Desarrollo de software de reconocimiento facial, como segundo factor de autenticación que permite realizar inicio de sesión en un computador y generar un reporte de personas no autorizadas.

Elaborado por:
Javier Alejandro Zárate Suárez – Ingeniería de Sistemas

Tutor de proyecto de grado:
Juan Diego Granada Suarez

Universidad EAN
Facultad de Ingeniería
Proyecto de Integración de Pregrado
Diciembre, 2023
Bogotá, D.C.

Tabla de Contenido

Tabla de Contenido 2
Tabla de Figuras 3
RESUMEN 4
1. INTRODUCCIÓN 5
1.1 Objetivos 6
1.1.1 Objetivo General 6
1.1.2 Objetivos Específicos 6
1.2 Definición del problema 7
1.3 Justificación 9
1.4 Análisis de Requerimientos 10
2. Marco de Referencia 12
2.1 Marco Conceptual 12
2.2 Análisis de restricciones 13
2.2.1 Derechos de Autor 13
2.2.2 Socioculturales 13
2.2.3 Técnicas 13
2.2.4 Legales 14
3. Diseño de solución 15
4. Herramientas y lenguajes de programación de reconocimiento facial 16
4.1 Lenguajes de programación 16
4.2 Paquetes y librerías de reconocimiento facial 16
4.2.1 Tensorflow 2.3.0 17
4.2.2 Numpy 17
4.2.3 OpenCV-Python 17
4.2.4 mtcnn (Multi-Task Cascaded Convolutional Neural Networks) 17
4.2.5 Scikit-learn 17
5. Software de reconocimiento facial 18
5.1 Obtención del programa de uso libre de reconocimiento facial 18
5.2 Configuración y funcionamiento Software de Reconocimiento Facial 18
5.3 Histórico y almacenamiento de imágenes de usuarios no autorizados 25
6. Análisis de costos 27
7. Plan de implementación 35

8. Conclusiones 36
9. Referencias 37

Tabla de Figuras

Figura 1 – Donde es más vulnerable nuestra información 7
Figura 2 – Archivo “train_v2.py” - 1 19
Figura 3 – Archivo “train_v2.py” - 2 19
Figura 4 – Archivo “detect.py” - 1 20
Figura 5 – Archivo “detect.py” - 2 21
Figura 6– Archivo “detect.py” - 3 21
Figura 7 – Archivo “detect.py” - 4 22
Figura 8 – Archivo “ProyAccesoJavi.py” - 1 22
Figura 9 – Archivo “ProyAccesoJavi.py” - 2 23
Figura 10 – Reconocimiento facial persona con accesos - 1 24
Figura 11 – Reconocimiento facial persona con accesos - 2 24
Figura 12 – Archivo “ProyAccesoJavi.py” - 3 25
Figura 13 – Reconocimiento facial Desconocido 26
Figura 14 – Reconocimiento facial Reporte de Desconocido 26
Figura 15 – Cuanto vale mi app - 1 28
Figura 16 – Cuanto vale mi app - 2 29
Figura 17 – Cuanto vale mi app - 3 29
Figura 18 – Cuanto vale mi app - 4 30
Figura 19 – Cuanto vale mi app – 5 30
Figura 20 – Cuanto vale mi app – 6 31
Figura 21 – Cuanto vale mi app - 7 31
Figura 22 – Cuanto vale mi app - 8 32
Figura 23 – Cuanto vale mi app - 9 32
Figura 24 – Cuanto vale mi app - 10 33
Figura 25 – Cuanto vale mi app - 11 33

RESUMEN

En el presente trabajo de grado se desarrolló un prototipo de reconocimiento facial como segundo factor de autenticación, por medio del cual un empleado podrá realizar inicio de sesión en su computador, y si un usuario sin autorización intenta acceder al sistema se le bloqueara el acceso y se guardara su imagen con fecha y hora en la que el usuario no autorizado intento acceder, con el fin de tener el registro de personas que intentaron acceder al computador y no estaban registradas como usuarios autorizados.

Este prototipo se desarrolló con modelos de machine learning y visión artificial, utilizando el lenguaje de programación de Python y algunas librerías existentes, desarrolladas en este lenguaje por otros autores.

Palabras clave: Factor de Autenticación, Seguridad, Reconocimiento Facial, Machine Learning, Visión Artificial.

1. INTRODUCCIÓN

Con el transcurrir del tiempo como lo menciona (Comisión Económica para América Latina y el Caribe (CEPAL), 2021) , el hombre ha investigado, experimentado y desarrollado nuevas tecnologías y aplicativos que le han ayudado a realizar sus tareas de una manera más eficiente, en donde se ha utilizado y almacenado información confidencial y no confidencial.

Por todo esto la seguridad de estos sistemas y de su información hoy en día son un factor muy importante para tener en cuenta, tal como lo señala (Avenía, 2017), existen muchas amenazas que pueden llegar a comprometer los sistemas informáticos, y el principal de ellos es el acceso a personas no autorizadas que logren ingresar a este.

Hoy en día el uso de contraseñas es la principal herramienta para permitir el acceso a un usuario, pero como lo explica (Abhishek et al., 2013), el uso de múltiples factores de autenticación, permiten generar una mayor seguridad, al garantizar que la persona que está intentando acceder al sistema es realmente la persona que tiene acceso a la información, dado que para acceder, el usuario debe ingresar la contraseña, y además debe validar por un segundo medio que es la persona asociada a dicho a usuario que intenta iniciar sesión en el sistema, un ejemplo de esto es el ingreso de un código de seguridad enviado al número de celular registrado con anterioridad.

1.1 Objetivos

1.1.1 Objetivo General

Desarrollar software de reconocimiento facial, por medio de machine learning y visión artificial, el cual le permita realizar inicio de sesión únicamente a los usuarios autorizados y almacenar la foto de la persona no autorizada que intente acceder al sistema.

1.1.2 Objetivos Específicos

- Identificar las mejores herramientas y lenguaje de programación que permitan desarrollar el software de visión artificial de reconocimiento facial.

- Construir el aplicativo que permita realizar el reconocimiento facial, en el que por medio de machine learning se le enseñe al algoritmo quienes si son usuarios autorizados y quienes no.

