo ldconfig

```

KeepassX

També hems hem de baixar el programa on volem aplicar-li les modificacions per aquest motiu anirem a la pagina web del programa <https://www.keepassx.org/downloads> i descarregar la versió per Linux, la versió que utilitzarem es la ultima, la 2.0.3, per instal·lar-lo el que hem de fer es:

```

$ tar xzvf KeePassX-2.0.3.tar.gz
$ cd keepassx-2.0.3
$ cmake .
$ make
$ sudo make install

```

8. Implementació

8.1 Introducció

En aquest capítol s'explicarà la implementació bàsica de el codi que compleixi els objectis proposats per aquest TFG, aquest serà dividit en dues parts, en la primera part s'explica com funciona el codi que permet el reconeixement facial i en la segona part s'expliquen els canvis necessaris al programa KeepasX per fer la integració de l'apartat anterior.

L'apartat del codi està dividit en tres parts, que són les funcions principals del codi:

- El primer apartat serà la detecció de la cara en una imatge. Aquest serà usat per les dues parts del codi següents.
- El segon apartat, és l'entrenament o "training" en el que gràcies a capturar la cara del usuari repetides vegades permet que el programa pugui fer el reconeixement.
- El tercer i últim apartat és el propi reconeixement de l'usuari, utilitzant l'entrenament de l'apartat anterior.

L'apartat de l'adaptació de KeepasX serà dividit en dos apartats:

- El primer és la modificació del programa per permetre fer la adaptació, aquest apartat inclou la modificació del codi i de les interfícies visuals
- El segon, consisteix en la modificació dels CMakeLists per permetre la correcta compilació del programa.

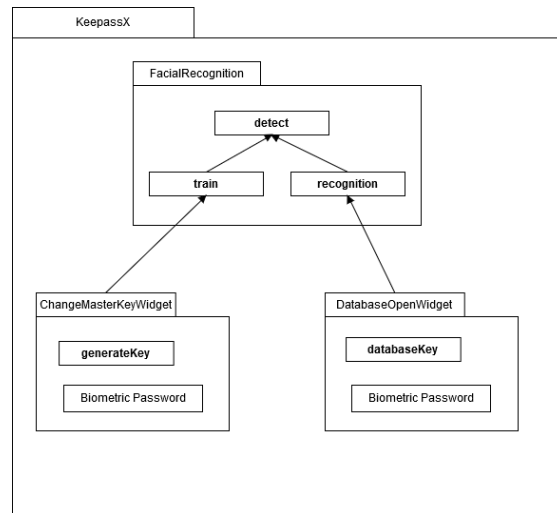


Figure 8.1: Diagrama amb relacions entre les classes modificades

El que es definirà aquí serà el .c++ però també es crearà un arxiu amb els headers (.h).

8.2 Codi reconeixement facial

Com s'ha comentat en aquesta apartat es comentara amb detall com funciona el codi de reconeixement facial que estarà dividit en les diferents funcions que s'han implementat en el codi, sense comptar las funcions creadores i destructores. Tot el codi es podrà trobar en la secció dels annexos

8.2.1 Detecció

```

std::vector<cv::Rect> FacialRecognition::detect
(cv::Mat frame, cv::CascadeClassifier face_cascade)
    
```

Aquesta es una funció auxiliar privada que utilitzaran les altres dues funcions per detectar una cara, com es pot veure com a paràmetres d'entrada té una matriu de la llibreria OpenCV que conte una imatge i un classificador per fer la detecció de la cara en la imatge, aquest classificador es en cascada, el que significa que la imatge es passara per diferents classificador, sent la entrada el resultat del classificador anterior, permeten fer una detecció d'objectes més precisa.

Abans de fer la detecció s'ha de preprocessar la imatge, es ha dir, fer tractar la imatge amb diferents algoritmes, per fer que la maquina pugi entendre la imatge i, en aquest cas, detectar la car aquestes accions son:

```
cvtColor ( frame, frame_gray, CV_BGR2GRAY );
equalizeHist (frame_gray, frame_gray );
```

El que fa el primer es convertir la imatge a blanc i negre, el que permet passar d'una matriu multi-dimensional (RGB) a una unidimensional amb valors compresos entre 0 (negre) i 255 (blanc), que fa que la imatge no sigui tan pesada i l'algoritme que s'utilitzara necessita que sigui així. Els paràmetres d'entrada son, en ordre, la imatge que es vol convertir, on es vol guardar la imatge convertida i quin es l'espai de colors en que es vol la imatge destí, en aquest cas de blau (Blue), verd (Green) i vermell (Red) a gris.

L'altre processat que se li fa a la imatge es l'equalització de l'histograma[8] el que augmenta el contrast, sense perdre informació estructural, això s'aconsegueix fent que l'histograma de la imatge sigui uniforme, es a dir que per nivell de gris de la imatge hi hagi el mateix numero de píxels. En aquest només hem d'introduir la imatge que volem equalitzar i on la volem guardar, en aquest cas es la mateixa perquè no es necessitara la imatge original.

