

Ismael Molinos Iglesias

**REPOSITORI REMOT D'IMATGES EN DJANGO AMB AUTENTICACIÓ
D'USUARIS PER BIOMETRIA**

TREBALL DE FI DE GRAU

dirigit per Sr. Jordi Duch Gavalrà

Grau d'Enginyeria Informàtica



UNIVERSITAT ROVIRA I VIRGILI

Tarragona

2023

Índex

1	INTRODUCCIÓ	4
1.1	CONTEXTUALITZACIÓ	4
1.2	ALTERNATIVES DEL MERCAT	5
1.3	OBJECTIUS DEL PROJECTE	6
2	DESCRIPCIÓ GENERAL DEL PROJECTE	7
2.1	ENTORN DE DESENVOLUPAMENT	7
3	REQUISITS.....	8
3.1	REQUISITS FUNCIONALS	8
3.2	REQUISITS NO FUNCIONALS	8
3.3	GLOSSARI DELS REQUISITS.....	8
3.4	REGLES DEL NEGOCI.....	8
3.5	ESPECIFICACIÓ TEXTUAL DELS CASOS D'ÚS.....	9
3.5.1	<i>Cas d'ús 01. Crear Directori.....</i>	<i>9</i>
3.5.2	<i>Cas d'ús 02. Afegir Contingut.....</i>	<i>9</i>
3.5.3	<i>Cas d'ús 03. Eliminar contingut.....</i>	<i>10</i>
3.5.4	<i>Cas d'ús 04. Afegir Client/s.....</i>	<i>10</i>
3.5.5	<i>Cas d'ús 05. Eliminar Client.....</i>	<i>10</i>
3.5.6	<i>Cas d'ús 06. Crear Codi Accés Únic.....</i>	<i>11</i>
3.5.7	<i>Cas d'ús 07. Cercar Directori.....</i>	<i>11</i>
3.5.8	<i>Cas d'ús 08. Accedir a un Directori com a Administrador.....</i>	<i>11</i>
3.5.9	<i>Cas d'ús 09. Accedir a un Directori com a Client</i>	<i>11</i>
3.5.10	<i>Cas d'ús 10. Prova d'identitat.....</i>	<i>12</i>
4	ANÀLISI DELS REQUISITS FUNCIONALS	13
4.1	DIAGRAMA DE CASOS D'ÚS:.....	13
4.2	DIAGRAMES DE SEQUÈNCIA	13
5	DISSENY DE L'APLICACIÓ	17
5.1	ARQUITECTURA.....	17
5.2	INTERFÍCIE GRÀFICA	18
5.3	DISSENY DE LA PERSISTÈNCIA DE DADES BD	20
5.3.1	<i>Model Directory</i>	<i>21</i>
5.3.2	<i>Model UniquePass</i>	<i>21</i>
5.3.3	<i>Model Content.....</i>	<i>21</i>
5.3.4	<i>Model Client.....</i>	<i>22</i>
5.3.5	<i>Model LoginAttempt.....</i>	<i>22</i>
5.3.6	<i>Conclusió.....</i>	<i>22</i>
6	IMPLEMENTACIÓ.....	24
6.1	DESENVOLUPAMENT DE LES FUNCIONS DE BIOMETRIA	24
6.1.1	<i>Autenticació del document acreditatiu.....</i>	<i>24</i>
6.1.2	<i>Reconeixement facial.....</i>	<i>25</i>
6.1.3	<i>Reconeixement de caràcters.....</i>	<i>26</i>
6.2	IMPLEMENTACIÓ DE L'APLICACIÓ WEB	28
7	AVALUACIÓ.....	30
7.1	PROVES DE LES FUNCIONS BIOMÈTRIQUES.....	30
7.2	AVALUACIÓ DE LA PÀGINA WEB.....	31
8	CONCLUSIONS.....	34
9	RECURSOS UTILITZATS	36
10	ANNEXES	37

Índex de figures

FIGURA 1. DIAGRAMA DE CLASSES DE LA APLICACIÓ	13
FIGURA 2. DIAGRAMA DE SEQÜENCIA Cd'ús 01.....	13
FIGURA 3. DIAGRAMA DE SEQÜENCIA Cd'ús 02.....	14
FIGURA 4. DIAGRAMA DE SEQÜENCIA Cd'ús 03.....	14
FIGURA 5. DIAGRAMA DE SEQÜENCIA Cd'ús 04.....	14
FIGURA 6. DIAGRAMA DE SEQÜENCIA Cd'ús 05.....	15
FIGURA 7. DIAGRAMA DE SEQÜENCIA Cd'ús 06.....	15
FIGURA 8. DIAGRAMA DE SEQÜENCIA Cd'ús 07.....	15
FIGURA 9. DIAGRAMA DE SEQÜENCIA Cd'ús 09.....	16
FIGURA 10. DIAGRAMA DE SEQÜENCIA Cd'ús 10.....	16
FIGURA 11. DIAGRAMA DE L'ARQUITECTURA D'UNA APLICACIÓ AMB DJANGO	17
FIGURA 12. EXEMPLE DE ETIQUETA, CAMP DE TEXT I BOTÓ SENSE BOOTSRAP.....	18
FIGURA 13. EXEMPLE DE ETIQUETA, CAMP DE TEXT I BOTÓ AMB BOOTSRAP	18
FIGURA 14. RESULTATS DE LOGOTIP D'APLICACIÓ GENERATS PER DALLE-2.....	19
FIGURA 15. LOGOTIP FINAL DE L'APLICACIÓ GENERAT PER DALLE-2	20
FIGURA 16. ESTRUCTURA DELS MODELS I ELS SEUS CAMPS.....	23
FIGURA 17. EXEMPLE D'EXTRACCIÓ DE PUNTS CLAU DE LA TARGETA	24
FIGURA 18. RECONeixEMENT DE LA CARA PAS A PAS I MATRIU 128-DIMENSIONAL DE CARACTERÍSTIQUES FACIALS.	26
FIGURA 19. ARQUITECTURA CRNN PERTANYENT A EASYOCR.....	27
FIGURA 20. PROBLEMA EN LA COMUNICACIÓ ENTRE VISTES.....	29
FIGURA 21. EXEMPLE D'IMATGE D'ENTRADA PER TEST DE ROTACIÓ	30
FIGURA 22. MISSATGE D'ERROR PER INTENT DE TRASPÀS D'IDENTIFICACIÓ	31

1 Introducció

1.1 Contextualització

Avui en dia, hi ha molts processos de vida diària que estan sent automatitzats per permetre als usuaris del servei ser més independents i estalviar temps i esforç. Aquesta conversió digital manca de la relació tradicional cara a cara que succeïa inherentment degut a la forma tradicional de realitzar aquestes transaccions.

En alguns casos, quan un servei requereix d'autenticació d'usuaris i hi ha eines específiques per proporcionar aquesta funcionalitat a una aplicació, [1] la majoria d'elles son solucions de pagament. Aquestes eines proporcionen una prova fiable d'autenticar la identitat dels usuaris, però es poden donar altres casos en què no es necessita un sistema tan robust d'autenticació i el cost d'implementació no val la pena. En aquests casos particulars, la verificació de la identitat usuaris seria útil, però no obligatòria.

Dins d'aquest projecte es proposa desenvolupar un sistema que proporcioni cert nivell d'autenticació d'identitat de l'usuari als serveis on no es tracti d'una part crucial del procés, sinó que seria útil en alguns casos. Un exemple podria fer un examen en línia, on l'estudiant només ha d'estar registrat amb les seves credencials per poder accedir a una prova i algú altre ho podria fer per en el seu lloc, sempre que li compartís les credencials.

Hi ha projectes que també fan reconeixement de targetes d'identificació, però principalment centrats en només un tipus de targeta [2]. A més, hi ha molts projectes que implementen el reconeixement facial, però cap no es fa específicament per a aquest propòsit. Normalment, els desenvolupadors opten per crear la seva pròpia verificació d'identitat, basada en les necessitats específiques dels seus serveis.

Per resoldre els problemes descrits anteriorment, aquest projecte implementa un procés d'autenticació de dues fases que pretén afegir una capa de verificació d'identitat en un entorn en línia. No és de l'àmbit d'aquest projecte crear un sistema irrompible mitjançant la implementació d'un mecanisme criptogràfic molt segur. En canvi, es pretén proporcionar eines útils que pugui ajudar en situacions com les que s'han comentat abans.

Més específicament, la aplicació que es planteja es un exemple de cas d'us del sistema d'autenticació que es proposa. No és una aplicació amb una finalitat comercial ni utilitària en el dia a dia, simplement es un model de com funcionaria i quines parts haurien d'estar involucrades per crear el sistema descrit abans.

La informació principal que es vol extreure és si l'usuari pot proporcionar la targeta d'identificació en el moment d'accedir al servei i també si l'usuari real que accedeix és la persona afirma ser. Per assolir aquests dos punts d'informació clau, es realitza una autenticació de dues fases, corresponent primer a la targeta d'identificació i segon al rostre humà. Aquests dos extractes d'informació seran processats i avaluats. El programa emmagatzema els fotogrames clau que demostren la identitat de l'usuari i alguns altres cares si hi hagués més d'una persona present.

En un ordinador compartit, diferents persones podrien accedir al mateix servei utilitzant que el mateix usuari compartit. Especialment ara, amb la tendència creixent d'estalviar automàticament totes les nostres contrasenyes en línia per tal de que no sigui necessari recordar-les, l'accés multiusuari augmenta i els serveis actuals son incapaçs de distingir-los.

Aquest acte podria passar innocentment o involuntàriament, però podria acabar provocant molts problemes. Hi ha escenaris en què la verificació d'identitat no és una cosa

tan important com per verificar-la en temps real per un ésser humà, però tenir algun tipus de verificació pot ser molt útil per evitar problemes després.

Com es mencionava anteriorment, un exemple d'aquesta situació podria ser tenir un fill i no prendre les mesures necessàries per restringir el seu accés a determinats serveis i que després el nen realitzés compres no autoritzades. En aquest cas concret, hi ha algunes polítiques comercials que no permeten reemborsament dels diners o tenen algun tipus d'impost o inconvenient en el moment de presentar una reclamació. Si s'hagués verificat la identitat de l'usuari abans que fins i tot pogués accedir a l'opció de comprar, s'hauria pogut evitar aquest problema.

A més, un altre problema que està passant amb molta freqüència avui dia és que algú faci un examen o qualsevol tipus de prova en línia en el lloc d'una altra persona. En aquest cas, en aquest cas específic no es fa gaire verificació de la identitat de l'usuari. Un exemple n'és el moodle, que no té cap tipus de verificació més enllà de les credencials.

Un altre dels principals problemes que s'aborda en aquest projecte és la suplantació voluntària d'accés als usuaris, cosa que significa que algun usuari permet que una altra persona accedeixi com a ell mateix en un entorn virtual. Els motius d'aquesta acció fraudulenta podrien ser l'intercanvi de recursos, per exemple en serveis de transmissió en línia o serveis de subscripció pagats de qualsevol altre tipus que tingui accés restringit. D'aquesta manera, el que el sistema percep com a usuari únic podria correspondre a molts usuaris del món real que comparteixen un compte.

Hi ha maneres existents de verificar la identitat d'un usuari, com ara utilitzar la signatura digital. En aquest cas, es resol el problema de la suplantació i es verifica l'autenticitat dels usuaris, però el problema de la presència física dels usuaris encara està present. Normalment, la solució per als exàmens en línia és la vigilància humana mitjançant algun tipus de trucada de vídeo i la validació es fa persona per persona per verificar la identitat de l'usuari. Aquesta solució, tot i que es demostra que és la més eficaç, no és la més pràctica i l'objectiu d'aquest programa és automatitzar aquest procés.

1.2 Alternatives del mercat

Dins del marc especificat ja existeixen diverses solucions a aquests problemes, en el moment de crear una aplicació des de zero son poques les empreses que creen el seu sistema d'identificació amb biometria, sovint perquè no es rentable econòmicament. Una empresa que ofereix un únic tipus de software especialitzat a múltiples clients pot destinar una major part dels recursos a aquest desenvolupament i, per tant, aconseguir un millor resultat.

Malauradament, obtenir informació concreta de les pàgines sobre les solucions que s'ofereixen no és senzill. No tenen informació sobre preus, ja que totes requereixen primer que contactis amb ells per fixar quin és l'abast del projecte que es vol dissenyar.

Totes les solucions tenen punts en comú, verifiquen documents, obtenen la informació clau i ofereixen un sistema antifrau, permeten reconeixement de múltiples fonts i països diferents, compten amb un sistema de xifratge molt potent, entre moltes altres.