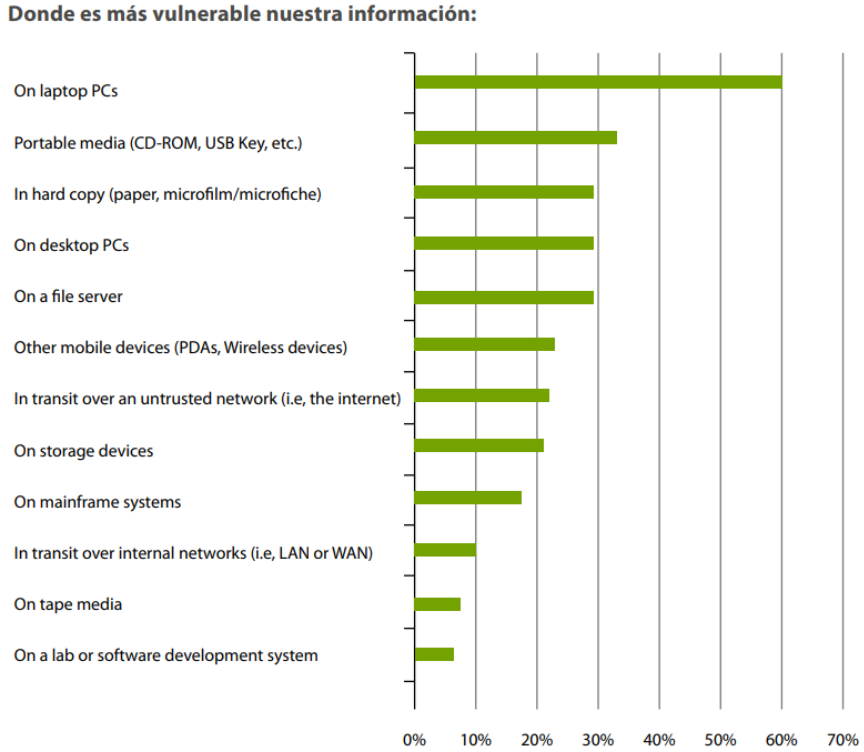
- Incluir en el software la funcionalidad de que si algún usuario no autorizado intenta iniciar sesión, no se lo permita y almacene la foto de esa persona, con la fecha y hora en la que intento acceder.

1.2 Definición del problema

Este tema de seguridad de la información es uno de los temas más importantes para todas las compañías del mundo, ya que de esto depende su prestigio y la confianza de sus clientes, ya que, al existir un acceso no autorizado o divulgarse información confidencial, como lo sustenta (IT User, 2023) , las empresas no solo podrán llegar a tener demandas y perdida de dinero, sino que al perder su prestigio, sus clientes no invertirán o no les confiaran su información, lo cual los puede llevar a quedar en banca rota.

Como se observa en la figura 1, los dispositivos con mayor vulnerabilidad, son los computadores portátiles, que como lo explica el autor, esto se debe principalmente a que son “herramientas que siempre están conectadas a una Red o en un entorno laboral donde siempre se buscara la manera de poder ingresar a información empresarial o institucional con algún fin, más del 90% del robo de información son por errores humanos atados a la inconciencia de la persona” (Avenía, 2017), relacionado a esto se puede señalar como un ejemplo de error humano, cuando algún empleado se conecta desde algún lugar público, y al colocar su contraseña una persona, o una cámara por detrás logré observar la contraseña que está ingresando en su computador, las personas que lo vean o que tengan está contraseña podrán acceder a su correo y plataformas de la empresa y hacer un uso inadecuado de la información.

Figura 1 – Donde es más vulnerable nuestra información



Fuente: (Avenía, 2017)

Por todo lo anterior, en la actualidad muchas compañías están implementando el doble factor de autenticación, donde los empleados, no solo deben ingresar la clave, sino que también se le pide registrar su número celular al que se le envía un código de verificación por medio de mensajes de texto o se le pide instalar una aplicación, como por ejemplo “Symantec VIP Access” la cual genera un código dinámico de 6 dígitos, para realizar una segunda validación y ahí si permitir el acceso al usuario autenticado.

1.3 Justificación

Como lo explica (Symantec, 2022), el uso de contraseñas no es suficiente para detener un ataque y el uso de un segundo factor de autenticación es fácil de activar y permite tener una mayor seguridad de la información.

Los segundos factores de autenticación que se han implementado para asegurar que el usuario que intenta acceder al sistema es realmente el empleado con accesos, como el mencionado anteriormente del aplicativo de “Symantec VIP Access” es funcional en esos momentos, pero también es importante considerar que si por alguna razón el empleado olvida, se le daña, o se les descarga el celular, también les va a ser imposible iniciar sesión para trabajar, dado que es un código obligatorio para poder ingresar a su computador o a las plataformas de la empresa.

El otro inconveniente que se ve con la utilización de envió de códigos de confirmación al celular, es que es un paso adicional que requiere tiempo y desgaste de encontrar el celular, esperar a que llegue el código y digitar el código en el computador, sin mencionar el hecho de mirar el celular al que continuamente llegan notificaciones de redes sociales y los empleados pueden distraerse entrando a las redes sociales en sus celulares y perder tiempo de trabajo.

Por todo lo anterior después de digitar la clave, el segundo factor de autenticación más efectivo por seguridad y practicidad es el de reconocimiento facial, que desde que el portátil cuente con cámara, simplemente después de digitar la contraseña automáticamente se activa la cámara y se le da acceso inmediato al usuario, sin desperdiciar tiempo con todos los pasos que requieren otros métodos de autenticación, como el mencionado anteriormente.

Es relevante aclarar que el reconocimiento facial se incluiría como un segundo factor de seguridad después de ingresar la contraseña, por lo tanto es obligatorio cumplir con dos factores de seguridad, en donde luego de digitar la contraseña, automáticamente se activara la cámara y se hará el reconocimiento facial respectivo, pero si por alguna razón el reconocimiento facial no se logra detectar, se tendría la opción de la verificación por medio del envío de códigos al celular, esto con el fin de que si por alguna razón la cámara no funciona, o no se tiene está opción, los empleados podrán cumplir con los 2 filtros de seguridad: la contraseña (que es obligatoria) y el reconocimiento facial o el envío de códigos de autenticación (en su defecto).

1.4 Análisis de Requerimientos

Con el fin de desarrollar el software que permita realizar el reconocimiento de rostros, es indispensable tener un amplio conocimiento en Machine Learning, Visión Artificial, lógica de programación y conocimiento de lenguajes de programación como python y librerías existentes que permitan realizar el reconocimiento de rostros y por su puesto una interfaz o ambiente de desarrollo, como visual studio code.

Para el uso como tal, es muy indispensable que todos los portátiles a los que se les active como segundo factor de autenticación el reconocimiento de rostros, cuenten con una cámara de buena calidad, que por lo general todas las compañías ofrecen a sus empleados.

Otros de los requerimientos técnicos son los siguientes:

Tabla 1 – Requerimiento 01

Número de requisito	REQ01
Nombre de requisito	Componente de Inicio de Sesión debe habilitar la cámara tan pronto se le solicite la contraseña al usuario
Tipo	Requisito
Fuente del requisito	Cuando se muestre el componente de inicio de sesión la cámara debe activarse.
Prioridad del requisito	Alta/Esencial ☐ ☐

Fuente: Creación propia

Tabla 2 – Requerimiento 02

Número de requisito	REQ02
Nombre de requisito	Por medio del software, la cámara del computador deberá tomar foto a la persona y realizar reconocimiento facial
Tipo	Requisito
Fuente del requisito	Tan pronto el usuario ingrese su contraseña y presione la tecla “Enter” o de click en el botón “Iniciar Sesión”, el aplicativo tomara una foto de la persona que está intentando acceder e identificara si realmente es la persona asociada a ese nombre de usuario se está intentando acceder.
Prioridad del requisito	Alta/Esencial ☐ ☐

Fuente: Creación propia

Tabla 3 – Requerimiento 03

Número de requisito	REQ03
Nombre de requisito	En caso de fallar el reconocimiento facial se debe habilitar la segunda opción de segundo factor de autenticación.
Tipo	Requisito
Fuente del requisito	Si por algún motivo no se logra realizar el reconocimiento facial, el software deberá habilitar la opción para ingresar un código de 8 dígitos, el cual se le enviará al celular registrado del usuario.
Prioridad del requisito	Alta/Esencial ☐ ☐