```
face_cascade.detectMultiScale
(frame_gray, faces, 1.4, 2, 0|CV_HAAR_SCALE_IMAGE, cv::Size(30, 30) );
```

Per últim el que fa aquest tros de codi és agafant la imatge ja tractada, detectar i extreure un vector amb les cares detectades, en el nostre cas només serà una, com s'ha comentat això ho aconseguim per mitja d'una serie de classificadors encadenats. Els paràmetres d'entrada que es necessitessen son: el primer es la imatge prèviament processada i transformada, el segon es el vector on s'emmagatzemaran totes les cares que es trobin, els tercer i quart paràmetres son quan es redueix la imatge a cada successiu classificador i el mínim de veïns que s'han de retenir i s'han trobat a partir de fer proves, el cinquè paràmetre son els flags i el que esta posat es per optimitzar la computació i l'últim paràmetre conte la mínima i màxima mida que tindran les imatges que tenen les cares que es trobaran.

8.2.2 Entrenament o "Training"

Aquesta funció serà la encarregada de du ha terme el entrenament de la maquina per poder fer el reconeixement facial de l'usuari, aquest proces es necessari ja que és creara l'arxiu amb les característiques de l'usuari que després permetrà que el programa l'identifiqui, per fer això el programa agafara la informació de la càmera, farà la detecció amb el codi explicat prèviament i creara l'arxiu anteriorment mencionat per poder fer la identificació, seguidament s'explicara com es fa tot el proces intern.

```
void FacialRecognition::train (const std::string& user)
```

Començarem per la definició, com es pot veure només consta de un paràmetre d'entrada que és un nom d'usuari. Com que el programa està pensat per ser d'usuari únic aquest nom serà escollit per el propi programa, però s'ha deixat preparat per que, en cas de voler-ho, es pogués fer multiusuari sense masses esforços i amb només unes quantes modificacions.

El primer que s'ha de fer és crear el model, això es farà amb la instrucció:

```
cv::Ptr<cv::face::FaceRecognizer> model =
cv::face::createLBPHFaceRecognizer(1,8,8,8,150.0);
```

Amb aquest s'utilitzara un algoritme anomenat Local Binary Patterns Histograms (LBPH)[13] histograma de patrons binaris locals, que el que fa és dividir en cel·les el que s'ha d'analitzar, després analitzar els veïns dels píxels i per cada un escriure un 0 si el valor del píxel central és major que aquest, si no, escriure un 1, el que dona per resultat un número binari de 8 xifres, un cop obtingut aquest número, que es sol convertir en decimal, es fa el histograma del número de cops que apareix cada número en la cel·la analitzada, aquest histograma es pot veure com un vector de 256-dimensional de característiques que és el que ens permetrà fer el reconeixement. Els paràmetres són, en ordre, el radi utilitzat per construir el patró circular, els veïns que és el número de píxels veïns que s'analitzaran, la distància en la coordenada de les x, horitzontal, de número de cel·les, la distància en la coordenada y, vertical, i l'últim és el límit en que es rebutjara que les dues imatges siguin iguals.

```
if (!face_cascade.load("./haarcascade_frontalface_default.xml"))
printf("--(!)Error loading\n");
```

Aquesta instrucció el que fa és carregar un arxiu que conté un classificador prèviament creat que està als arxius de la pròpia llibreria de OpenCV, per evitar possibles problemes amb rutes relatives aquest arxiu s'haurà d'agafar des de `opencv/data/haarcascades/haarcascade_frontalface_default` i copiar-l'ho al directori del programa. En cas d'error es mostrarà per terminal que s'ha produït l'error mentre carregava l'arxiu.

En el segment de codi següent es mostrarà com s'agafaran les mostres per poder crear el fitxer d'entrenament, això ho farà amb un bucle, a cada iteració d'aquest capturara una frame des de la càmera del dispositiu, utilitzara la funció de detecció prèviament explicada i mostrarà per pantalla la imatge captada amb la cara, si és que hi ha, emmarcada per un quadrat, i així fins que es tinguin totes les mostres que hem trobat que eren necessàries per poder fer una bona identificació.