Es difícil comparar aquestes solucions entre sí sense haver pogut provar-les però en definitiva és segur que aquestes solucions funcionen de forma transparent al client i que resolen perfectament els requeriments que plantejarem. Per desgràcia no he sigut capaç de trobar cap solució que no sigui de pagament.

1.3 Objectius del projecte

L'objectiu principal del projecte és arribar a crear una solució per al problema introduït anteriorment. Actualment existeixen tecnologies que ens permeten realment crear sistemes funcionals i útils en aquest àmbit. Un nou sistema d'autenticació com el que proposo podria arribar a tindre una utilitat teòrica en algunes situacions del món real.

A més a més, avui en dia se'n parla molt de les noves tecnologies que apareixen al mercat molt ràpidament, però des del meu punt de vista considero que no s'acaben d'integrar en els sistemes que ja estem utilitzant i crec que en aquesta situació existeix una possibilitat de millora respecte al que existeix actualment.

Cal mencionar però que la meua visió sobre aquest projecte no s'enfoca des del punt de vista comercial, tot i que penso que un model similar podria arribar a vendre's, sinó que vull explorar des d'un punt de vista acadèmic, fins a quin punt aquestes tecnologies de les que tant se'n parla estan realment al nostre abast, per a poder utilitzar-les de forma lliure i extraure'n el màxim potencial donant-li'ls noves utilitats.

D'altra banda com a objectiu personal considero que el projecte és una molt bona oportunitat per a consolidar els coneixements obtinguts durant la carrera en àmbits com Bases de Dades, Disseny d'Aplicacions i Programació en general. A més a més, en el meu cas, no he tingut la oportunitat de cursar cap assignatura de programació web. Aquest projecte m'ha servit per complementar els meus coneixements i obtenir una base en aquesta àrea, fins ara desconeguda per mi.

La idea és arribar a realitzar el disseny i el desenvolupament (Frontend¹ i Backend²) de la aplicació des de zero fins a tindre una aplicació funcional. Queda fora de l'objectiu del projecte posar la web en marxa dins d'un servidor i configurar-lo per tal de que quedi accessible des d'internet.

Com a objectiu addicional es troba el domini del Framework³ de Django, un dels frameworks més importants i potents per Python avui en dia. Aquesta habilitat, no només es rellevant per aquest framework en específic, ja que hi ha molta altres frameworks que tenen un funcionament i estructura similar.

D'altra banda, el que per mi és més important és investigar, descobrir i utilitzar diferents tècniques de intel·ligència artificial aplicades a la visió per computador. Hauré de executar i comparar diferents solucions actuals per problemes d'aquest estil, per finalment dissenyar una solució real que combini aquestes tècniques.

¹ Frontend: Part del desenvolupament centrada en la part gràfica de la aplicació.

² Backend: Part del desenvolupament centrada en el tractament de les dades i les funcions de processament que sustenten la aplicació.

³ Framework: Entorn de desenvolupament d'aplicacions que proporciona eines per facilitar aquest procés tals com classes, funcions i preconfiguracions

2 Descripció general del projecte

Aquest projecte de Fi de Grau consta d'una aplicació web en la que es proporciona un servei d'allotjament d'imatges protegit mitjançant autenticació amb biometria, més concretament amb identificació mitjançant un document d'identitat o un carnet acreditatiu.

Dit en altres paraules, la aplicació és com un *Google Drive* o un *Dropbox* d'imatges en el qual els usuaris s'identifiquen amb credencials reals mitjançant un dispositiu amb càmera verificant la seva identitat.

2.1 Entorn de desenvolupament

Per realitzar la part més tècnica de reconeixement facial i processament d'imatges s'ha utilitzat el llenguatge de programació Python, juntament amb els paquets més populars per aquests propòsits com son OpenCV, Face_recognition, EasyOCR

A més a més, per crear la web s'ha utilitzat un dels Frameworks més populars per Python, es tracta de Django, el qual conté tots els components i eines necessaris per muntar una web funcional. Django conté fitxers, classes i configuracions propis que faciliten la creació de la estructura i la posada en marxa.

Pel que fa a l'entorn de programació, s'ha utilitzat PyCharm, un entorn específic de Python, que permet la integració amb Django, facilita la gestió del projecte així com dels paquets utilitzats i permet integrar Git com a controlador de versions i Github com a repositori online de codi.

Tot i que Django proporciona una base de dades ja configurada com a base per crear la web, en aquest projecte he decidit canviar aquesta configuració i adaptar-la per poder utilitzar la meua instància pròpia de MySQL per tal de tindre les dades en un entorn més controlat i accessible.

Per realitzar els testos, s'ha utilitzat principalment el navegador Google Chrome en el cas de la web, per fer les proves de navegació. Per la part de l'anàlisi d'imatges, que s'ha desenvolupat a part, s'ha fer un petit main que permet fer proves en local de les tècniques de reconeixement facial i de processament de imatges abans de haver de fer tota la web i dependre d'aqueta per poder fer les proves.

El projecte en sí està creat dins d'un entorn virtual (Python Virtual environment) que conté un intèrpret de Python 3.9 i que permet fer totes les instal·lacions de paquets de forma aïllada per tal de no emplenar l'emmagatzemament del sistema.

3 Requisits

3.1 Requisits funcionals

- S'han de poder crear directoris
- S'han de poder afegir clients per tal de que accedeixin als directoris
- S'han de poder esborrar clients
- S'ha de poder pujar imatges al directori
- S'han de poder esborrar les imatges
- S'ha de poder cercar directoris
- S'ha de poder accedir als directoris
- Els clients han de poder veure les imatges penjades al directori
- Els administradors han de poder veure quins clients han accedit al directori i veure les imatges generades en el seu accés
- Els administradors han de poder editar les paraules clau dels clients
- Els clients no poden saber qui més té accés al directori
- Els administradors han de ser capaços de crear accessos d'un sol ús per a casos especials

3.2 Requisits no funcionals

- La aplicació ha de ser segura
- Les dades han de ser permanents (base de dades)
- La aplicació ha de ser fàcil d'utilitzar
- La aplicació ha de ser ràpida
- La prova d'accés ha de ser fiable
- La seguretat de la aplicació ha de ser basada en biometria
- La aplicació no ha de tindre errors
- La aplicació ha de poder-se executar en diferents navegadors
- S'ha de poder tancar sessió de forma segura

3.3 Glossari dels requisits

Directorí o repositori: Si bé aquestes dues paraules tenen un significat diferent, durant aquesta secció i també durant tot el projecte s'utilitzen ambdues de forma intercanviable. Si bé cap de les dues definicions s'ajusta del tot al que tenim, que és un servei d'allotjament d'imatges, s'entén en quin context s'utilitzen.

Contingut: Únicament es tracta d'imatges, en un futur podria ampliar-se els tipus de contingut a altres tipus de fitxers.

Intent d'accés: Conjunt de la informació generada al moment en què un **Client** realitza un accés a un **Directorí**, consisteix en les imatges que certifiquen la seva identitat.

Codi d'accés únic: Conjunt de paraules que permeten fer un accés de **Client** al **Directorí** sense haver de fer la prova d'accés.

3.4 Regles del negoci

- No pot existir un **Directorí** sense **Administrador**.
- Cada **Client** pertany a un sol **Directorí**.

- No poden existir dos **Directoris** amb el mateix nom.
- Si s'elimina un **Directori**, s'eliminarà l'**Administrador**, tots els **Clients** i tots els **Continguts** del **Directori**.

3.5 Especificació Textual dels Casos d'ús

3.5.1 Cas d'ús 01. Crear Directori

Resum de la funcionalitat: Es crea un nou Directori juntament amb un Administrador per poder emmagatzemar imatges.

Paràmetres d'entrada: cap

Paràmetres de sortida: cap

Usuaris: Administrador

Precondició: cap.

Postcondició: S'ha creat un Directori.

Procés normal principal:

1. El sistema demana el nom del directori (obligatori) i els suggeriment de format (opcional).
2. El **administrador** introdueix, com a mínim, el nom del directori.
3. El sistema crea el **Directori** correctament.
4. El sistema demana el document acreditatiu de **L'Administrador**
5. **L'Administrador** proporciona una imatge del document
6. El sistema verifica que el document contingui un rostre i emmagatzema el document.
7. El sistema executa el Cd'ús 10.

Alternatives de procés i excepcions:

- 3a. El nom introduït ja existeix al sistema.
 - 3a1. El sistema mostra un missatge d'error i torna demanar el nom
 - 3a2. Torna al pas 2.
- 6a. La imatge proporcionada no és vàlida.
 - 6a1. Torna al pas 4 i mostra l'error.

3.5.2 Cas d'ús 02. Afegir Contingut

Resum de la funcionalitat: l'**Administrador** afegeix una imatge al **Directori**

Paràmetres d'entrada: cap

Paràmetres de sortida: cap

Usuaris: Administrador

Precondició: Administrador registrat i identificat al sistema .

Postcondició: S'ha afegit una imatge al directori.

Procés normal principal:

1. El sistema demana el fitxer (imatge) per penjar.
2. L'**administrador** selecciona la imatge.
3. El sistema els mostra el nom del fitxer a penjar.
4. L'**administrador** confirma que vol pujar el fitxer.
5. El sistema puja la imatge al **Directori** i mostra el **Directori** amb la imatge nova.

3.5.3 Cas d'ús 03. Eliminar contingut

Resum de la funcionalitat: l'Administrador elimina una imatge del **Directori**

Paràmetres d'entrada: cap

Paràmetres de sortida: cap

Usuaris: Administrador

Precondició: Administrador registrat i identificat al sistema .

Postcondició: S'ha eliminat una imatge del directori.

Procés normal principal:

1. El sistema mostra les imatges actuals.
2. L'**administrador** clica la imatge que vol esborrar.
3. El sistema elimina la imatge.

3.5.4 Cas d'ús 04. Afegir Client/s

Resum de la funcionalitat: l'Administrador afegeix nous clients al Directori

Paràmetres d'entrada: cap

Paràmetres de sortida: cap

Usuaris: Administrador

Precondició: Administrador registrat i identificat al sistema .

Postcondició: S'han afegit nous clients

Procés normal principal:

1. El sistema mostra la llista de clients.
2. L'**administrador** clica afegir clients.
3. El sistema mostra la pantalla d'afegir clients.
4. L'**administrador** afegeix les Paraules Clau del client
5. El sistema crea un client amb aquestes paraules clau

Alternatives de procés i excepcions:

- 4a. L'**Administrador** vol afegir clients en un fitxer csv.
 - 3a1. El sistema mostra la opció "afegir per csv".
 - 3a2. L'**administrador** selecciona el fitxer desitjat
 - 3a3. El sistema afegeix els **Clients** al sistema.

3.5.5 Cas d'ús 05. Eliminar Client

Resum de la funcionalitat: l'Administrador elimina un **Client** del sistema.

Paràmetres d'entrada: cap

Paràmetres de sortida: cap

Usuaris: Administrador

Precondició: Administrador registrat i identificat al sistema .

Postcondició: S'ha eliminat un **Client** del sistema

Procés normal principal:

1. El sistema mostra la llista de clients.
2. L'**administrador** clica eliminar al **Client** desitjat.
3. El sistema elimina el **Client**.
4. El sistema mostra la Llista de **Clients** actualitzada

3.5.6 Cas d'ús 06. Crear Codi Accés Únic

Resum de la funcionalitat: L'Administrador crea un Codi d'accés únic al Directori

Paràmetres d'entrada: cap

Paràmetres de sortida: cap

Usuaris: Administrador

Precondició: Administrador registrat i identificat al sistema .

Postcondició: S'ha creat un codi d'accés únic

Procés normal principal:

1. El sistema mostra la opció de crear accés únic.
2. L'administrador afegeix les paraules del codi.
3. El sistema emmagatzema el codi.

3.5.7 Cas d'ús 07. Cercar Directori

Resum de la funcionalitat: L'usuari cerca un directori per accedir

Paràmetres d'entrada: cap

Paràmetres de sortida: Directori cercat

Usuaris: Administrador i Client

Precondició: cap

Postcondició: cap

Procés normal principal:

1. El sistema mostra la barra de cerca
2. L'usuari introdueix el nom o part del nom del Directori.
3. El sistema mostra els resultats més similars a la cerca.

Alternatives de procés i excepcions:

4a. L'usuari no troba el Directori que buscava:

3a1.Torna al pas 1.