Fuente: Creación propia

Tabla 4 – Requerimiento 04

Número de requisito	REQ04
Nombre de requisito	El servidor web de la compañía debe permitir el almacenamiento de imágenes de rostro asociados a cada usuario
Tipo	Requisito
Fuente del requisito	Es indispensable al momento de crear un usuario, o asignar un computador a los empleados, que cada empleado mediante el mismo computador, se tome 3 fotos de su rostro en diferentes posiciones y las cargue al sistema.
Prioridad del requisito	Alta/Esencial ☐ ☐

Fuente: Creación propia

2. Marco de Referencia

2.1 Marco Conceptual

- Múltiple Factor de Autenticación MFA: Como lo explica (CISA, 2021), el Multi Factor Authetication (MFA), es una capa que permite proteger los datos y aplicaciones en donde se le solicita al usuario que cumpla con una combinación de 2 o más credenciales para verificar la identidad del usuario para su inicio de sesión. El MFA aumenta la seguridad, ya que, si alguna de las credenciales no se cumple, usuarios que no cumplan con el primer requisito, no seguirán al segundo filtro de validación y por consiguiente no tendrán acceso al espacio físico, al equipo de cómputo, a la red o la base de datos.

- Factores Biométricos

  En su paper (Misini & Lajçi, 2022), señala que la Biometría, es todo lo que tiene que ver con el uso de las características del cuerpo humano para identificar a una persona. Y estás se dividen principalmente en fisiológicas y comportamentales.

  Características fisiológicas, tal como lo son: el reconocimiento facial, huellas dactilares, geometría de la mano, reconocimiento del iris o retina y reconocimiento de firma, entre otras.

  Características comportamentales: como, por ejemplo, el reconocimiento de voz, el movimiento del mouse y la velocidad de tipeo en un teclado.

- Reconocimiento Facial: (Parmar & Mehta, 2014) destacan que el reconocimiento facial, es un sistema biométrico utilizado para identificar a una persona, a partir de una imagen digital, de la cual se detecta la cara y se extraen sus características fisiológicas. Lo cual requiere un procesamiento a la imagen, con el fin de eliminar otros objetos aparte del rostro que no son relevantes, e independientemente de la luz u otros factores, se debe garantizar el reconocimiento de la persona.

- Visión Artificial: La visión artificial o en inglés computer visión, como lo explican los profesores de (Stanford, 2017), este es un campo científico que permite extraer información a partir de imágenes digitales. El tipo de información obtenida de una imagen puede aplicarse para realizar identificación, medida de espacios, o realidad aumentada. En donde este entendimiento y procesamiento de imágenes se realiza a partir de la construcción de algoritmos, desarrollados en lenguajes de programación, donde los 3 principales como lo señala (Noema, 2021), son MatLab, C++ y Python.

2.2 Análisis de restricciones

A continuación, se expondrán las principales restricciones a considerar al momento de implementar el software, en el inicio de sesión en todos los sistemas operativos.

2.2.1 Derechos de Autor

El prototipo desarrollado para la sustentación de este proyecto, se realizó mediante el uso de paquetes y librerías existentes desarrolladas en el lenguaje de programación de Python por terceros, que son de uso libre, pero se puede dar el caso de que alguno de estos terceros solicite una remuneración o un permiso adicional por el uso de este código desarrollado por ellos. Por lo tanto, en el caso de que se desee implementar para obtener un beneficio económico, se debe analizar si al momento de ofrecer este software, se utilizara este código ya desarrollado por terceros, o de ser necesario se desarrollara un código exclusivo de reconocimiento facial a la compañía que desee adquirir el aplicativo, o la otra opción que es la mejor, es si las compañías de sistemas operativos ya cuentan con un aplicativo que permite reconocer a los usuarios por reconocimiento facial en móviles, adaptarlo para que este mismo se pueda incluir en el inicio de sesión de un computador y así incorporarle nuestro componente de seguridad que permita utilizar esta funcionalidad de reconocimiento como segundo factor de autenticación y permitir tomar fotos y guardar las evidencias de los usuarios no autorizados que intentaron acceder al sistema.

2.2.2 Socioculturales

La implementación de un factor de autenticación adicional a la contraseña que usualmente utilizan los empleados de las compañías para acceder a sus portátiles, como lo es el reconocimiento facial, puede llegar a tener implicaciones socioculturales, ya que adicional a lo que los empleados realizaban habitualmente para acceder a sus computadores va a cambiar, y tal vez en un inicio al presentarles este segundo factor como medio de protección de la información, lo más probable es que muestren un poco de resistencia al cambio, pero con el uso cotidiano no tendrá una afectación directa, ya que el reconocimiento facial se realiza de una manera automática tan pronto ellos ingresan su contraseña correctamente.

2.2.3 Técnicas

Los empleados se podrán llegar a ver afectados si su cámara no funciona o no se logra realizar el reconocimiento facial del usuario por algún motivo, en este caso el usuario tendrá otra opción, en la cual validará su segundo factor de autenticación por medio de un código de seguridad de 6 dígitos que se le enviará a su número de celular registrado.

2.2.4 Legales

Otra de las restricciones más importantes a tener en cuenta son las legales y el uso de información de identificación personal (PII), que como lo explica IBM, “la PII confidencial no está disponible a nivel público, y la mayoría de las leyes actuales de protección de datos exigen que las organizaciones los protejan mediante cifrado, controlen quién accede a ellos o adopten otras medidas de ciberseguridad.” (IBM, 2022)

Por lo cual, el almacenamiento de datos biométricos que pueden identificar a una persona debe dársele un uso adecuado, ya que estos no deben compartirse con terceros, ni usarse para otro fin diferente al que el empleado ha autorizado a la compañía, que es el de utilizarlos para validar que realmente es el, el usuario que realiza el inicio de sesión en el equipo de cómputo.

3. Diseño de solución

Según (Verdium, 2019), en Julio de 2019 Microsoft presento que para el siguiente lanzamiento de su sistema operativo Windows 10, por medio de su tecnología biométrica de Windows Hello, sus usuarios podrían remplazar sus contraseñas, por métodos de reconocimiento facial para realizar inicio de sesión.

Los problemas que presentan este software de Windows Hello, es que no se utiliza como un segundo factor de autenticación, después de ingresar la contraseña, sino como único factor de autenticación, lo cual no garantiza un segundo filtro para prevenir el acceso a personas que por alguna razón conozcan la contraseña, pero no sean el usuario con permisos.

Otras de las desventajas como lo menciona (Verdium, 2019), son que únicamente se puede activar en dispositivos con sistema de operativo de Microsoft y a partir de Windows 10; por lo tanto, si algún empleado tiene MacBook (de Apple), no le será posible habilitar este sistema de seguridad.

En el caso de Apple, de acuerdo con (Soriano, 2023) aún no se ha incluido, por lo tanto, los usuarios de MacBooks tendrán que esperar y ver si la seguridad biométrica del Face ID se integra algún día en sus dispositivos o no.