```

while (true) {

    confidence = 0;
    cap.read(frame);
    faces = detect(frame, face_cascade, path);

    if (faces.size() > 0) cv::rectangle(frame, faces[0],
                                        cv::Scalar(255, 255, 255));

    cv::namedWindow("Training");
    imshow("Training", frame);
    cv::waitKey(1);

```

Seguidament es fa una comprovació de si s'ha trobat alguna cara i de si tenim el numero de cares que hem decidit, aquest també s'ha tret fent proves i s'ha trobat que encara que es fes més no millorava significativament i que aquest numero tampoc comportava un temps excessiu per a l'usuari, un cop comprovades aquestes condicions el que fa és una serie de transformacions a la imatge, aquestes transformacions serveixen per intentar minimitzar el màxim l'impacte que te la lluminositat de l'ambient en la imatge, per evitar que no es reconegui la persona quan la lluminositat fos diferent, per això la imatge primer ha de ser transformada a CIE $L^*a^*b^*$ on la L^* es la lluminositat i la a^* i la b^* son les coordenades vermell/verd i coordenades blau/groc respectivament, un cop fet això el que fem es extreure el canal de la l^* separant les tres capes, un cop fet això apliquem l'algoritme CLAHE[11] que es l'acrònim de "Contrast Adaptative Histogram Equalization", que es semblant al que s'ha aplicat en l'apartat anterior però aquest limita el contrast. finalment es tornen a ajuntar les tres capes i ho tornem a transformar a BGR i de BGR a gris i un cop fet això tenim la imatge amb la lluminositat equalitzada i a partir d'aquest agafem la part de la imatge que ens interessa i la guardem al vector de training, també guardem en un altre vector la etiqueta per representar la cara, com que tots seran iguals sempre afegirem un 0.

```

if (faces.size() > 0 && found_faces < 200) {

    std::vector<cv::Mat> lab_planes(3);
    cv::split(lab_image, lab_planes);

    cv::Ptr<cv::CLAHE> clahe = cv::createCLAHE();
    clahe->setClipLimit(4);
    clahe->apply(lab_planes[0], dst);

    dst.copyTo(lab_planes[0]);
    cv::merge(lab_planes, lab_image);

```

```

cv::Mat image_clahe;
cv::cvtColor(lab_image, image_clahe, CV_Lab2BGR);

cvtColor(image_clahe, image_clahe, CV_BGR2GRAY);

aux = image_clahe(faces[0]);

trainImages.push_back(aux);
trainLabels.push_back(0);

```

Un cop tenim les imatges necessàries en el vector d'entrenament i comprovant que encara no s'hagi entrenat el programa, el que s'ha de fer és utilitzar els vectors que s'han omplert a la part anterior per fer l'entrenament. Això ho es fa amb la instrucció:

```

model->train(trainImages, trainLabels);

```

El que es fa en aquest apartat es fer un petit test de reconeixement per poder extreure la confiança que es té, aquesta confiança es un valor numèric que ens indica com de diferent es la imatge que es comprova amb el model, com més aprop del zero més semblant, això servira per posar un límit en que es deixa de considerar que la imatge introduïda pertany al mateix usuari que la del model.

```

model->predict(aux, predicted, confidence);

```

En aquest tros de codi "aux" es on hem guardat la imatge a la que també hem aplicat les transformacions anteriors, "predicted" es la etiqueta de la que el programa creu que es el propietari de la imatge i la "confidence" es la confiança explicada anteriorment.

L'últim que fa aquesta funció es guardar el model que ha fet a partir de les imatges de l'usuari i fer la mitjana de la confiança per establir el límit.

8.2.3 Reconeixement

```

std::string FacialRecognition::recognition(const std::string& user)

```

En aquesta funció es la que fa el reconeixement facial i es en la que es decideix si l'usuari es el que ha de tenir accés o no, i en cas afirmatiu retornara el nom del usuari, que també es el parametre d'entrada, si no es retornara un -1, aquesta funció no s'explicara tan exhaustivament ja que comparteix moltes similituds amb

l'anteriorment explicat.

Una de les coses que canvia es que en aquest es carrega el limit anteriorment calculat.

```
model->load(path + user + ".xml");
model->setThreshold(decision_threshold);
```

En aquest es faran 100 captures de frames des de la cama com a l'anterior, però només es retornara un positiu si les correctes son 60 o més.