3.5.8 Cas d'ús 08. Accedir a un Directori com a Administrador

Resum de la funcionalitat: L'Administrador accedeix al seu Directori

Paràmetres d'entrada: El Directori seleccionat

Paràmetres de sortida: cap

Usuaris: Administrador

Precondició: El Directori ha d'existir

Postcondició: cap

Procés normal principal:

1. El sistema realitza el Cas d'ús 10
2. El sistema verifica l'usuari
3. L'Administrador accedeix al Directori

3.5.9 Cas d'ús 09. Accedir a un Directori com a Client

Resum de la funcionalitat: El Client accedeix al seu Directori

Paràmetres d'entrada: El Directori seleccionat

Paràmetres de sortida: cap

Usuaris: Client

Precondició: El **Director** ha d'existir

Postcondició: cap

Procés normal principal:

1. El sistema obté les credencials del **Client**
2. El sistema demana les paraules clau del **Client**
3. **El Client** introdueix les seves paraules clau.
4. El sistema verifica les paraules clau
5. El sistema executa el Cas d'ús 10.
6. El sistema verifica l'usuari
7. El **Client** accedeix al **Director**

3.5.10 Cas d'ús 10. Prova d'identitat

Resum de la funcionalitat: El sistema verifica la identitat de l'usuari

Paràmetres d'entrada: L'usuari que vol accedir i les credencials

Paràmetres de sortida: cap

Usuaris: Administrador i Client

Precondició: L'usuari ha de tindre credencials

Postcondició: cap

Procés normal principal:

1. El sistema es connecta amb la càmera de **l'usuari**
2. El sistema demana les credencials a **l'usuari**
3. **L'usuari** mostra les credencials
4. El sistema verifica les credencials.
5. El sistema demana a l'usuari que mostri el seu rostre.
6. **L'usuari** mostra el rostre
7. El sistema verifica el rostre i finalitza

Alternatives de procés i excepcions:

1a. Error amb la càmera:

1a1. **L'usuari** no es pot identificar, torna error

4a i 7a. Les credencials son incorrectes:

4a1 i 7a1: Torna al pas anterior o bé tanca la sessió

4 Anàlisi dels requisits funcionals

4.1 Diagrama de Casos D'ús:

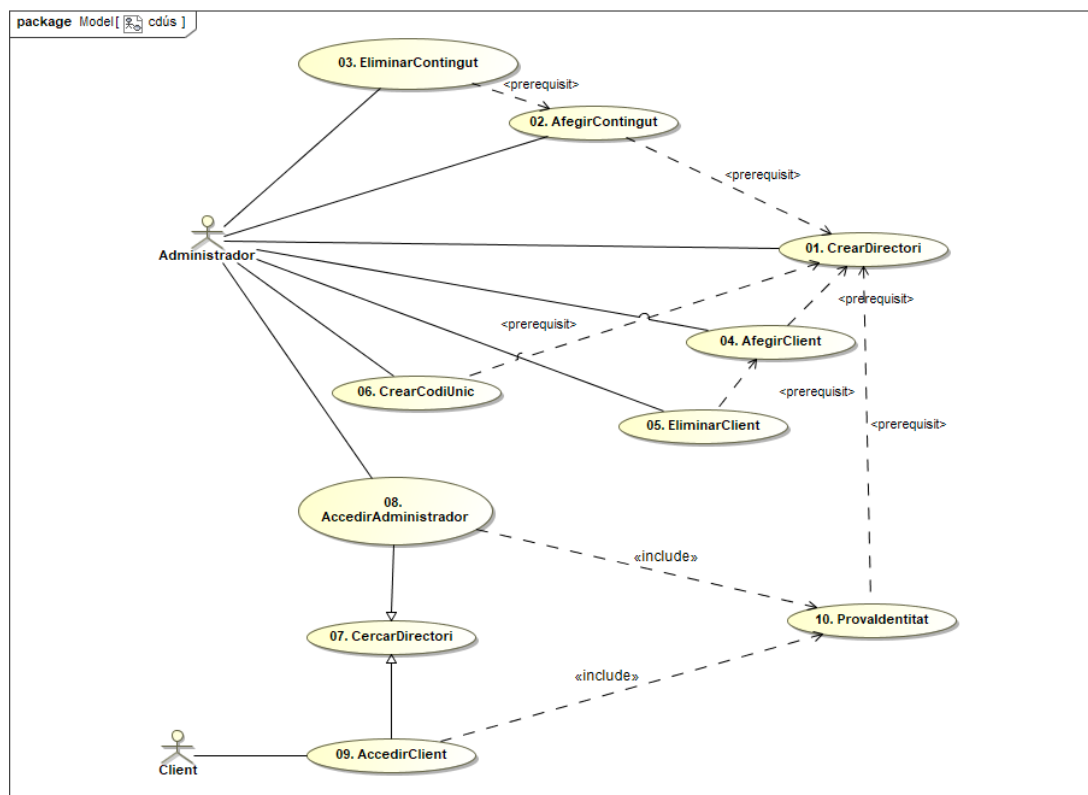


Figura 1. Diagrama de Classes de la Aplicació

4.2 Diagrames de seqüència

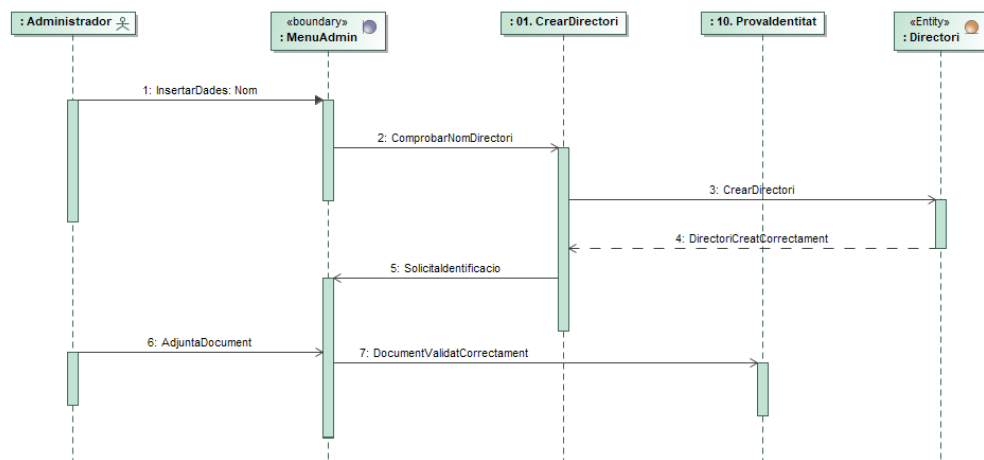


Figura 2. Diagrama de Seqüència Cd'ús 01

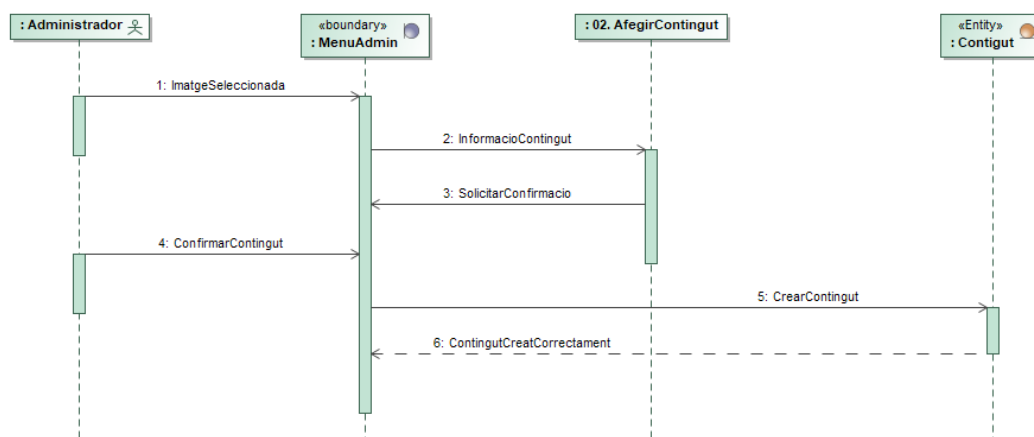


Figura 3. Diagrama de Seqüència Cd'ús 02

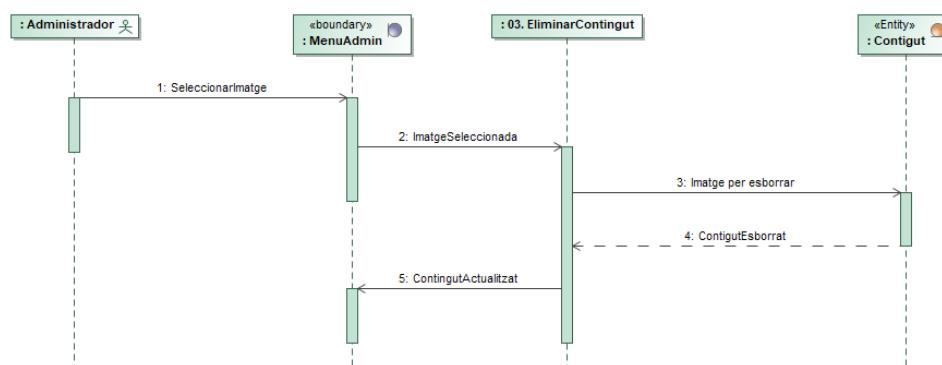


Figura 4. Diagrama de Seqüència Cd'ús 03

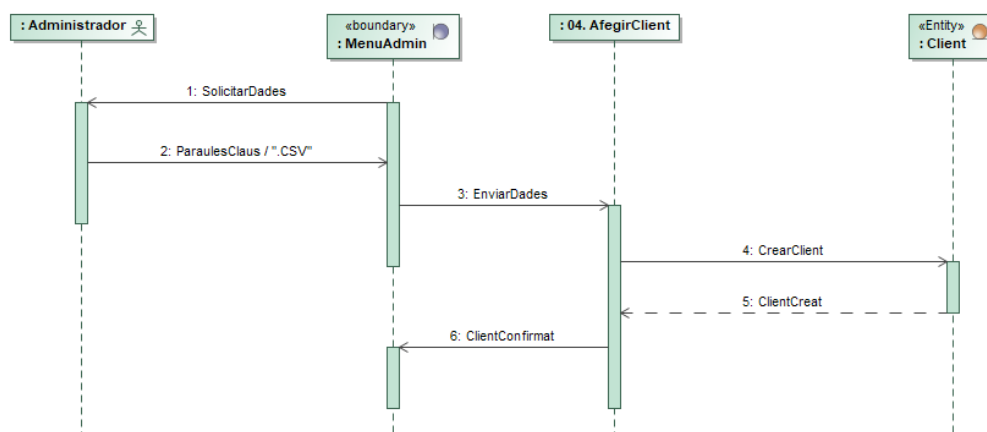


Figura 5. Diagrama de Seqüència Cd'ús 04

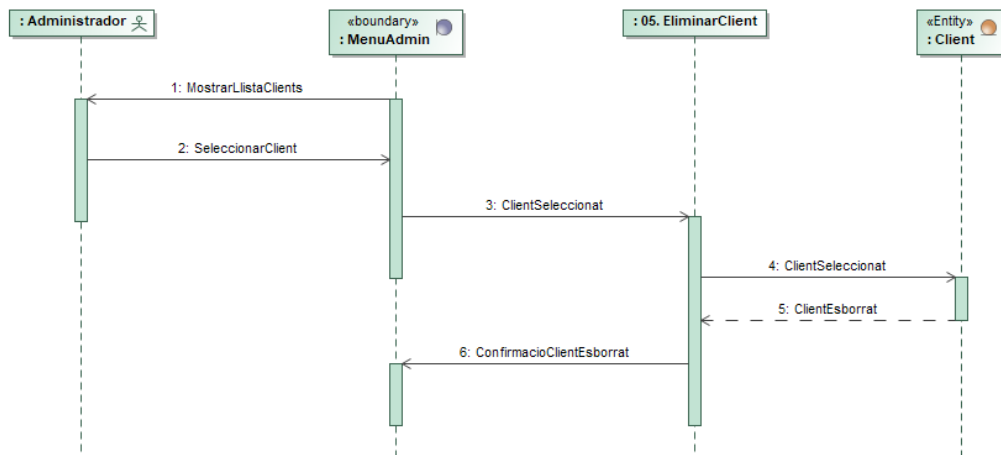


Figura 6. Diagrama de Seqüència Cd'ús 05

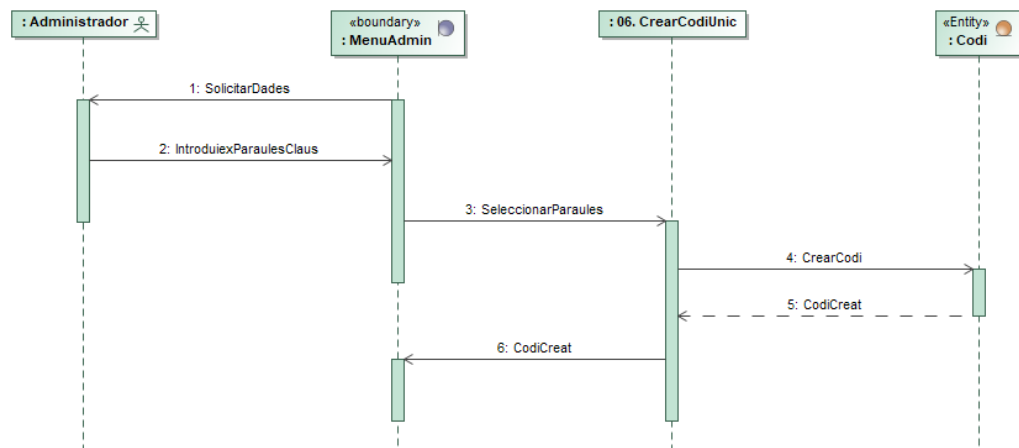


Figura 7. Diagrama de Seqüència Cd'ús 06

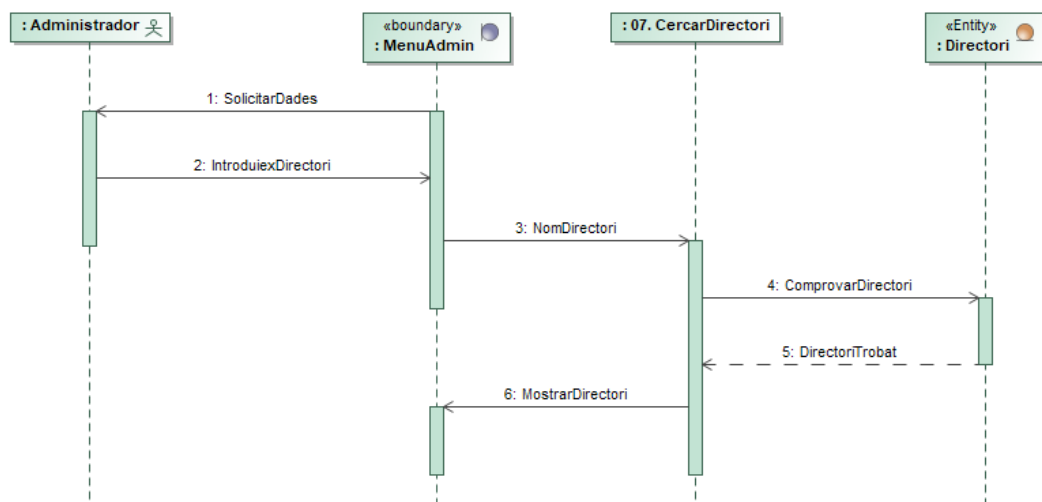


Figura 8. Diagrama de Seqüència Cd'ús 07

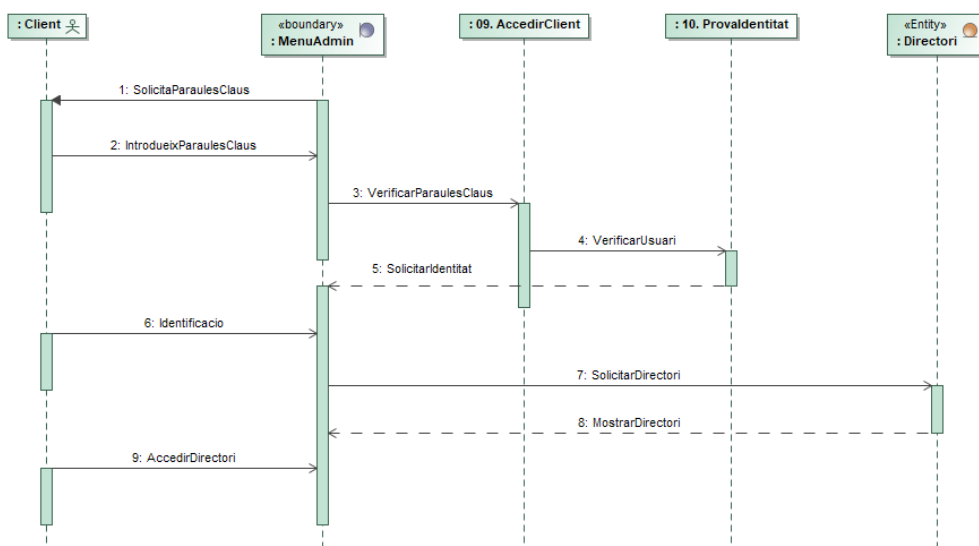


Figura 9. Diagrama de Seqüència Cd'ús 09

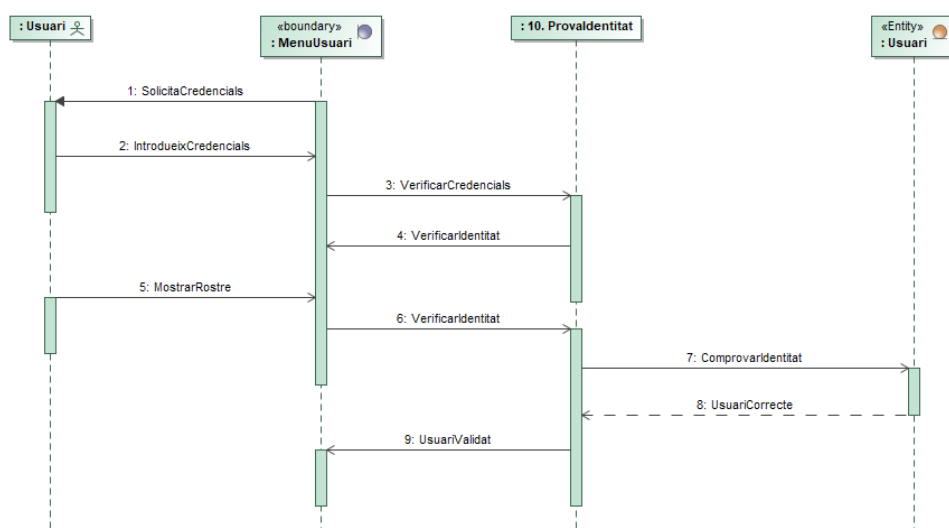


Figura 10. Diagrama de Seqüència Cd'ús 10

5 Disseny de l'aplicació

5.1 Arquitectura

Per realitzar aquesta part del projecte i com ja s'ha mencionat a la introducció, ens basarem en les eines proporcionades pel framework de Django per fer la gestió del Backend, per construir tota la arquitectura i per dissenyar el model de BD de la aplicació.

Començant per la part de la arquitectura, Django funciona amb un model on es permet crear tota la lògica necessària per la nostra aplicació. Com tots els projectes Django, tenim diferents fitxers que interactuen entre ells per poder obtenir i organitzar el funcionament de la aplicació de la forma més ordenada possible.

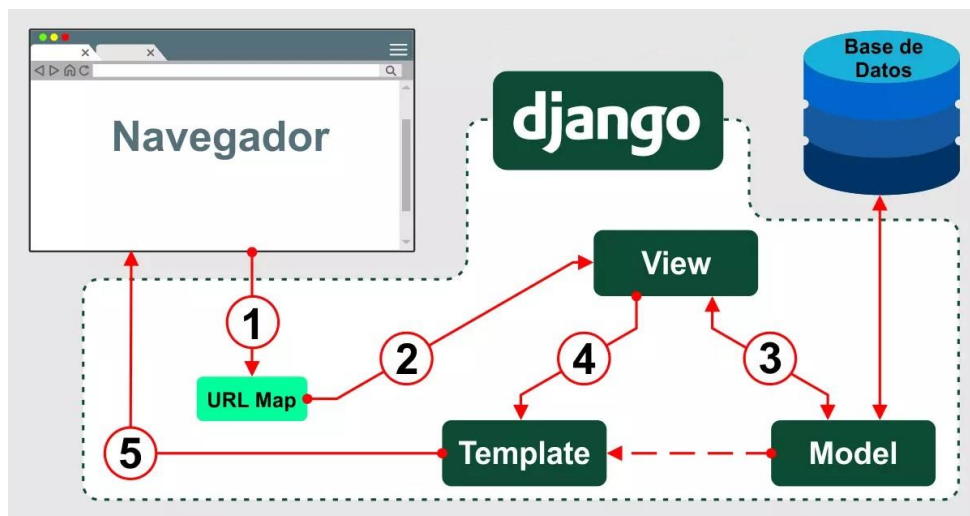


Figura 11. Diagrama de l'arquitectura d'una aplicació amb Django

Per fer un petit resum de les parts involucrades tenim:

- **Templates:** fitxers HTML que contenen els elements gràfics de cada pantalla de la aplicació (bonons, quadres de text, imatges, ...)
- **Views:** Funcions del Backend que permeten la gestió dels objectes (Models) i les dades, preparant-les per als Templates.
- **Urls:** Fitxers de definició de les direccions que conté la nostra web, actua també com a nexce de unió entre les Views i els Templates.
- **Models:** Definició dels objectes de la aplicació, aquest objectes guarden una relació directa amb la estructura de la base de dades.

L'estructura de la aplicació, correspon en gran part amb el definit ens els casos d'ús, per cadascun s'utilitza un o més de cadascun dels elements anteriors.

Dins de l'estructura general, hi ha dos parts diferenciades, la part corresponent a la prova d'identitat i la part posterior, corresponen a la gestió del directori. Cadascuna és una Aplicació diferent dins de l'estructura del Projecte, per tan de separar els model, les vistes i les urls de cada part i tindre una estructura més clara.

5.2 Interfície gràfica

Aquesta part no ha sigut el focus del treball, no obstant he intentat crear una aplicació d'estil simple però interessant a la vista.

En el cas de la meua aplicació, la part de la Interfície gràfica l'he realitzat tota purament definint els elements necessaris de cada pàgina en HTML i els hi he donat estil amb els tags “style” i els atributs corresponents, a més d'incloure alguns elements de Bootstrap per millorar la aparença general dels elements.

Bootstrap és un Framework de disseny de pàgines web que proporciona una aparença molt més moderna als elements de la pàgina, s'afegeix com un enllaç dins de la pàgina web i el navegador ja n'obté la informació en línia a l'hora de crear la pàgina per mostrar-la.

Per utilitzar Bootstrap, dins dels elements de la pàgina has d'afegir el camp “class=...” i triar una de les classes que proporciona Bootstrap per tal d'aconseguir l'estil desitjat. Cada classe té associada un estil de CSS⁴.



The image shows a web form with the title "Cerca un repositori" in a large, bold, black serif font. Below the title is a text input field with the placeholder text "Nom del repositori..." in a blue serif font. To the right of the input field is a blue rectangular button with the word "Enviar" in white sans-serif font.

Figura 12. Exemple de etiqueta, camp de text i botó sense Bootstrap



The image shows a web form with the title "Cerca un repositori" in a large, bold, black sans-serif font. Below the title is a text input field with the placeholder text "Nom del repositori..." in a grey sans-serif font. To the right of the input field is a blue rectangular button with the word "Enviar" in white sans-serif font.

Figura 13. Exemple de etiqueta, camp de text i botó amb Bootstrap

Pel que fa al disseny general, les pàgines son totes de fons blanc i els elements son blaus a excepció d'alguns botons verds que representen una acció que comporta enviar algun tipus de dades o confirmació. El tipus de lletra, la mida, les formes i l'aparença dels elements no els he especificat jo expressament sinó que formen part de l'estil que li dona Bootstrap a les pàgines.

Durant tota la fase de disseny de la Interfície he intentat que tots els elements dins de cada pàgina quedessin compensats, que s'entengués com funciona la aplicació i per a què serveix cada botó. En alguns casos, he hagut de crear pàgines addicionals per encabir

⁴ CSS: Cascading Style Sheets: Llenguatge creat per a donar format als elements HTML d'una pàgina web

funcionalitats quan en realitat no m'hauria fet falta, ja que dins d'una mateixa pantalla hauria pogut fer tota la funcionalitat de l'administrador. Si ho hagués fet d'aquesta manera hagués sigut més senzill des del punt de vista de la programació, però molt més confús des del punt de vista de l'usuari.

Una altra part que caldria destacar és el logotip de la aplicació. Aprofitant que la aplicació utilitza intel·ligència artificial, he decidit seguir amb la temàtica i crear un logotip utilitzant una de les intel·ligències artificials del moment com és DALLÉ-2.

Aquesta eina en principi és de pagament, però mitjançant el xat de Bing de Microsoft (un dels grans inversors de OpenAI, la empresa creador de DALLÉ-2), es pot utilitzar de forma gratuïta i oberta.