Teniendo en cuenta todo lo anterior, la solución planteada en este proyecto es tener el factor de autenticación de reconocimiento facial que pueda ser utilizado para todos los computadores, incluyendo todos los sistemas operativos y utilizándolo como segundo factor de autenticación, luego de que el usuario ingrese correctamente la contraseña.

Adicionalmente este programa al momento de que se ingrese la contraseña correctamente y si por alguna razón se identifica a una persona que no es el usuario con permisos, se guardara una imagen de esta persona, con la información de la fecha y hora en que intento acceder este usuario no autorizado. Adicionalmente, si por algún motivo a algún usuario se le daña la cámara del portátil o el computador no cuenta con cámara, tendrá otra forma de realizar la validación por medio del envío de un código de seguridad de 6 dígitos al celular registrado, e ingresándolo en el computador, con el fin de que el usuario pueda acceder a realizar su trabajo. Vale la pena aclarar que, si el usuario no cumple con los 2 filtros de seguridad, no se le dará acceso al sistema, con el fin de asegurar la protección de los datos.

4. Herramientas y lenguajes de programación de reconocimiento facial

4.1 Lenguajes de programación

Existen muchos lenguajes de programación, mediante los cuales se puede desarrollar programas de visión artificial, como lo explica (Fresno, 2020), los más reconocidos son: C++, Java y Python, en donde:

- C++: lenguaje compilado, orientado a objetos y en el cual se pueden encontrar gran variedad de bibliotecas que facilitan la labor del programador.

- Java: posiblemente el lenguaje de programación más usado y puede funcionar en diversas plataformas, pero adolece de la ausencia de una biblioteca de Visión Artificial suficientemente probada y funcional.

- Python: lenguaje fácilmente portable a otras plataformas, con una gran biblioteca estándar, además de sencillo de utilizar debido a su sintaxis, que es muy limpia y se parece mucho al inglés.

De los cuales, para el desarrollo del aplicativo de reconocimiento facial, se realizó con Python, debido que ha es el lenguaje más utilizado para este tipo de desarrollo de algoritmos y además por las importantes librerías y herramientas que se han desarrollado en este lenguaje y que pueden ser reutilizados de una manera más fácil y obtener excelentes resultados.

4.2 Paquetes y librerías de reconocimiento facial

Como lo menciona (Ghimire, 2021), las principales librerías utilizadas para desarrollar un software que permita realizar el reconocimiento facial, son las siguientes:

- Tensorflow 2.3.0
- numpy
- opencv-python
- mtcnn
- scikit-learn
- scipy

De las cuales se utilizaron para el desarrollo del software:

- Tensorflow 2.3.0
- numpy
- opencv-python
- mtcnn
- scikit-learn

4.2.1 Tensorflow 2.3.0

Como lo explican en su página web oficial, “TensorFlow es una plataforma de código abierto de extremo a extremo para el aprendizaje automático”, en donde se pueden utilizar modelos previamente entrenados o el desarrollador puede entrenar su propio modelo. Para nuestro caso se tomó un modelo previamente entrenado con muchas imágenes de rostros de seres humanos a los cuales se les identifico ciertas características relevantes para enseñarle al algoritmo a reconocer rostros humanos y diferenciar unos de otros. (TensorFlow, 2023)

4.2.2 Numpy

De acuerdo con su documentación (NumPy, 2022), Numpy es una librería que permite trabajar con arreglos de multiples dimensiones (tales como matrices y mascaras aplicadas a imágenes), también permite realizar operaciones de los arreglos matemáticas, lógicas. Manipulación de datos, ordenamiento de datos, simulaciones y mucho más.

4.2.3 OpenCV-Python

Es una de las librerías optimizadas y más completas que permiten realizar reconocimiento facial y verificación de rostros, que puede ser utilizada en lenguajes de programación como Python, C++ y Java. (OpenCV team, 2023)

4.2.4 mtcnn (Multi-Task Cascaded Convolutional Neural Networks)

Como lo menciona (Rajput, 2020), el mtcnn es una red neuronal que permite realizar reconocimiento de rostros y realizar recuadros en imágenes, el cual se basó en la publicación realizada por por Zhang, K et al en el año 2016. Y este consiste en 3 redes neuronales conectadas en cascada, el cual viene implementado en la detección de rostros que se encuentra dentro de la librería de Keras en Python 3.4.

4.2.5 Scikit-learn

Es una librería de Python, que permite realizar machine learning, por medio de análisis de datos predictivos, el cuál permite entrenar a los algoritmos, por medio de la identificación de las características más importantes en la imagen e identificar y de esta manera diferenciar los rostros de las personas autorizadas y no autorizadas. (Scikit Learn, 2023)

5. Software de reconocimiento facial

5.1 Obtención del programa de uso libre de reconocimiento facial

Para el desarrollo de este prototipo, se tomó como base el programa de uso libre desarrollado por (Ghimire, 2021), el cual cuenta con todo el entrenamiento en Machine Learning, así como imágenes de prueba que permiten identificar a 4 usuarios que cuentan con permisos.

Es importante mencionar, que este programa incluye el archivo de entrenamiento “facenet_keras.h5”, él cual fue desarrollado por (Taniai, 2018), en el que se realizó el entrenamiento por medio de machine learning, analizando muchas imágenes de rostros de personas y otras imágenes que no lo eran, con el fin de que el algoritmo logre identificar cuales si son rostros humanos y cuales no lo son.

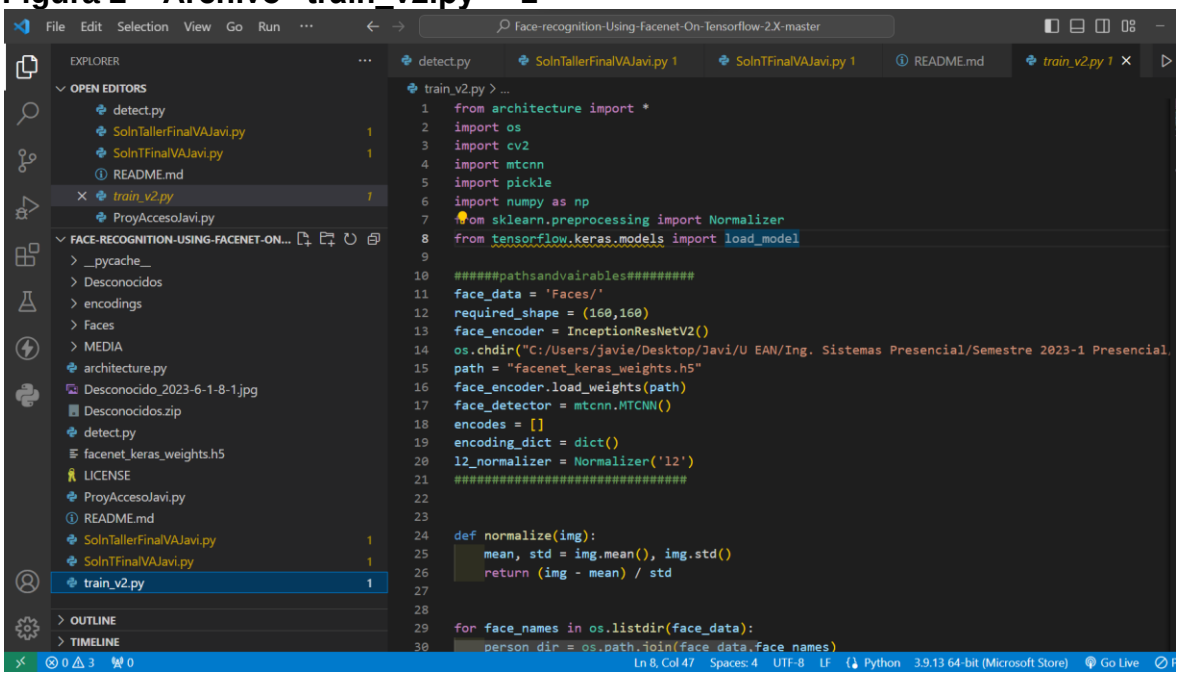
5.2 Configuración y funcionamiento Software de Reconocimiento Facial