```
if (correct_predictions >= 60)
    return user;
else
    return "-1";
```

8.3 KeepassX

En aquest apartat s'explicaran els canvis fets en el programa KeepassX per tal d'afegir el codi que s'ha comentat en l'apartat anterior, per fer-ho amb més comoditat s'ha optat per utilitzar un IDE anomenat QTCrator, que permet modificar més fàcilment les finestres gràfiques. El primer pas que hem de fer es copiar el codi de reconeixement facial, tant el .cpp com el .h, a keepassx/src/core/ ja que aquí es on estan ubicats tots els arxius que utilitza el programa, un altre directori en el que haurem de modificar es el keepassx/src/gui, ja que es allà on estan tots els codis de les interfícies gràfiques, un cop fet això, veurem les modificacions que s'han fet a aquestes per encabir la nova opció de seguretat.

8.3.1 Modificacions a les interfícies gràfiques i al codi

les modificacions que s'han fet a la interfície gràfica son molt simples, només s'ha afegit la opció de, al crear la base de dades de les contrasenyes, a part de les opcions que ja venien predefinides, una amb la típica contrasenya i l'altre amb un arxiu el qual es pot crear, s'ha afegit la opció de "Biometric Password", com podem comprovar en les imatges següents:

En el capítol següent d'aquest document es podran veure els efectes de seleccionar aquestes opcions.

En el codi següent és veu que s'activa al seleccionar la opció de contrasenya biomètrica en la pantalla de creació de una nova base de dades de contrasenyes,

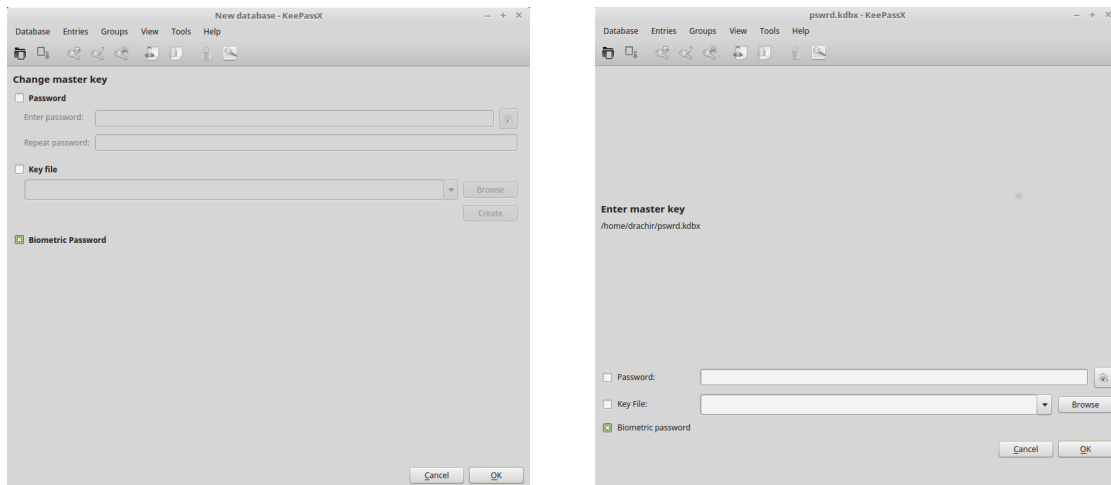


Figure 8.2: Interfícies gràfiques un cop modificades

i com que es una nova la funció que executara es l'entrenament explicat en la secció anterior, el que fa primer de tot es cancel·lar la opció de clicar a el boto de confirmació mentre s'esta realitzant l'entrenament, ja que si no es bloquejava podia causar error en el programa, després el que fa es inicialitzar els paràmetres necessaris, com es pot veure el nom de l'usuari s'ha deixat predeterminat, preo com s'ha comentat anteriorment s'ha deixat preparat per si es vol convertir en multi-usuari, un cop tenim tots els paràmetres el que fa es executar la funció, i un cop creat l'arxiu del model s'utilitza una de les opcions amb les que ja contava el programa, la de utilitzar un fitxer com a contrasenya, per establir el fitxer que s'ha creat com a tal, comprovant que no es produeixi cap error.

```

m_ui->buttonBox->button(QDialogButtonBox::Ok)->setEnabled(false);
QApplication::processEvents();
FileKey fileKey;
std::string user = "User1";
FacialRecognition fr;
fr.train(user);
QString errorMsg;
QString path = "./User1.xml";
if (!fileKey.load(path, &errorMsg)) {
    QMessageBox::critical(this, tr("Failed to set key file"),
        tr("Failed to set %1 as the Key file:\n%2")
            .arg(m_ui->keyFileCombo->currentText(), errorMsg));
    return;
}
m_key.addKey(fileKey);

```

Amb això el que s'ha fet es crear una base de dades que emmagatzema les contrasenyes i que te com a contrasenya principal l'arxiu model de la persona. Un cop fet això ja nom,es falta modificar el codi per obrir aquestes bases de dades i això s'ha fet amb el tros de codi següent:

```

if (m.ui->checkBiometric->isChecked()){
    std::string user = "User1";
    FileKey key;
    FacialRecognition fr;
    std::string recognized = fr.recognition(user);
    if (recognized == user) {
        QString errorMsg;
        QString path = "./User1.xml";
        if (!key.load(path, &errorMsg)) {
            MessageBox::warning(this, tr("Error"), tr("Can't open key file")
                .append("\n").append(errorMsg));

            return CompositeKey();
        }
        masterKey.addKey(key);
    }
    else {
        MessageBox::warning(this, tr("Error"), tr("Access denied"));
        return CompositeKey();
    }
}

```

Que el que fa es utilitzant la funció de reconeixement, per treure un resultat de usuari més possible, agafant el resultat d'aquesta i fent la comparació amb l'usuari, si es correcte segueix amb el funcionament i carrega la clau, en cas de clau incorrecte s'impedirà l'accés.

8.3.2 Modificacions en els makefiles

En aquesta secció s'indiquen els canvis que s'han hagut de fer en els cmakefiles. Els cmakefiles son les instruccions que segueix cmake per tal de generar un makefile, que a la seva vegada son les instruccions de compilació necessàries per generar l'executable del programa. Aquests canvis son necessaris de que es pogués compilar el programa modificat, en aquests s'inclouen les noves dependències que han sorgit per la implementació de OpenCV i per que es reconeguessin els dos nous fitxers de codi introduïts.

```

find_package( OpenCV REQUIRED )

```

`core/facialRecognition.cpp`

El primer ens indica que un nou requisit per la compilació del programa és que estiguin instal·lades les llibreries d'OpenCV. I el segon on esta ubicat el codi que s'ha afegit.

8.3.3 Proves

Totes les proves que s'han fet estat per assaig i error, ja que com requeria la interacció de l'usuari no es podien fer d'altre forma. Per provar el funcionament del programa el que s'ha utilitzat es aquest tros de codi:

```
imshow("aux", aux);  
cv::waitKey(10);  
cout << " Predicted: " << predicted << " Confidence: " << confidence<<endl;*/
```

En el que es mostraven tots els resultats i a partir de les dades obtingudes es variava el numero de mostres. També s'han provat altres algoritmes de reconeixement però no donaven tan bons resultats.

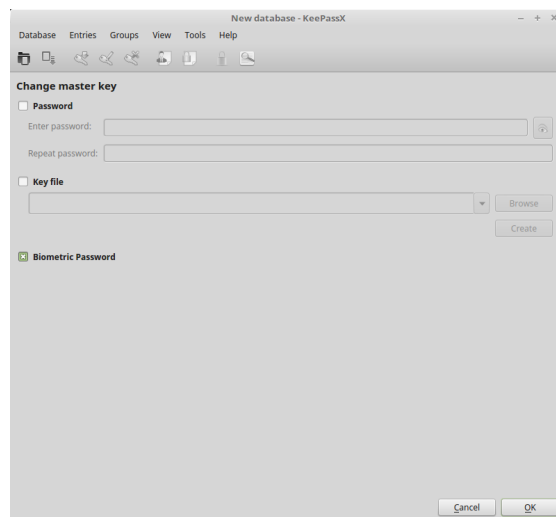
9. Resultats

9.1 Resultats

En aquest capítol s'expliquen les noves funcionalitats del programa amb algunes captures de pantalla.

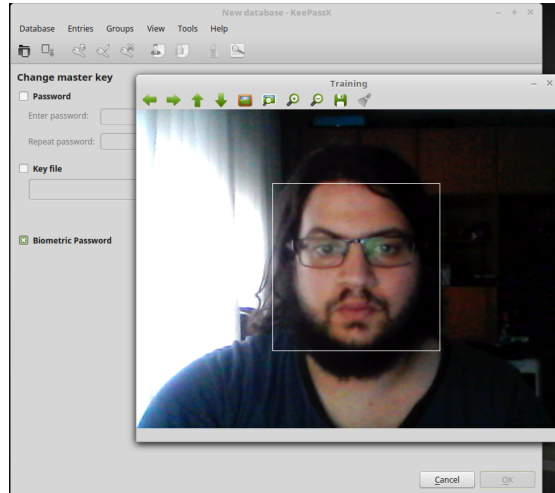
Serà un procés senzill enfocat a l'usuari, per tant no requerirà massa complicació per aquest, però per assegurar una millor fiabilitat amb el reconeixement es recomana moure el cap en diferents posicions ja que així l'entrenament serà més exhaustiu i provocara menys errors en la identificació, també s'ha intentat minimitzar l'impacte en l'usuari, però no s'ha pogut evitar que el procés de creació de la base de dades de contrasenyes sigui d'uns 35 segons i el del procés d'obertura sigui d'uns 15 segons.

9.1.1 Creació de la base de dades

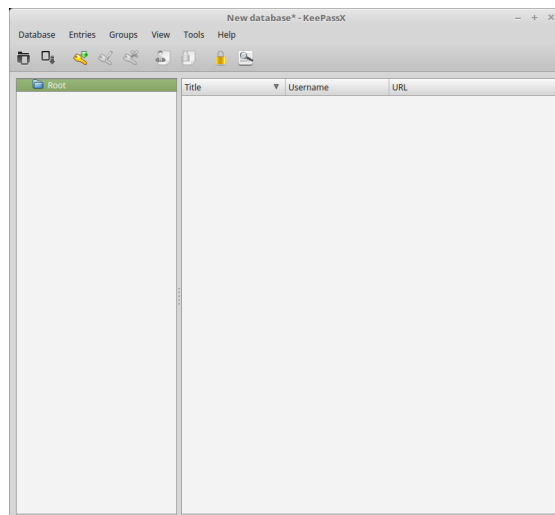


En la imatge anterior es pot veure com es la pantalla de creació de una nova base de dades, com es pot veure hi han tres opcions, password, key file i biometric password, aquestes opcions seran les que defineixin la contrasenya mestre, aquestes

opcions son combinables, es a dir, es pot seleccionar més d'una opció per augmentar la protecció oferida, en el aquest cas la última opció es la que s'ha afegit i si es selecciona i es clica al boto Ok el resultat es:



A la imatge es poden veure algunes de les coses comentades anteriorment, el requadre emmarcant la cara, com es bloqueja el boto per evitar possibles errors, aquesta finestra durara fins que s'hagi fet l'entrenament i un cop fet es tancara automàticament i donara pas a la següent pantalla:

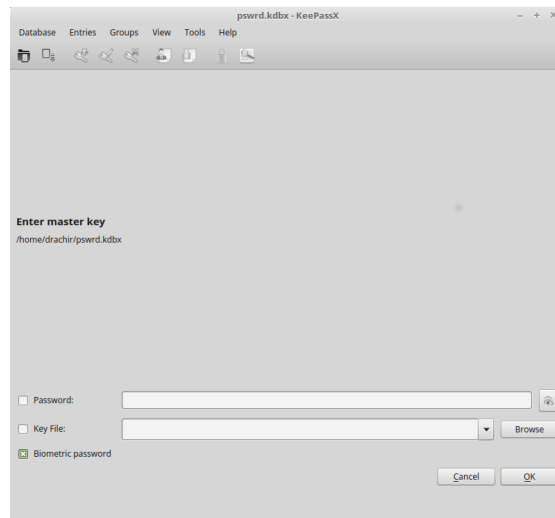


Que ens indica que tot ha funcionat correctament.

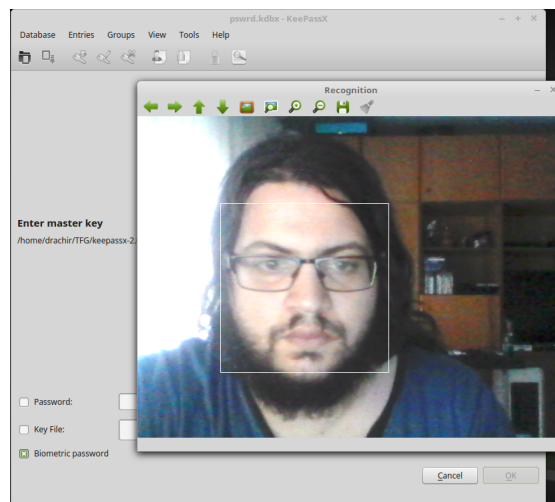
9.1.2 Carregar la base de dades

Ara es mostrara com es carrega una base de dades creada amb la opció de reconeixement facial, primer s'haurà de seleccionar la opció d'obrir base de dades i

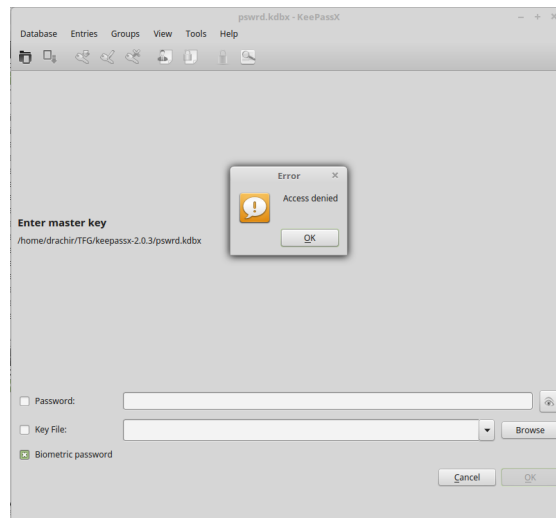
elegir la que volem carregar, i s'obrirà la següent pantalla:



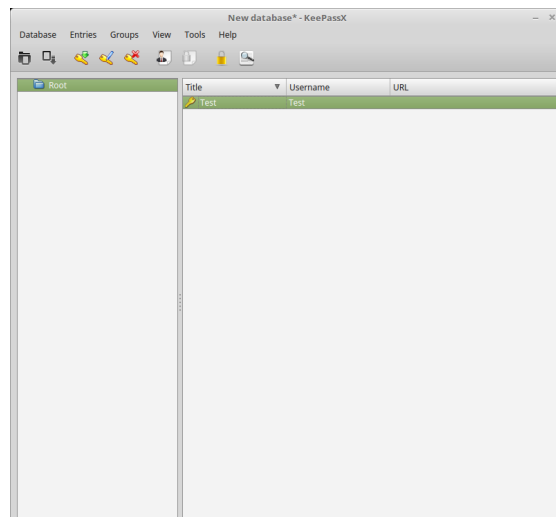
Si es selecciona la opció de biometric password i es clica a ok apareix una pantalla molt similar a la anterior:



Que fa el reconeixement, i si es incorrecte apareix, que ens impedeix la entrada:



En canvi si es correcte accedim directament a la base de dades.



10. Sostenibilitat

En aquest capítol es farà una anàlisi de la sostenibilitat del projecte, responent unes preguntes i a partir d'aquestes preguntes s'omplirà la matriu de sostenibilitat.

10.1 Dimensió Econòmica

- *Existeix una avaluació de costos, tant de recursos materials com humans?*
Sí que existeix l'avaluació de costos, és l'apartat titulat com a Pressupost i allà es tenen en compte tots els recursos necessaris per fer el projecte.
- *S'ha tingut en compte el cost dels ajustaments / actualitzacions / reparacions durant la vida útil del projecte?*
No s'han tingut en compte, ja que en ser un TFG no s'espera haver de fer modificacions un cop acabat el projecte.
- *El cost del projecte ho faria viable si hagués de ser competitiu? Si només es mira per costos probablement sí.*