El procés de creació del logotip ha consistit en una sèrie de iteracions on li proposava diferents enunciats amb l'objectiu d'anar afinant el resultat fins aconseguir un logo que fos convincent. El logo havia de contenir la essència de la aplicació, fent èmfasi en les credencials o la persona com a clau de la seguretat.

Després d'intentar en anglès i aconseguir pocs resultats, en castellà vaig aconseguir uns resultats satisfactoris dels quals vaig triar un que fos minimalista i que en certa forma contingués la idea que havia proposat.



Figura 14. Resultats de logotip d'aplicació generats per DALLÉ-2

Finalment vaig aconseguir un resultat decent amb l'enunciat següent:

“Un logo minimalista, azul y que transmita la idea de que la identidad del usuario es la llave para acceder, con elementos como un candado o una cerradura y al usuario como la llave para acceder, con más énfasis en la figura del usuario y más claridad en el diseño”



Figura 15. Logotip final de l'aplicació generat per DALLE-2

5.3 Disseny de la persistència de dades BD

Respecte a la persistència de dades, gràcies a Django no he hagut de crear totes les taules a mà directament a la base de dades. He utilitzat la classe `django.db.models.Model` i he creat subclasses que contenen camps estàndard que poden ser creats directament a la base de dades. Per tant, el disseny de persistència de dades es troba en forma de models de Django.

A més Django conta amb una sèrie de dades pròpies que es creen a la base de dades per defecte, algunes de les quals sí que he utilitzat explícitament durant el projecte com ara `Users` o `Request`. Tot i això, la informació rellevant per els casos d'ús es guarda als models creats per mi.

En aquesta secció també cal mencionar el tema de les imatges. A la meva aplicació, les imatges s'emmagatzemen localment dins d'un directori determinat amb un path propi generat a partir de a què correspongui la imatge. Això es degut a que una base de dades SQL no permet l'emmagatzemament directe d'una imatge dins d'una taula, així dons només s'emmagatzema una referència a la ubicació local d'aquest arxiu dins del projecte.

Per el tema de les imatges, existeix un directori dins del projecte que es diu `"/directories/"`, dins d'aquest trobarem els subdirectoris corresponents amb els noms del repositoris creats dins de l'aplicació. Cadascun conté certs subdirectoris més que corresponen amb les credencials de l'administrador, les credencials dels clients i un subdirectori per cada intent que realitza cada client. Finalment dins de cadascun d'aquests subdirectoris trobem les imatges que corresponen.

En cas de que l'aplicació es poses en funcionament en un servidor i s'obris per tal de que els usuaris la utilitzin, caldria fer modificacions per tal de millorar aquest emmagatzemament. No està dins de l'abast d'aquest projecte posar la aplicació en marxa, però si fos així optaria per un emmagatzemament al núvol, de l'estil AWS o Microsoft Azure. Una altra alternativa seria una base de dades NoSQL⁵, com ara MongoDB. En qualsevol cas, Django permetria generar el codi necessari per replicar els models a qualsevol d'aquestes plataformes amb la comanda `'manage.py makemigrations'`.

⁵ NoSQL: és un enfocament de disseny de base de dades que permet emmagatzemar i consultar dades fora de les estructures tradicionals que es troben a les bases de dades relacionals, per exemple, imatges.

A continuació faré una petita explicació de cada model (que correspondria amb el disseny de les Taules) així com una explicació de la seva utilitat i dels seus camps.

5.3.1 Model Directory

Correspon al directori, dins de la jerarquia de taules és la més important ja que tres de les següents la referencien. És la base de la aplicació, els clients pertanyen a un directori, el contingut pertany a un directori.

Camps:

name = models.CharField(max_length=100, unique=True): Correspon al nom del directori i a més l'identifica ja que no poden existir dos directoris amb el mateix nom.

suggestion = models.CharField(max_length=200, blank=True): Suggerència de format, text que es mostra a l'usuari a l'hora d'introduir les seves credencials. En aquest cas, no és un camp obligatori ja que no és crucial per al funcionament de la web, això s'expressa amb l'atribut "blank=True".

admin_credentials = models.ImageField(upload_to=credentials_admin_path, null=True, blank=True): Imatge de les credencials de l'administrador.

user = models.ForeignKey(User, blank=True, null=True, on_delete=models.SET_NULL): Referència a la classe User interna de Django, aquesta associació permet fer comprovacions de sessió i evitar que usuaris no autoritzats o no identificats accedeixin a la part d'administració del directori. Aquí veiem per primer cop l'atribut "on_delete", aquest equival al que en una base de dades seria la política d'eliminació, en aquest cas, no volem que s'esborri tot el Directori, per tant deixariem el camp a null, tot i que dins de la nostra aplicació no hi ha cap situació on es pugui eliminar un usuari (referint-me a la classe usuari de Django).

5.3.2 Model UniquePass

Aquest és el model més senzill de tots, serveix per emmagatzemar un codi d'accés únic al directori. I dins del que és la base de dades, aquest objecte hauria de ser temporal, ja que un cop utilitzat s'esborra del sistema. En el cas de estendre l'aplicació, una de les millores en emmagatzematge seria limitar en nombre màxim d'instàncies d'aquesta classe que es poden crear per directori així com eliminar les instàncies no utilitzades després d'un cert temps.

Camps:

dir = models.ForeignKey(Directory, on_delete=models.CASCADE): Directori al qual pertany el codi. Aquí veiem un altre dels possibles valors de l'atribut "on_delete", en aquest cas, en eliminar un directori, tots els codis d'accés a aquest també haurien de desaparèixer.

pass_code = models.CharField(max_length=100): Correspon al text que utilitzarà el client per accedir.

5.3.3 Model Content

Aquest model representa les imatges pujades dins del Directori, igual que en el model anterior, un sol directori pot contenir múltiples imatges, per tant tenim una relació 1 a molts, que es tradueix en una clau forana a la taula única (el Directori).

Camps:

`directory = models.ForeignKey(Directory, on_delete = models.CASCADE):` Exactament igual que en el cas anterior, amb la mateixa política d'esborrament

`image = models.ImageField(upload_to=content_directory_path, default=None):` La imatge penjada al directori.

5.3.4 Model Client

Aquest model representa els clients que l'administrador ha autoritzat a accedir al repositori, aquest però hauran de coincidir amb les paraules clau que l'administrador ha fixat.

Camps:

key_words = `models.CharField(max_length=200):` Les paraules clau definides per l'administrador, aquest és el camp més important dins de la arquitectura de la aplicació i un dels punts claus del disseny, però a nivell de base de dades, és un camp de text senzill.

directory = `models.ForeignKey(Directory, on_delete=models.CASCADE):` Com anteriorment una referència a una instància del model Directori

credentials = `models.ImageField(upload_to=credentials_directory_path, blank=True, null=True):` Aquest camp correspon a la imatge del carnet o document del client, es permet que sigui null ja que no s'inicialitza fins que el client faci el seu primer accés i proporcioni una **imatge** que compleixi els requisits.

user = `models.ForeignKey(User, blank=True, null=True, on_delete = models.SET_NULL):` Exactament igual que l'usuari de l'Administrador

5.3.5 Model LoginAttempt

L'últim dels models creats per mi, es crearà una instància d'aquest objecte (que equivaldria a una entrada a la taula a la BD), cada cop que un client faci un accés profitós al sistema. El propòsit es emmagatzemar les imatges recollides en el moment de l'accés.

client = `models.ForeignKey(Client, on_delete=models.CASCADE):` Client que fa l'accés.

time = `models.DateTimeField(auto_now=True):` Hora en la que s'ha produït l'intent, podria ser rellevant en alguns casos. En aquest cas, a l'hora de crear entrades a la taula no cal afegir aquest camp explícitament ja que amb l'atribut *auto_now* aquest camp es completa amb l'hora actual per defecte.

id_frame = `models.ImageField(upload_to=client_id_directory_path, blank=True, null=True):` Imatge del moment en que es confirma el document

face_frame = `models.ImageField(upload_to=client_face_directory_path, blank=True, null=True):` Imatge del moment en que es confirma el rostre de l'usuari

failed_frame1, failed_frame2, failed_frame3, failed_frame4, failed_frame5 = `models.ImageField(upload_to=client_failed_directory_path, blank=True, null=True):` Imatges d'altres rostres detectats durant la comprovació, aquests poden o no existir, per tant s'afegeix els atributs `null=True` i `blank=True` per evitar errors durant la creació.

5.3.6 Conclusió

Cal mencionar que tots els camps, objectes, models i atributs del codi es troben en anglès tot i que la aplicació sigui en català, ja que és una bona pràxis.

Tots els models contenen un mètode toString que permet la seva visualització dins de la interfície d'Administrador de Django que permet al desenvolupador crear instàncies noves a la base de dades mitjançant una interfície gràfica. Durant el desenvolupament aquesta ha sigut una de les eines més útils sobretot abans de crear les pàgines i les vistes que permeten la creació d'objectes.

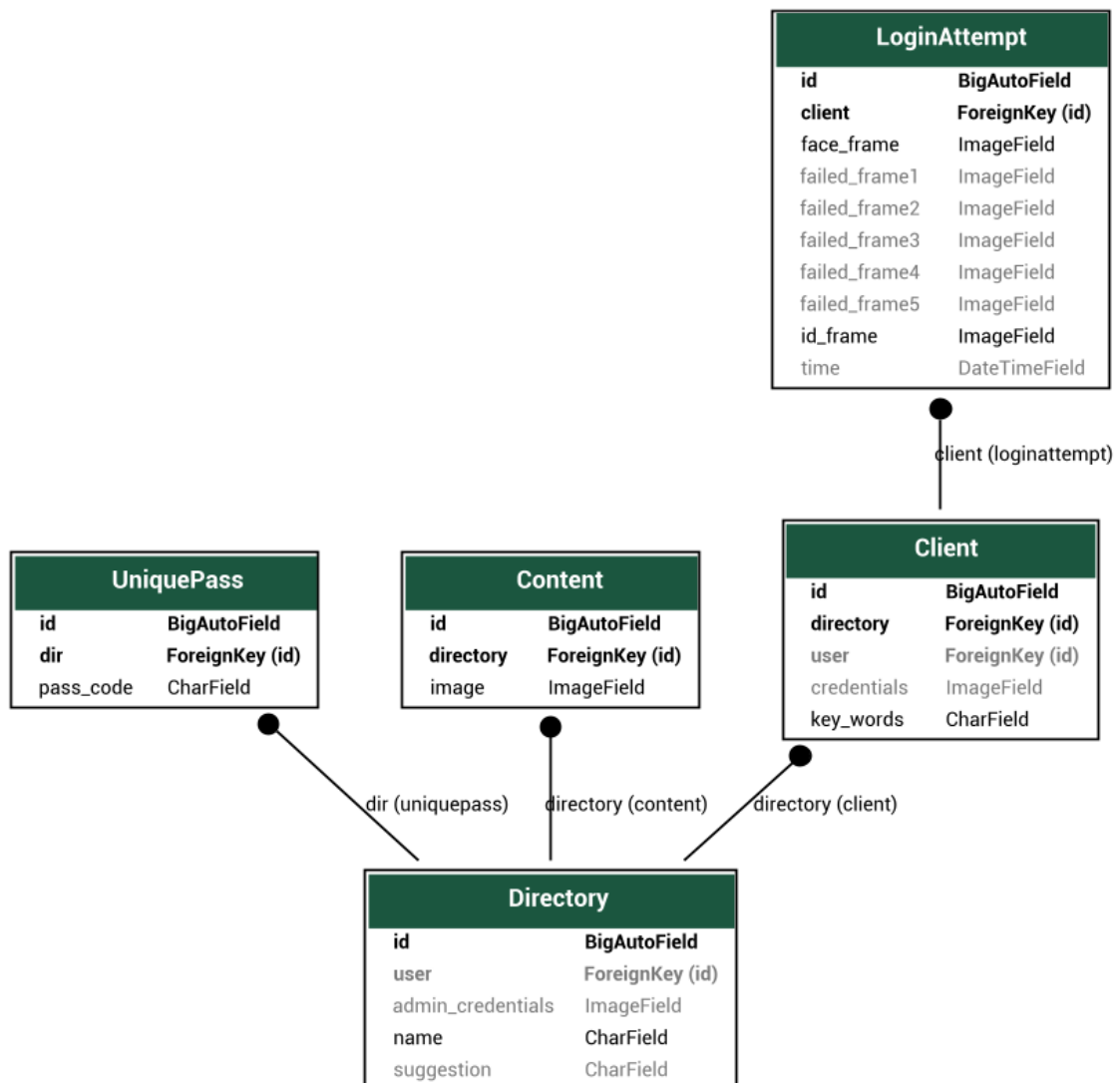


Figura 16. Estructura dels models i els seus camps

Aquesta imatge ha sigut generada directament (gràcies a alguns paquets addicionals de Django)[8] dels models definits a la aplicació. Tot i que la nomenclatura no és la convencional per a un esquema de taules d'una base de dades es pot entendre l'estructura general i la relació entre les taules.