Como se menciona en la documentación del desarrollo que se tomó de base de (Ghimire, 2021):

Para su ejecución, primero se corre el archivo “train_v2.py”, el cuál es un archivo en formato de Python, que permite realizar el entrenamiento al sistema para indicarles quienes son los usuario autorizados, y se le hace un tratamiento a las imágenes almacenadas, con el fin de únicamente tener en cuenta los factores biométricos del rostro de las personas que se encuentren en las imágenes, obteniendo así las características que se deben considerar para enseñarle al algoritmo quienes si son usuarios autorizados, y de esta manera luego el software permite identificar quien si es el usuario con permisos y quienes son personas desconocidas que no cuenta con accesos al sistema.

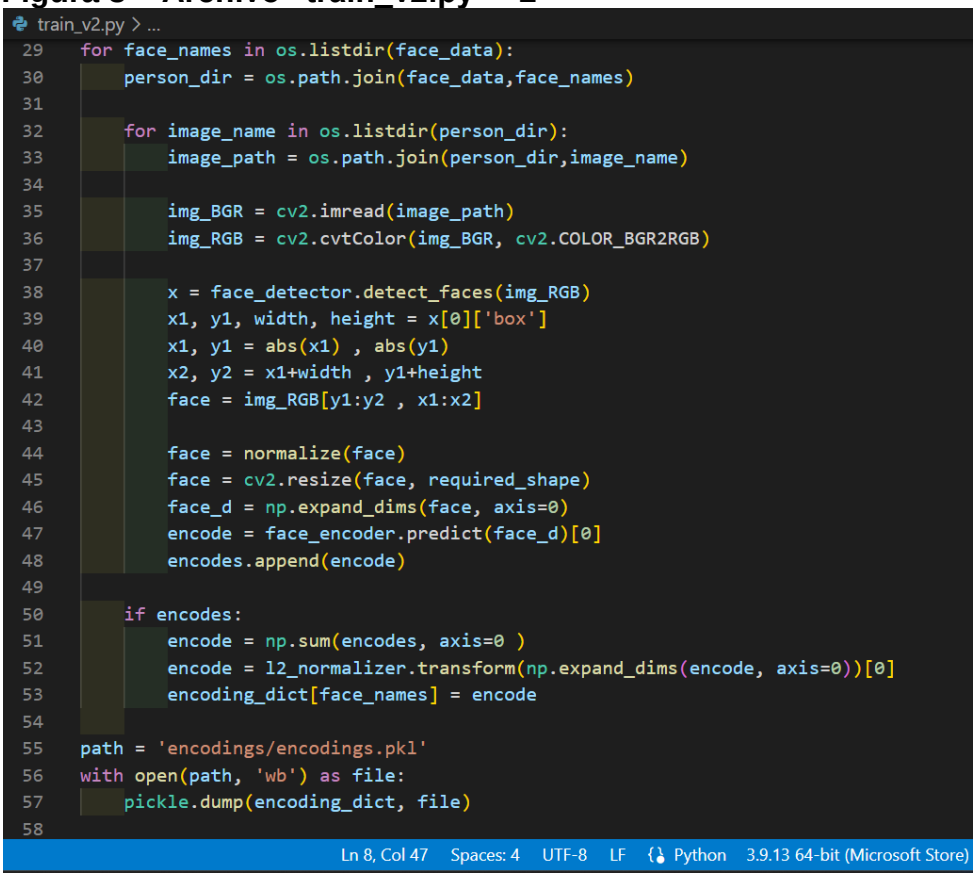
En las siguientes Figuras se mostrará el código correspondiente.

Figura 2 – Archivo “train_v2.py” - 1



Fuente: Creación propia

Figura 3 – Archivo “train_v2.py” - 2



Fuente: Creación propia

Y luego, desde el archivo principal “ProyAccesoJavi.py”, se importa la arquitectura (la cual cuenta con los pesos y características a considerar para el entrenamiento del algoritmo), y el archivo de “detect.py” que se encuentra en las Figuras 4, 5, 6 y 7, es como tal el algoritmo que permite habilitar la cámara y empezar a grabar video en vivo, el cual al activarse empezara a funcionar de corrido hasta que el usuario presione la tecla “Enter”, que como se mencionó anteriormente es lo que simularía el inicio de sesión de un usuario.

Este archivo de python de “detect.py” de igual manera a como se realizó con las imágenes de entrenamiento, también a las imágenes que va capturando del video les hace un tratamiento de imagen, permitiendo señalar el rostro de la persona en un recuadro en verde y ha está sección de interés le realiza un tratamiento adicional para verificar si realmente cumple con las características de los rasgos faciales que el algoritmo reconoce como alguna de las personas autorizadas o no.