- *Es podria realitzar un projecte similar en molt menys temps o amb molts menys recursos i, per tant, menor cost?*
En la meua opinió no, els recursos ja són bastant ajustats i el temps del qual es disposa també és bastant escàs.
- *El temps dedicat a cada tasca és proporcional a la seva importància (s'ha dedicat molt de temps a desenvolupar parts del projecte que podien haver estat reutilitzades de tecnologies / projectes / coneixements existents)?*
S'ha intentat ajustar el màxim els temps, per aquest motiu s'utilitzen tecnologies existents per fer algunes funcions del projecte.
- *Està prevista o hi ha col·laboració amb algun altre projecte (acadèmic, empresa, associació, etc.)?*
Col·laboració directa no, però com ja s'ha dit s'utilitzen llibreries de codi obert, ja que si s'haguessin d'implementar totes les funcionalitats, el projecte requeriria més temps de l'assignat.

10.2 Dimensió Social

- *Quina és la situació social i política del país / lloc / ciutat / ... on realitzaràs el teu projecte? I la del sector a què inclou el teu projecte?*
La situació social és estable, no hi ha conflictes armats ni revoltes a gran escala. El sector és un sector emergent, utilitzant tecnologies encara poc utilitzades.
- *Creus que la teva activitat podria afavorir o empitjorar aquesta situació?*
El meu projecte no és cap tecnologia que pugui alterar la situació.
- *Hi ha una necessitat real del teu producte / servei?* Necessitat no, però jo personalment li veig moltes possibilitats.
- *El resultat del projecte, En què / Com canviarà la vida de l'usuari?*
Facilitarà molt la identificació a la xarxa i reduirà la necessitat de contrasenyes.
- *Hi ha algun col·lectiu que es vegi perjudicat pel TFG, i en quina mesura?*
Els únics que poden ser afectats són els gestors de contrasenyes i no significativament.

10.3 Dimensió ambiental

Aquest projecte per ser una solució *Software* no es preveu que tingui cap impacte mediambiental, ni en la seva producció ni en la seva distribució i tots els recursos ja han estat especificats en l'apartat anterior.

10.4 Matriu de sostenibilitat

Sostenibilitat	PPP	Vida útil	Riscos
Ambiental	Consum del disseny 7 [0:10]	Petjada ecològica 12 [0:20]	Riscos ambientals 0 [-20:0]
Econòmica	Factura 7 [0:10]	Pla de viabilitat 13 [0:20]	Riscos econòmics 0 [-20:0]
Social	Impacte personal 9 [0:10]	Impacte social 6 [0:20]	Riscos socials 0 [-20:0]
Valoració total		54 [-60:90]	

Table 10.1: Matriu de sostenibilitat

11. Conclusions i millores futures

En aquest apartat es descriuen les conclusions extretes després de realitzar aquest TFG. Primer es descriuran les conclusions de de la visió de seguretat, seguides de les conclusions personals i finalment es detallarà els plans de treball futurs amb les possibles ampliacions i millores del sistema, que han anat sorgint durant la realització del mateix, però que no s'han pogut dur a terme.

En general, l'objectiu principal del projecte, la substitució de la contrasenya tradicional per paràmetres biomètrics, s'ha complert, amb condicions, s'ha hagut de descartar la idea de la identificació per mitja de la veu ja que no s'ha trobat cap eina que complís els requeriments.

11.1 Conclusions de seguretat

Després de la realització d'aquest projecte no he acabat del tot satisfet ja que encara veig problemàtiques amb la seguretat general del programa, ja que tot es basa en un arxiu que esta en el sistema, encara que protegit, això pot portar problemes de seguretat si algun individu accedeix al sistema, però no tinc les capacitats tècniques per arreglar-ho una altre cosa es el no haver pogut seguir amb l'idea original. Però a part d'això estic molt satisfet.