6 Implementació

6.1 Desenvolupament de les funcions de biometria

La primera part que vaig fer correspon a les funcions de Backend de reconeixement facial, processament d'imatges i reconeixement de lletres.

Per implementar la solució prevista i provar el seu funcionament abans de fer la app, vaig implementar un programa en el llenguatge de programació Python. Aquest permet fer una prova de totes les funcions així com poder comprovar directament els resultats produïts. Aquesta fase em va anar molt be per provar les funcions de forma aïllada.

6.1.1 Autenticació del document acreditatiu

La primera part de la prova d'identitat obre una finestra que mostra a l'usuari la imatge que s'està rebent mitjançant la càmera predeterminada configurada al sistema. En aquesta finestra hi ha una regió marcada amb les dimensions aproximades d'una targeta d'identificació típica. L'usuari hauria de procedir a col·locar la targeta al rectangle marcat. A sota de la zona designada hi ha un indicador del nivell de coincidència de l'ID mostrat i de la imatge real. Quan el nivell de confiança arriba al 100%, s'emmagatzema el fotograma clau amb la màxima coincidència.

Per identificar la imatge que coincideix s'utilitza l'algorisme SIFT⁶, ja que permet el reconeixement d'imatges d'objectes independentment de l'orientació, la perspectiva i altres transformacions morfològiques. Aquest algorisme funciona extraient els principals punts clau de la imatge, com ara regions fronteres, lletres i altres punts d'interès. Després d'això, intenta trobar punts clau de la segona imatge que es distribueixen d'una manera equivalent, respecte els uns als altres.



Figura 17. Exemple d'extracció de punts clau de la targeta

La gran avantatge de l'algorisme SIFT és la seva robustesa davant de moltes de les transformacions amb les que ens podríem trobar al integrar aquesta part dins de la aplicació. L'usuari pot proporcionar una molt bona imatge de gran qualitat però que després la seva càmera no sigui capaç de mostrar tots els detalls del carnet. Si utilitzéssim un enfoc tradicional seria impossible determinar quines imatges son equivalents i quines no.

⁶ SIFT: Scale Invariant Feature Transform: Transformació de característica a escala invariable

La característica més important de l'algorisme i el que ha fet que el triés per aquesta part és el fet de que no només analitza els punts clau en la imatge, sinó que emmagatzema la relació entre aquests punts, la seva distància relativa en l'espai, per a després poder detectar imatges dels mateixos objectes des de perspectives diferents.

Aquest algorisme és tant potent que en certes ocasions, mentre feia proves de funcionament, sense adonar-me he col·locat el document del revés i tot i això el programa ha donat una coincidència positiva a les imatges.

Amb aquest mètode no cal tenir imatges similars en el sentit estricte de la paraula, perquè l'algorisme s'adapta a diferents condicions d'il·luminació, orientació, rotació i escalada de mida. Per tant, les operacions de preprocessament necessàries són mínimes, una mica de difuminació (amb la funció `GaussianBlur`) i una transformació de l'espai de colors, passant de BGR a *Grayscale* (escala de grisos), reduint de 3 a 1 cada valor de cada píxel i augmentant així la velocitat de processament de l'algorisme.

Tot i això, el model és complex, en algunes ocasions, tot i que les imatges es trobin de forma molt similar una respecte de l'altra en quant a posició, li pot costar de trobar coincidències si la il·luminació canvia molt

6.1.2 Reconeixement facial

Quan comença la següent fase s'obre una nova finestra que mostra l'entrada de vídeo, en aquest cas tota la imatge es processa per trobar tots els rostres presents. El programa pot retornar "cap rostre trobat" o "rostre desconegut", segons quin és l'error al "frame" processat. Finalment si troba un rostre coincident, enregistra el fotograma concret.

Per aquesta part, el reconeixement facial no es va programar des de zero, sinó que es va utilitzar la biblioteca `face_recognition`. Aquesta biblioteca ofereix funcions per a trobar la ubicació dels rostres, codificar-los i comparar-los. Aquest paquet utilitza un model entrenat per extreure una matriu de 128 dimensions de característiques facials. La funció de concordança de cara calcula la distància entre cadascun dels codificacions i basada en el grau d'exigència, produeix un valor booleà corresponent al resultat.

Aquest paquet és el millor que vaig trobar i s'adapta a les necessitats d'aquest problema en concret per dues raons clau.

La primera característica que em va cridar la atenció és que per una part ha sigut entrenat per reconèixer rostres de forma eficient, es a dir, no fa una simple comparació d'imatges sinó que es capaç de extreure la informació necessària per identificar què és un rostre i transformar el que era un tipus d'informació abstracte en uns paràmetres matemàtics per tal de poder fer operacions amb ells. En aquest cas calcular la distància entre els descriptors de les imatges i obtenir de forma matemàtica i quantificada com de diferents són els rostres.

La segona raó per la que vaig decidir utilitzar aquest algorisme és perquè permet extreure d'una sola imatge de referència suficient informació com per poder reconèixer una gran gamma de rostres, corresponents al subespai de totes les petites variacions que es troben dins de la distància mínima establerta. En altres paraules, des d'un punt de partida determinat que és el rostre original es reconeixeran tots els rostres iguals però amb petites variacions.

Traduint aquesta segona característica a un benefici que ens interessa ens trobem que resol un dels majors temors que tenia: què passa quan una persona canvia respecte del seu document? Gràcies a aquesta segona característica aquesta preocupació esdevé trivial. Ja

sigui perquè la persona es fa més gran i té més arrugues, es canvia de pentinat o canvia la quantitat de barba que té, mentre es mantingui en el subespai de rostres permessos segueix sent detectat.

Llavors existeix un altre dubte, com de gran és aquest subespai de rostres per tal de reconèixer tots aquests canvis en una persona sense reconèixer també a altres persones? El fet és que els atributs que defineixen una persona no són els que més distingim els humans sinó que són altres característiques invariants. Per un humà un canvi estètic molt gran el pot confondre, però aquest model ha estat entrenat per distingir entre les característiques variants i les invariants (que són les que identifiquen a la persona).

Així doncs, la matriu de 128 dimensions que crea aquest model correspon a característiques desconegudes o, més aviat, que són difícils de descriure per a un humà, això es deu al fet que utilitza un model ja entrenat que ha estat format per extreure les funcions més rellevants per optimitzar la quantitat de dades útils i minimitzar la informació redundant. Aquest model és precisament potent per aquest motiu, ja que els 128 paràmetres que es tenen en compte no han estat triats per el programador, sinó que són paràmetres que relacionen diferents atributs per distingir de la forma més estricta i eficient diferents rostres, dividint tot el subespai de rostres possibles de forma eficient.

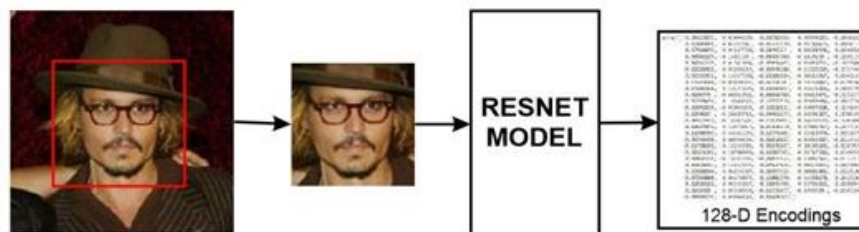


Figura 18. Reconeixement de la cara pas a pas i matriu 128-dimensional de característiques facials.

A més, quan el programa implementat detecta una cara que no coincideix amb la referència, emmagatzema aquesta imatge també dins dels resultats. Això es fa com una capa de seguretat addicional. En alguns casos, aquesta podria ser una prova d'una altra persona que intenta accedir al servei. El nombre de rostres fallits està limitat a 5 per no produir una quantitat massiva de dades redundants.

6.1.3 Reconeixement de caràcters

La última funció de processament d'imatges correspon a la detecció de caràcters. Per aquesta part el paquet utilitzat es EasyOCR. Aquest paquet és una de les solucions més senzilles d'utilitzar però més completes i que millor s'adapta al problema que tenim.

EasyOCR utilitza una Xarxa Neuronal Convolutiva Recursiva, una de les arquitectures més potents sempre i quan estigui ben entrenada. Aquesta arquitectura pertany al que es coneix com a Deep Learning, on tenim una gran quantitat de capes dins de la xarxa neuronal de les que no podem extreure ben bé que contemplen.

Entrant més en detall en el funcionament d'aquest model, cal destacar la seva adaptabilitat, gràcies a una llarga cadena de funcions que s'executen en successió el resultat que s'obté en utilitzar les funcions d'aquest paquet són sorprenents.

Per començar, en el nostre cas no li proporcionem la font de la lletra que ha de buscar, per tant cal que identifiqui què és una lletra i què no ho és. Per realitzar aquesta tasca primer

cal adaptar la imatge ja que pot variar molt segons la il·luminació o la saturació. Cal trobar quin és el valor del Threshold⁷ correcte per tal de separar els caràcters del fons de la imatge, procurant que no es perdin parts del caràcters ni que el fons s'interposi entre les lletres.

Un cop obtingut el threshold adequat, es prova de reconèixer el màxim de cossos aïllats que podrien ser lletres, s'afegeixen dins d'un conjunt i aquest es processa per tal de comparar-lo amb totes les fonts corresponents a l'idioma o idiomes seleccionats.

És en aquesta part on entra en joc la CRNN. Aquesta és la part més complicada però la més interessant des del punt de vista computacional.

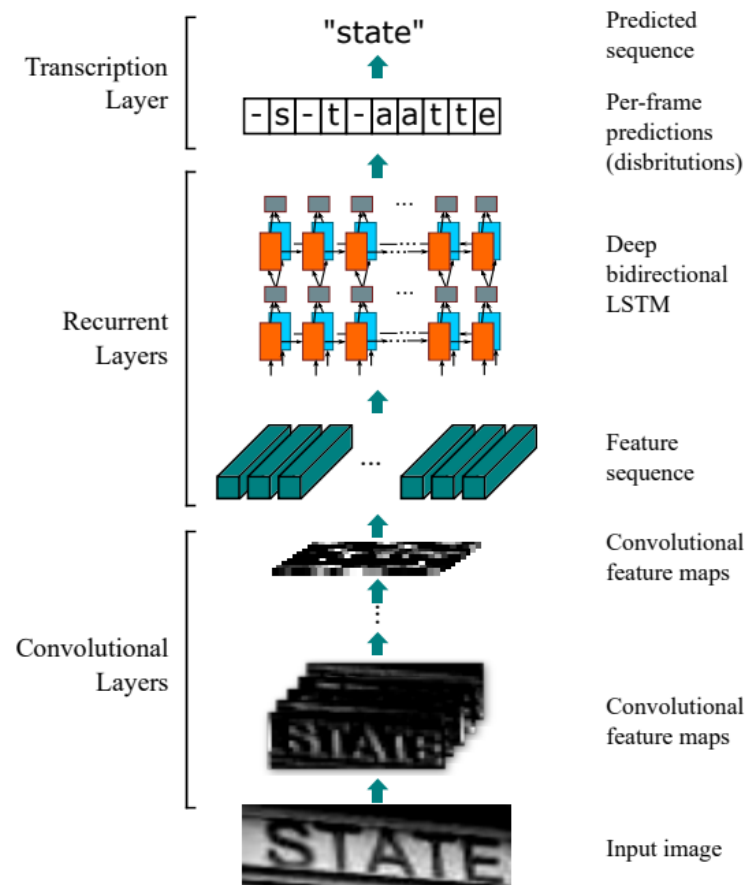


Figura 19. Arquitectura CRNN pertanyent a EasyOCR

Pel que fa al paquet de Python en sí, deixant de banda el model, és molt complet ja que compta amb més de 80 llenguatges de diferents alfabet, això podria anar be per si s'hagués d'adaptar el sistema de reconeixement a una aplicació en un altre idioma, ja que aquesta solució seguiria funcionant.

En aquest punt també cal mencionar que haver realitzat una solució pròpia d'aquest problema ja hagués suposat una part important del projecte si es volgués obtenir uns resultats

⁷ Threshold: Correspon a un valor llindar, en una imatge binària cada punt es trobarà per sobre o per sota i la quantitat de punts en cada regió depèn d'on es situï aquest llindar.

mínimament decents. Cal mencionar també que es va provar de fer des de zero i es va descartar aquesta idea ja que era més important realitzar una bona integració i que totes les eines funcionessin el millor possible.