Figura 4 – Archivo “detect.py” - 1

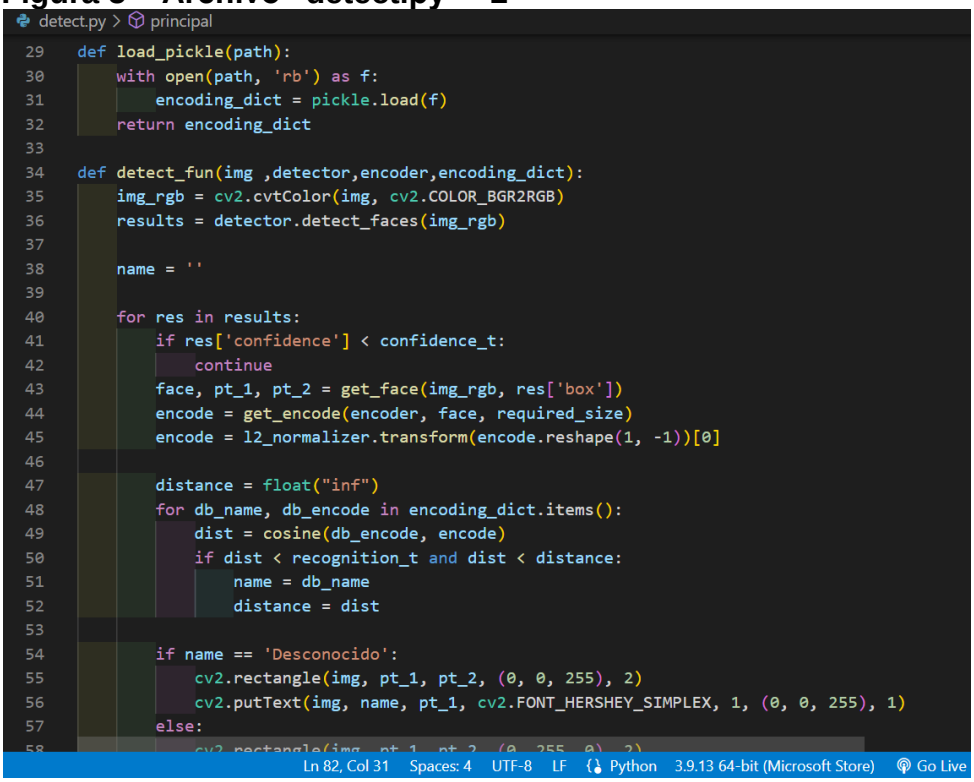
detect.py > principal

```python
import cv2
import numpy as np
import mtcnn
from architecture import *
from train_v2 import normalize,l2_normalizer
from scipy.spatial.distance import cosine
from tensorflow.keras.models import load_model
import pickle


confidence_t=0.99
recognition_t=0.5
required_size = (160,160)

def get_face(img, box):
    x1, y1, width, height = box
    x1, y1 = abs(x1), abs(y1)
    x2, y2 = x1 + width, y1 + height
    face = img[y1:y2, x1:x2]
    return face, (x1, y1), (x2, y2)

def get_encode(face_encoder, face, size):
    face = normalize(face)
    face = cv2.resize(face, size)
    encode = face_encoder.predict(np.expand_dims(face, axis=0))[0]
    return encode


def load_pickle(path):
    with open(path, 'rb') as f:
```

Ln 82, Col 31 Spaces: 4 UTF-8 LF Python 3.9.13 64-bit (Microsoft Store)