11.2 Conclusions personals

En aquest TFG he apres les dificultats que es pot tenir en un projecte començant per no trobar les eines que son necessàries per a complir la funció que volem realitzar, fins a petits errors en la programació que comporten dies de retràs, i en aquest sentit estava limitat ja que només era un. Però de la part positiva he apres sobre temes diversos ja que m'he hagut d'endinsar en visió per computador i aprenentatge autònom i he aprofundit el meu coneixement en c++.

11.3 Possibles millores i treball futur

Aquesta es una idea a la que li veig molt de futur i no descarto, en un futur, continuar fent coses per millorar-ho. Aquesta es una petita llista de les coses que jo crec que es podrien millorar o fer de diferent forma:

- Primer de tot seria aconseguir fer la implementació del sistema d'identificació per mitja de la veu com estava originalment plantejat, cosa que augmentaria la seguretat.
- Un altre punt important seria aconseguir blindar l'arxiu de model per que no pugui ser utilitzat per cap altre finalitat.
- Una altra cosa es que es comproves que a les captures hi hagués moviment ja que es podria donar el cas que es poses una imatge per burlar la seguretat.
- Aconseguir que fos una llibreria externa que contingues tot el necessari per executar-se, per així poder-ho fer independent del programa.
- Per ultim es podrien provar altres algoritmes o mètodes de aprenentatge autònom, per millorar els resultats i reduir l'impacte que tenen els factors externs en el reconeixement.

12. Bibliografia

[1]Anthony Dean. 2012. KeePassX: keeping your passwords safe. Linux J. 2012, 213, pages.

[2]Bradski, G. R., Kaehler, A. (2008). Learning OpenCV : Computer vision with the OpenCV library O'Reilly.

[3]Florencio, D., Herley, C. (2007). A large-scale study of web password habits. Proceedings of the 16th International Conference on World Wide Web - WWW '07, pp. 657.

[4]Jain, A. K., Bolle, R., Pankanti, S. (1999). Biometrics : Personal identification in networked society Kluwer.

[5]Jain, A., Hong, L., Pankanti, S. (2000). Biometric identification. Communications of the ACM, 43(2), 90-98.

[6]Morris, R., Thompson, K. (1979). Password security: A case history. Communications of the ACM, 22(11), 594-597.

[7]O'Gorman, L. (2003). Comparing passwords, tokens, and biometrics for user authentication. Proceedings of the IEEE, 91(12), 2021-2040.

[8]Pizer, S., Amburn, E., Austin, J., Cromartie, R., Geselowitz, A., Greer, T., ter Haar Romeny, B., Zimmerman, J. and Zuiderveld, K. (1987). Adaptive histogram equalization and its variations. Computer Vision, Graphics, and Image Processing, 39(3), pp.355-368.

[9]Society., I. S. P., Electrical, I. o., Engineers., E. (2006). IEEE transactions on information forensics and security. IEEE Signal Processing Society.

[10]Technet, M., Technet, M. Virtual private networking: An overview

- [11] Weaver, A. C. (2006). Biometric authentication. *Computer*, 39(2), 96-97.
- [12] Karel Zuiderveld. 1994. Contrast limited adaptive histogram equalization. In *Graphics gems IV*, Paul S. Heckbert (Ed.). Academic Press Professional, Inc., San Diego, CA, USA 474-485.
- [13] Ahonen, T., Hadid, A. and Pietikainen, M. (2006). Face Description with Local Binary Patterns: Application to Face Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12), pp.2037-2041.