6.2 Implementació de l'Aplicació Web

De les dues parts principals aquesta és la que personalment m'ha costat més realitzar degut a la meva inexperiència en l'àmbit i en més d'un moment m'he adonat massa dins en el projecte de coses que podria haver fet diferents des del principi.

El gruix d'aquesta part es troba als fitxers de Views on es preparen totes les dades per cada petició i es tracta totes les dades que és reben del FrontEnd. Moltes d'aquestes funcionen de forma similar. En el cas de peticions tipus GET, s'obtenen els objectes necessaris mitjançant una Query interna als models de la aplicació, a la qual es poden aplicar filtres en cas de ser necessaris utilitzant els IDs rebuts a la petició. Finalment aquestes acaben fent un render d'algun dels templates juntament amb un argument que conté tot el context necessari per emplenar la pàgina.

En el cas de les peticions de tipus POST, s'obtenen les dades rebudes (fitxers, imatges, texte...), es comprova la seva validesa i normalment s'emmagatzemen en algun dels objectes, ja sigui existent o nou.

```
def add_admin(request, dir_id):
    dir = Directory.objects.get(id=dir_id)
    if request.method == "GET":
        return render(request, "idverification/add_admin.html",
                      {"dir": dir})
    if request.method == "POST":
        if dir.admin_credentials:
            dir.admin_credentials.delete()
        dir.admin_credentials = request.FILES.get('credentials')
        dir.save()

        if not check_face(dir.admin_credentials.path):
            messages.success(request, ("No s'a trobat cap rostre a la
                                      teva imatge, torna-ho a intentar."))
            return render(request, "idverification/add_admin.html",
                          {"dir": dir})

        user = User(username=dir.name)
        user.save()
        dir.user = user
        dir.save()
        return redirect("idverification:identity_check",
                        dir_id=dir.id,
                        credentials_path=dir.admin_credentials.path,
                        user_id=dir.user.id)
```

Aquest fragment correspon a la vista de la funció per afegir un Administrador, s'executa just després de crear un Directori i és una mostra de l'estil de les vistes realitzades per cada part de la aplicació.

La aplicació compta amb 10 vistes per la part del repositori i la seva administració i 14 vistes més per tota la part de la verificació de la identitat, la creació i l'accés als directoris.

Cal destacar la implementació de la part en la que s'ha de visualitzar la imatge de la càmera dins del navegador, ja que és la més peculiar, complexa e interessant. Per realitzar aquest efecte, aprofitem el "Django Template Language" al nostre favor. Aquest llenguatge

facilita la obtenció de certs elements de la aplicació des del nostre template així com fer ús del context del que parlava quan deia que fèiem el render de un template.

Aquest llenguatge ve marcat per els símbols `{{ }}` o `{% %}` en el cas d'utilitzar alguna etiqueta. En el nostre cas, el tag o etiqueta que ens interessa és `{% url ... %}`, aquest permet obtenir contingut mitjançant una petició a una url definida. Per tant si definim una url i una vista a aquesta, podem utilitzar aquesta etiqueta dins de l'element `src` que pertany a l'objecte `` d'HTML. Això permet que una vista dedicada proveeixi la font d'imatge al template en temps real.

En aquest punt vaig trobar un dels majors reptes del desenvolupament i és que un cop realitzada aquesta petició, des de la vista que proporciona la imatge no es pot canviar de pàgina de cap manera, encara que a la lògica del processament de la imatge s'hagi aconseguit tota la informació necessària. Les vistes són les encarregades de fer el render de la nova pàgina però si el resultat d'aquesta va a parar dins del camp "src" de una imatge, l'únic que s'aconsegueix és que es perdi la imatge, però el template no canvia.

La única solució possible és mostrar dins de la imatge un missatge de confirmació a l'usuari per tal de que sigui aquest el que premi un botó que ara sí farà una petició a una vista i el resultat sí que es podrà processar com a un render de una nova pàgina. Tot i això cal que aquesta vista sàpiga si s'ha completat la verificació.

Aquí és on vaig trobar el segon major obstacle en el desenvolupament. Fins a aquest punt, tota la informació necessària passava per paràmetre a les vistes i amb el context als templates, però ara tenia una vista que necessitava informació només accessible a la vista que es troba encastada al `src` i que no té manera de comunicar-se.

Les solucions que vaig intentar van ser utilitzar el camp `request.session`, que és propi de Django y que conté variables que es mantenen fins que no acabi la sessió, com ara l'usuari autenticat, el path anterior (per quan l'usuari prem el botó de retornar del navegador), entre altres camps, utilitzar paquets com "pickle" i "marshall" i cap funcionava per al meu propòsit.

Llavors vaig haver de pensar fora de la capsa i vaig adonar-me de que estava massa centrat en seguir el funcionament al que estava acostumat amb totes les altres vistes i que havia obviat una solució molt més senzilla. La solució al meu dilema va ser utilitzar variables globals que persistien entre trucades de diferents vistes i em permetien compartir la informació que necessitava.

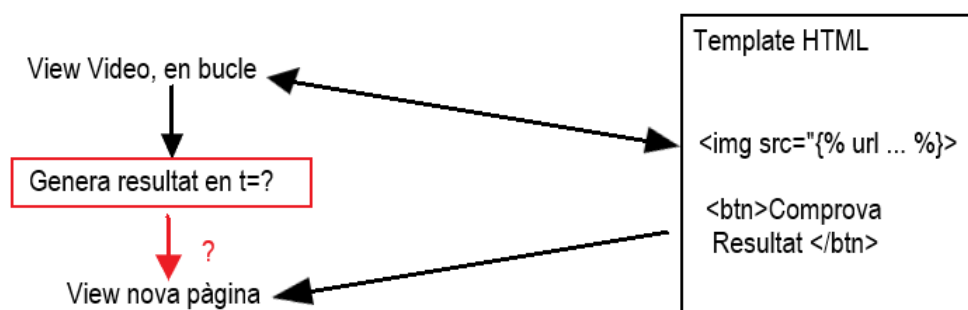


Figura 20. Problema en la comunicació entre vistes

7 Avaluació.

7.1 Proves de les funcions biomètriques

Per provar la robustesa del programa i la seva adaptabilitat a diferents imatges d'entrada, vaig crear un conjunt d'imatges de prova. Vaig provar amb 3 tipus diferents de targetes d'identificació meves i diverses cares de diverses persones diferents. A més, algunes de les imatges es van modificar per provar si el programa era sensible a transformacions com ara remodelació, reorientació, canvi d'espai de colors i difuminació de la imatge.

Els resultats de les proves són difícils d'expressar de manera quantitativa a causa del fet que hi ha un llindar no definit on la qualitat de la imatge de referència és massa dolenta o les diferències amb les condicions reals són massa grans per a l'algorisme SIFT per detectar similituds.

En aquest moment, hi ha un compromís que s'ha d'abordar entre reduir la rigidesa dels requeriments i arriscar-se a obtenir massa falsos positius o augmentar-la i arriscar-se a no obtenir coincidències tot i ser la persona adequada i les targetes d'identificació són les mateixes.

Tot i que l'objectiu d'aquest projecte no era crear un sistema de seguretat irrompible tant com proporcionar una prova fiable d'autenticitat en un moment determinat, el programa resultant està ben adaptat a una sèrie de situacions diferents.

Les situacions que s'han provat són rotacions (esquerra i dreta), cap per avall, blanc i negre, una rotació no múltiple de 90 graus i imatges borroses. Tots aquests casos s'han provat aplicant transformacions en imatges de prova i comprovant manualment si el programa encara reconeixia la identitat de l'usuari. Tots els resultats han estat positius per a totes les diferents transformacions.

La part del reconeixement facial ha donat molt bons resultats fins i tot amb targetes d'identificació que tenien imatges que eren anys més joves que l'edat real de la persona. Per aquest motiu, s'ha decidit que el nivell d'estrictament seria una mica superior a la mitjana, de 6 a 5,5 de distància màxima. Les proves s'han refet amb aquest petit augment, donant lloc de nou a un reconeixement positiu de les cares.



Figura 21. Exemple d'Imatge d'entrada per test de rotació

Durant les proves, també hi va haver variació dels resultats en funció de molt de les condicions de llum de l'habitació en què es van fer les proves. Les habitacions amb llum natural van funcionar molt millor que les habitacions amb llum artificial. Aquesta qüestió seria fatal en una situació regular, però es veu mitigada per l'ús de l'espectre de colors grisos.

Tot i això, és important esmentar que en una situació de cas real, canviar la il·luminació de les habitacions podria ajudar amb els casos en què la qualitat tant de la referència com de la imatge en directe sigui de menor qualitat.

Finalment, en el cas de la detecció de caràcters, el resultat depèn molt de la qualitat de la imatge i dels reflexos que puguin haver-hi sobre el document, si una lletra falla, el document ja no és vàlid com a credencial pel client per tal cal anar amb compte. Tot i això, imatges preses amb mòbils de gama baixa com les que he utilitzat jo, segueixen funcionant si es prenen amb cura.

No és d'estranyar que les funcions utilitzades, els models han estat entrenats i perfeccionats amb màquines molt potents i amb una gran quantitat de dades diverses.

7.2 Avaluació de la pàgina Web

Pel que fa a la avaluació de la pàgina web, he provat diversos casos dins de la pàgina web, totes les funcions creades amb paràmetres diferents per tal de veure el resultat i la resposta del navegador i l'aplicació davant d'aquests resultats. Tot i això, una gran part dels errors típics de les pàgines web es donen degut a errors al servidor, i en el meu cas, la aplicació no s'ha posat en marxa per tant aquest errors no els he contemplat.

He tractat tota mena d'errors durant el desenvolupament de la pàgina, he trobat errors amb el hardware ja sigui la càmera o fins i tot el processador de gràfics al executar la funció de extreure lletres del document. Aquests errors provenien tant de codi com de configuració del meu propi sistema i els he arreglat en la mesura del possible.

Per fer una bona validació del sistema he considerat que calia comprovar tots els camins, botons i accions disponibles al màxim de navegadors possibles. En el meu cas, s'han fer proves a Google Chrome, Mozilla Firefox i Microsoft Edge.

Dins del que seria el tractament d'errors, he intentat que quan aquest apareguin, siguin el més explicatius i el menys molestos possibles. En alguns casos com quan falla la càmera o quan els arxius no es poden desar correctament, es mostra un missatge a la part superior de la pantalla en color groc i sempre amb un petit botó de tancar.

Un dels errors més importants que he tractat és quan un usuari maliciós vol accedir a una pàgina a la qual no té accés. En aquest cas ho podria intentar introduint la URL de la pàgina del administrador per accedir i modificar-la. A la meva aplicació he fet que per accedir a les pàgines d'administrador s'hagi d'estar registrat i identificat com a tal, sinó la pàgina queda totalment bloquejada i l'usuari només pot tornar enrere.

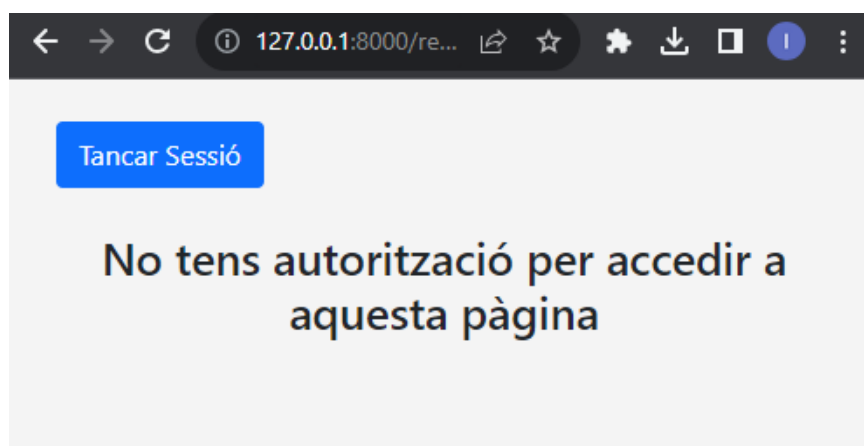


Figura 22. Missatge d'error per intent de traspàs d'identificació

Un cop solucionat i com a forma de validació, he realitzat el que es coneix com a “happy path execution”, que consisteix en realitzar i completar tota la execució de totes les funcionalitats de la web i confirmar-ne el resultat.