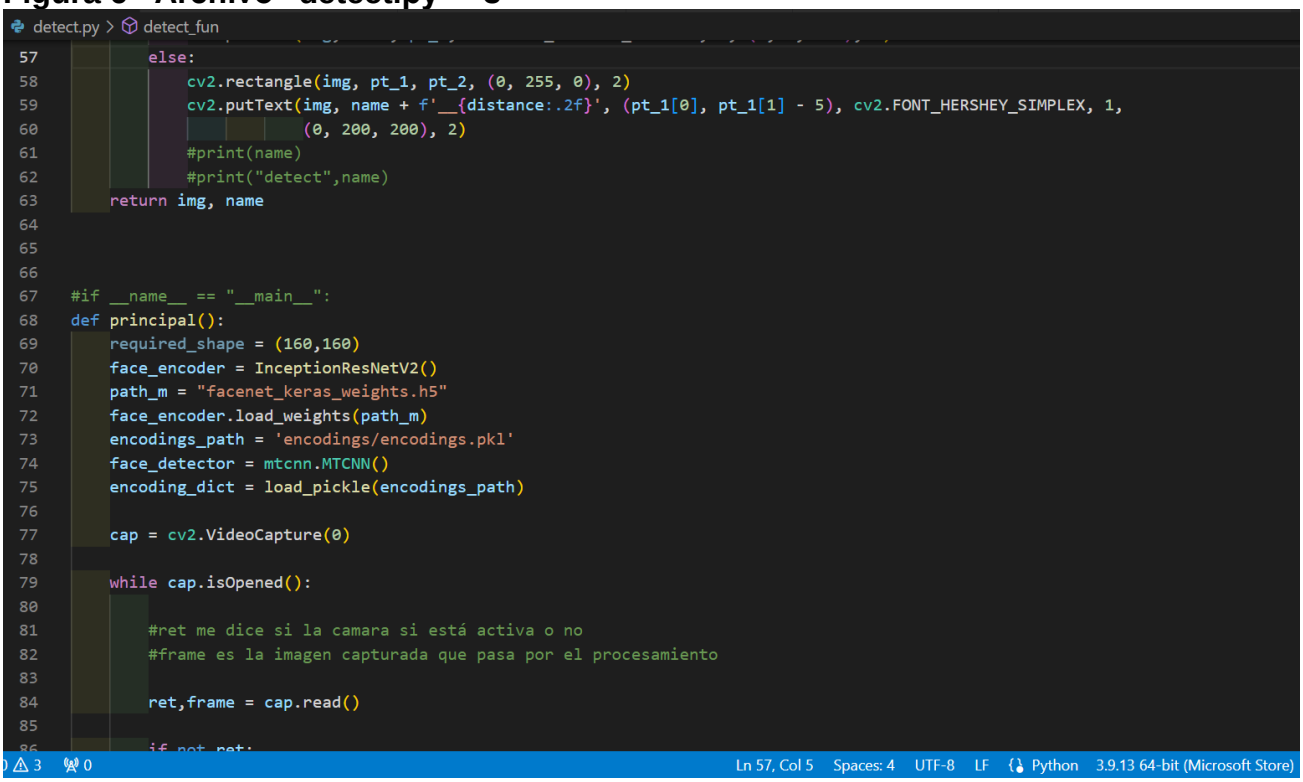
Fuente: Creación propia

Figura 5 – Archivo “detect.py” - 2



Fuente: Creación propia

Figura 6– Archivo “detect.py” - 3



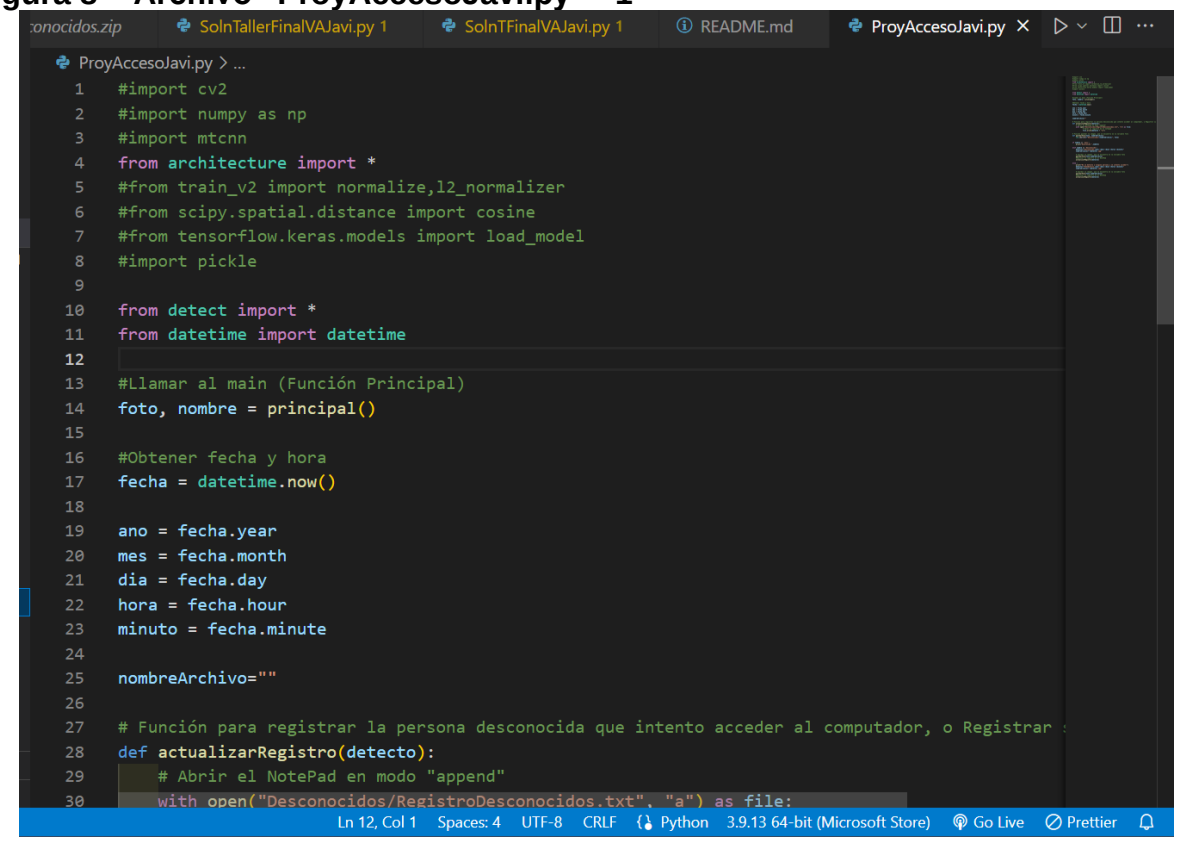
Fuente: Creación propia

Figura 7 – Archivo “detect.py” - 4

```
detect.py > detect_fun
84          ret,frame = cap.read()
85
86          if not ret:
87              print("CAMARA NO ABIERTA")
88              break
89
90          frame, name= detect_fun(frame , face_detector , face_encoder , encoding_dict)
91          #print("principal llama detect",name)
92
93          cv2.imshow('camera', frame)
94
95          if cv2.waitKey(1) & 0xFF == ord('\r'):
96              break
97
98      return frame, name
99
100
101 #principal()
102
```

Fuente: Creación propia

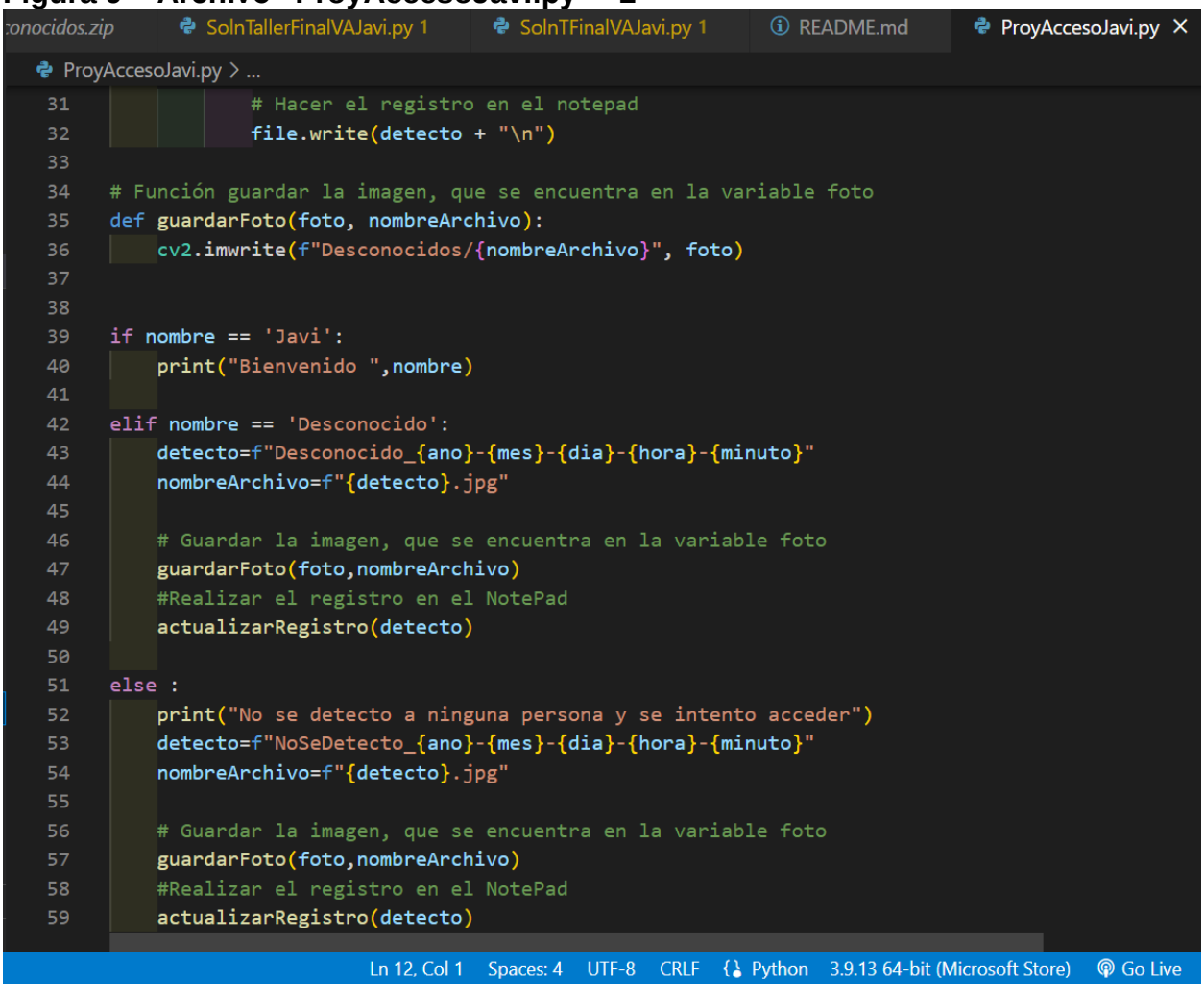
Figura 8 – Archivo “ProyAccesoJavi.py” - 1



```
:onocidos.zip   SolnTallerFinalVAJavi.py 1   SolnTFinalVAJavi.py 1   README.md   ProyAccesoJavi.py
ProyAccesoJavi.py > ...
1   #import cv2
2   #import numpy as np
3   #import mtcnn
4   from architecture import *
5   #from train_v2 import normalize,l2_normalizer
6   #from scipy.spatial.distance import cosine
7   #from tensorflow.keras.models import load_model
8   #import pickle
9
10  from detect import *
11  from datetime import datetime
12
13  #Llamar al main (Función Principal)
14  foto, nombre = principal()
15
16  #Obtener fecha y hora
17  fecha = datetime.now()
18
19  ano = fecha.year
20  mes = fecha.month
21  dia = fecha.day
22  hora = fecha.hour
23  minuto = fecha.minute
24
25  nombreArchivo=""
26
27  # Función para registrar la persona desconocida que intento acceder al computador, o Registrar
28  def actualizarRegistro(detecto):
29      # Abrir el NotePad en modo "append"
30      with open("Desconocidos/RegistroDesconocidos.txt", "a") as file:
Ln 12, Col 1   Spaces: 4   UTF-8   CRLF   Python   3.9.13 64-bit (Microsoft Store)   Go Live   Prettier
```