Per la meua web aquesta prova consisteix en:

1. Crear un Directori.
2. Afegir nom del Directori (i opcionalment suggeriment de format).
3. Afegir credencials d'Administrador.
4. Comprovar Document proporcionat via càmera.
5. Comprovar rostre de l'Administrador via càmera.
6. Fer log in del Administrador i accedir a la interfície de l'Administrador.
7. Afegir una o més imatges.
8. Eliminar una o més imatges.
9. Comprovar llista de Clients (buida).
10. Afegir Clients manualment i comprovar resultat.
11. Afegir Clients per csv i comprovar resultats.
12. Eliminar clients i comprovar el resultat.
13. Actualitzar el suggeriment de format i comprovar el resultat
14. Comprovar historial d'accessos (buit).
15. Tancar sessió.
16. Cercar Directori.
17. Comprovar que el Directori creat existeix.
18. Fer accés de Client al directori creat.
19. Comprovar el suggeriment de format.
20. Afegir paraules clau del Client.
21. Afegir credencials del Client.
22. Comprovar Document proporcionat via càmera.
23. Comprovar rostre del Client via càmera.
24. Accedir al Directori i veure les imatges que conté.
25. Tancar sessió.
26. Cercar Directori.
27. Fer accés d'Administrador al directori creat.
28. Comprovar Document proporcionat via càmera.
29. Comprovar rostre de l'Administrador via càmera.
30. Fer log in del Administrador i accedir a la interfície de l'Administrador.
31. Comprovar l'historial d'accessos.
32. Veure en detall l'accés del Client.
33. Comprovar les imatges obtingudes al log in del Client.
34. Tornar a la llista de Clients.
35. Crear un codi d'accés únic.
36. Tancar sessió.
37. Cercar Directori.
38. Fer accés de Client al Directori creat.
39. Utilitzar el codi d'accés únic per accedir al Directori.

Aquest procés l'he repetit diverses vegades amb diferents paràmetres per confirmar-ne el funcionament. A continuació detallo les alternatives d'execució o errors que es poden donar durant la execució.

2a. El directori ja existeix: No es poden crear dos directoris amb el mateix nom, per tant es mostra un altre com la pàgina d'afegir el nom juntament amb un missatge de l'error.

3a i 21a. Els credencials no contenen cap rostre, no contenen les paraules clau (en el cas del client) o error inesperat processant la imatge: Es torna a demanar una nova imatge i es mostra quin ha sigut l'error.

4a, 5a, 23a, 24a, 28a, 29a. La càmera no es pot activar o succeeix qualsevol error inesperat amb el processament de les imatges: En aquest cas, com que els errors es produeixen en una vista aïllada com s'ha explicat a la implementació anteriorment, no es pot controlar aquest error. La conseqüència per tant tampoc és greu, simplement no es mostra la càmera o es queda congelada, en aquests casos cal tancar la sessió i tronar a iniciar el procés.

4b, 5b, 23b, 24b, 28b, 29b. L'usuari prem el botó de continuar massa aviat, abans de que s'hagi completat el processament: En aquest cas es retorna a la mateixa pàgina i es mostra un missatge d'error informant de com s'ha de procedir i donant un consell per accelerar el procés.

7a. La imatge afegida no és vàlida: En aquest cas, el formulari que s'envia no és vàlid i la imatge no s'afegeix al directori. Aquest error no pot ocórrer en funcionament normal ja que quan s'obre l'explorador de fitxers, només apareixen fitxer amb extensió d'imatge.

11a. El fitxer no es vàlid: De forma similar al cas anterior, només s'afegiran els clients vàlids, si es produeix un error inesperat no s'afegiran.

20a. El client introdueix unes paraules clau que no corresponen a cap client: Es torna a demanar les paraules clau i es mostra un missatge informant del error.

Pel que fa al testeig, per la meua part no he trobat cap error més dels mencionats anteriorment, en el cas de rebre algun error en execució han estat tots investigats i arreglats.

8 Conclusions.

Considero que he completat l'objectiu de crear una solució prou robusta i que s'adapta correctament als objectius marcats. La part d'autenticació dels usuaris i el programa creat compleix els requisits originals. La feina feta és suficient per considerar les demandes originals finalitzades, tot i que el projecte encara té marge per millorar.

La majoria de les millores possibles serien mitjançant l'especialització del codi per adaptar-se a determinats tipus de documents, formar un model per reconèixer només aquests tipus i les seves característiques particulars. Seria interessant explorar aquesta possibilitat tot i que en el cas d'aquest projecte era crear un programa que s'adapti a tot tipus de document per poder ser utilitzat per tothom.

Aquest projecte també m'ha posat davant de problemes comuns relacionats amb programes de visió per computador. Durant el desenvolupament he hagut d'arreglar in comptables errors dels quals no entenia el motiu (alguns dels quals encara no entenc del tot), però al final he sigut capaç d'arreglar-los o trobar una solució alternativa.

Dins de les habilitats que considero que he millorat es troba el fet de atrevir-me a començar un projecte des de 0 jo sol, ja que m'he adonat que durant la carrera, sempre he tingut un paper molt més de dissenyador i de programador a petita escala, en el sentit de no treballar les parts més relacionades amb la configuració, instal·lacions, l'entorn que s'utilitza, el hardware al que s'accedeix...

Tot i que la meua menció és en Computació, em sento orgullós del projecte de Software que he dissenyat i produït i considero molt útil per al meu desenvolupament com a programador aquest complement de la meua formació que penso que em faltava.

Cal mencionar que durant el desenvolupament del projecte he sentit que hi havia maneres de millorar el projecte i alguns cops, m'he adonat massa tard de coses que podria haver fet diferent i que m'haurien estalviat molta feina. Tot i això l'aprenentatge està fet.

Aquest és el projecte més gran en quantitat de feina, temps i línies de codi que he fet per mi mateix i estic content d'haver-lo pogut tirar endavant tot i les dificultats. Ara tinc més confiança en mi com a programador i estic una mica més preparat per seguir formant-me en aquest món.

Pel que fa a les possibilitats d'expansió del projecte, hi ha encara molta feina per fer. Considero aquest projecte com una primera versió del que pot arribar a ser. Per la part de la aplicació web en sí, podria introduir-se alguna funcionalitat que realment requerís de aquesta seguretat extra que proporciona aquest sistema. La aplicació actual funciona però realment no s'aprecia el valor addicional que hi aporta.

Com a idees d'aplicacions on es podria integrar hi hauria tot el que estigui relacionat amb pagaments, ja que interessa que una compta bancària s'associï amb una sola persona o també tindríem la possibilitat de integrar aquest sistema en alguna aplicació amb un sistema d'assistència obligatòria, en aquest context el sistema creat proporcionaria a l'administrador proves suficients de l'assistència de cada client.

D'altra banda, com a millores en la part de processament, considero que no he arribat a exprimir al màxim el potencial de les eines que he utilitzat. A més a més, per tal d'estalviar recursos al sistema, el processament de les imatges està limitat a 10 fotogrames per segon, donant la sensació a l'hora d'utilitzar el programa de que va lent o que es talla molt.

Aquesta és una de les coses que menys m'agrada del resultat final ja que tot i que les funcions de reconeixement son molt ràpides i eficients, dona la sensació de que el programa

va tallat, però és una decisió que vaig haver de prendre degut a que no dispo de d'una màquina suficientment potent com per incrementar la quantitat de imatges processades per segon.

Tornant al tema de les possibles expansions, una idea que es va descartar durant la fase de disseny, i que s'ha mencionat de forma breu anteriorment, seria la especialització del programa per tal de crear un model que reconegués certs tipus de documents en concret.

Un exemple de l'esmentat seria entrenar un model per reconèixer DNIs espanyols, carnets de conduir espanyols i carnets universitaris de la URV. Cadascun amb les seves característiques i camps propis. Això donaria a l'administrador encara més possibilitats de restricció, podent seleccionar el tipus de document requerit, a l'hora de afegir les credencials es comprovaria també el tipus de document e inclús es podria fer una verificació de certs camps que es considerin rellevants,

Per últim, considero que existeixen algunes funcionalitats imprescindibles que s'haurien d'afegir abans de plantejar-se posar la aplicació en marxa. Caldria una millor gestió de l'emmagatzematge, més funcions per modificar les dades existents al sistema, per exemple eliminar un directori completament o poder modificar les credencials de l'administrador entre altres.

Aquestes funcions no s'han incorporat en aquesta versió ja que no son imprescindibles per comprovar el funcionament del sistema que es proposava.

9 Recursos utilitzats

- [1] Opcions de reconeixement i verificació d'identificació al mercat <https://seon.io/resources/comparisons/identity-verification-software-tools/>
- [2] Projecte de reconeixement d'identificació espanyol <https://github.com/Suaro/DNIDetector>
- [3] S. Chintalapati i M. V. Raghunadh, "Sistema de gestió d'assistència automatitzada basat en algorismes de reconeixement de cares", Conferència internacional IEEE 2013 sobre investigació computacional i informàtica, Enathi, Índia, 2013, pp. 1-5, doi: 10.1109 / ICCIC.2013.6724266.
- [4] Comparació de l'algorisme SIFT i les seves variants <https://kd.nsf.gov.cn/paperDownload/1000006777694.pdf>
- [5] Llocs 2D-Recognició an Identificació Utilitzant el mètode de descriptors HOG, Vyacheslav Voronov, Vladimir Strelnikov, Liliya Voronova, Artyom Trunov, Andrey Vovik, Universitat Tècnica de Comunicacions i Informàtica de Moscou, Moscou, Rússia <https://fruct.org/publications/volume-24/acm24/files/Vor.pdf>
- [6] Explicació del procés de reconeixement facial <https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cfc121d78>
- [7] Documentació de Django <https://docs.djangoproject.com/en/4.2/>
- [8] Documentació per extreure les taules dels models en Django https://django-extensions.readthedocs.io/en/latest/graph_models.html
- [9] Documentació de OpenCV https://docs.opencv.org/4.x/d6/d00/tutorial_py_root.html
- [10] Documentació de face_recognition <https://pypi.org/project/face-recognition/>
- [11] Documentació i codi de easyOCR <https://github.com/JaidedAI/EasyOCR>
- [12] Llista de reproducció de exemple d'aplicació Django <https://youtube.com/playlist?list=PLCC34OHNcOtqW9BJmgQPPzUpJ8hl49AGy&si=dq9F7VrmnV6AIbyn>

10 Annexes

Pel que fa a la instal·lació, el codi es accessible en aquest [enllaç](#) des de la plataforma Github, on existeix un repositori remot amb tot el necessari per provar la app a qualsevol dispositiu.

Per realitzar una instal·lació correcta de la aplicació cal clonar el repositori remot al IDE que l'usuari prefereixi, havent creat primer l'entorn que desitgi. Un cop clonat cal executar la comanda següent dins de la terminal, estant situat dins del repositori del projecte.

```
pip install -r requirements.txt
```

Aquesta comanda, si es compta amb una versió prou actualitzada de pip⁸, instal·larà tots els paquets que son necessaris per executar l'aplicació correctament. Aquests paquets corresponen a totes les llibreries externes afegides.

A continuació, caldria modificar el contingut del fitxer `'tfg/settings.py'`, en concret de la línia 81 a la línia 90, ja que contenen la configuració per a la meua pròpia instància de MySQL i no funcionaran per a altres màquines.

Hi ha diferents alternatives per aquest pas, la primera i més senzilla és utilitzar la configuració bàsica de Django que té la seva pròpia base de dades i per la qual no es necessita cap tipus de configuració addicional. Caldria afegir el següent:

```
DATABASES = {
    "default": {
        "ENGINE": "django.db.backends.sqlite3",
        "NAME": "mydatabase",
    }
}
```

També existeix la possibilitat de afegir una configuració adaptada a una base de dades pròpia. Django suporta diversos tipus de bases de dades diferents. Visita aquest [enllaç](#) per saber-ne més.

En principi aquest hauria de ser tot el procés i a partir d'aquí la aplicació hauria d'executar-se correctament tot i que no he fet cap prova amb un altre ordinador per comprovar-ho.

Pel que fa a l'ús de la aplicació, només executant-la ja estaria disponible per visualitzar-se a qualsevol navegador a l'URL <http://127.0.0.1:8000/>, aquesta però donarà un error ja que no hi ha cap configuració per aquest endpoint, la pàgina principal del projecte es troba a la URL <http://127.0.0.1:8000/idverification/home>, a partir d'aquí la funcionalitat és la descrita anteriorment.

Per poder executar la aplicació, els requeriments seria un dispositiu amb intèrpret de Python 3, en el meu cas utilitzava el 3.9, i un dispositiu amb càmera i potència suficient per poder executar la aplicació (no és gaire pesada).

⁸ Pip: Manegador estàndard de dependències i paquets per Python