Fuente: Creación propia

Figura 9 – Archivo “ProyAccesoJavi.py” - 2

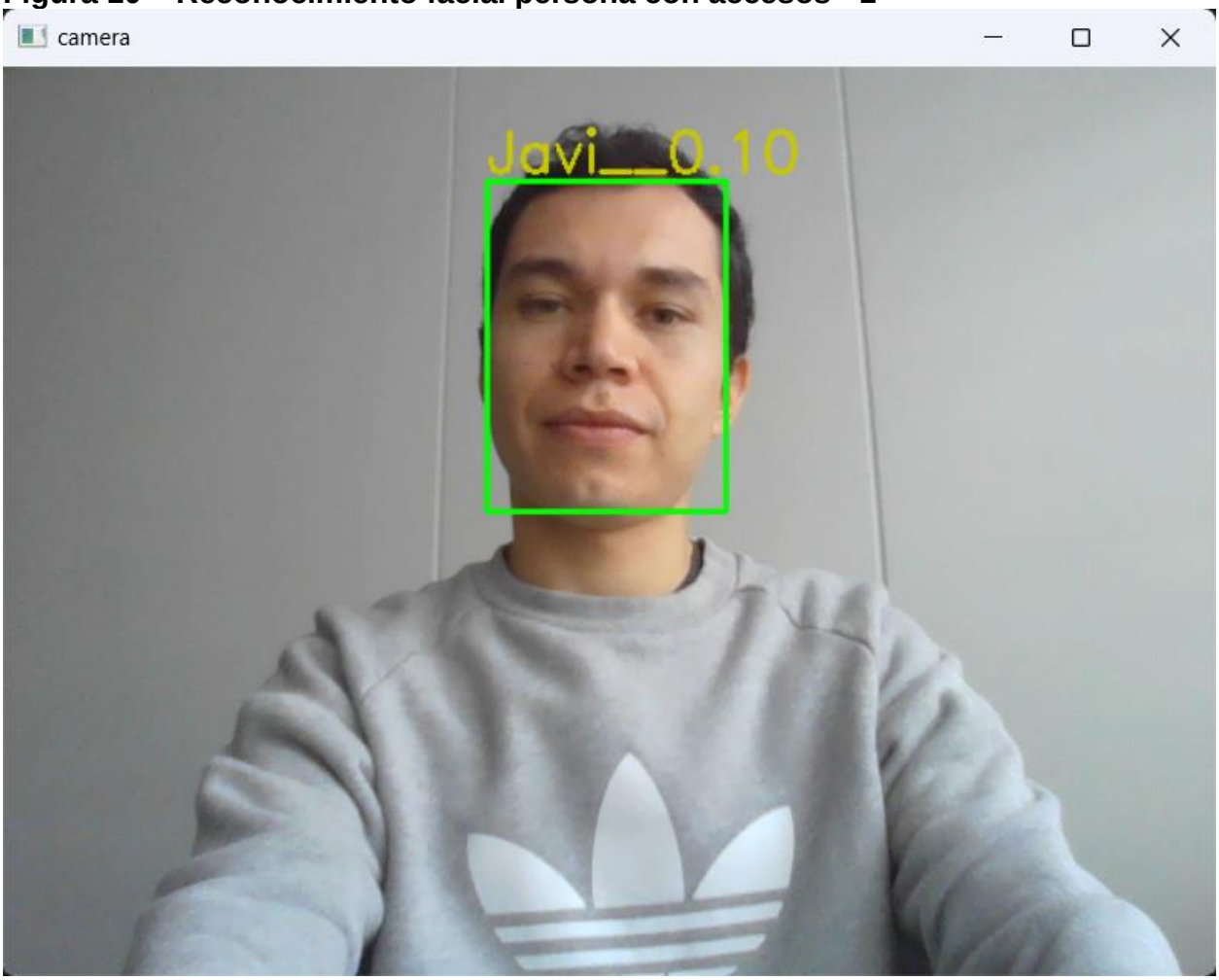


Fuente: Creación propia

En el momento en que el usuario presione la tecla “Enter” o de clic en el botón de “iniciar sesión”, en ese momento el programa toma una captura y a esta imagen se le realizará un tratamiento, el cual se enfocará en el rostro y obtendrá las características de sus rasgos faciales para identificar si realmente es un usuario con accesos o no.

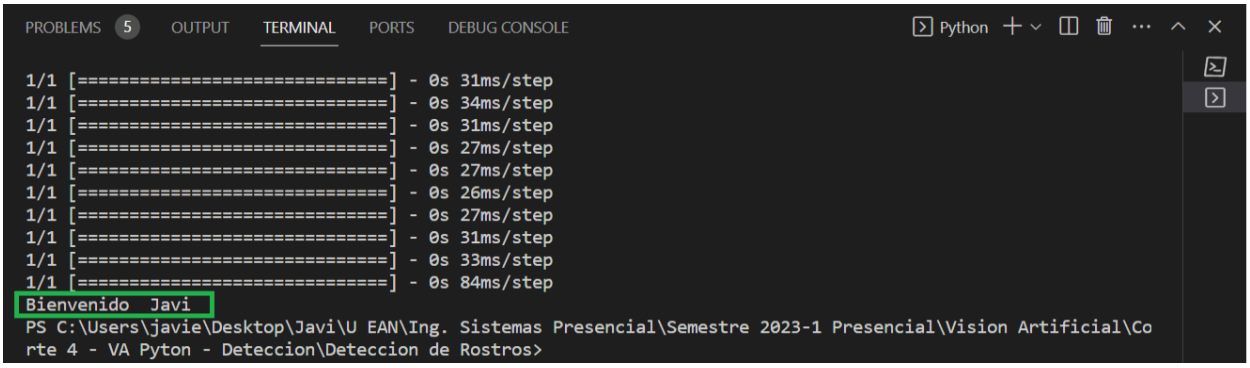
En las Figuras 10 y 11, se muestra cuando el software identifica que es un usuario autorizado, en el cual cuando se verifica que es una persona con permisos, en el computador se le mostrara un mensaje de bienvenida y se le dará acceso al sistema, mientras que, si por alguna razón no se reconoce un rostro en la imagen, o si la persona que está intentando acceder es un usuario sin permiso o no registrado en el sistema, la imagen se guarda en una carpeta, y en un archivo de notas .txt se almacenara la información de la fecha, hora y el nombre de la imagen asociada a ese momento en que se intentó acceder al sistema.

Figura 10 – Reconocimiento facial persona con accesos - 1



Fuente: Creación propia

Figura 11 – Reconocimiento facial persona con accesos - 2



Fuente: Creación propia

5.3 Histórico y almacenamiento de imágenes de usuarios no autorizados

Como se mencionó anteriormente, al software de reconocimiento, se le incluyo la opción de generar un histórico y datos de seguridad adicionales que permiten identificar quien, y cuando intento acceder al sistema, lo cual se logra mediante el siguiente código, que se evidencia en la Figura 12. Y gracias a esto, es que en una misma carpeta se almacena la imagen de la persona no autorizada y en el archivo de notas .txt se registra el nombre de la imagen y la hora, para tener un mayor control e identificar a la persona.

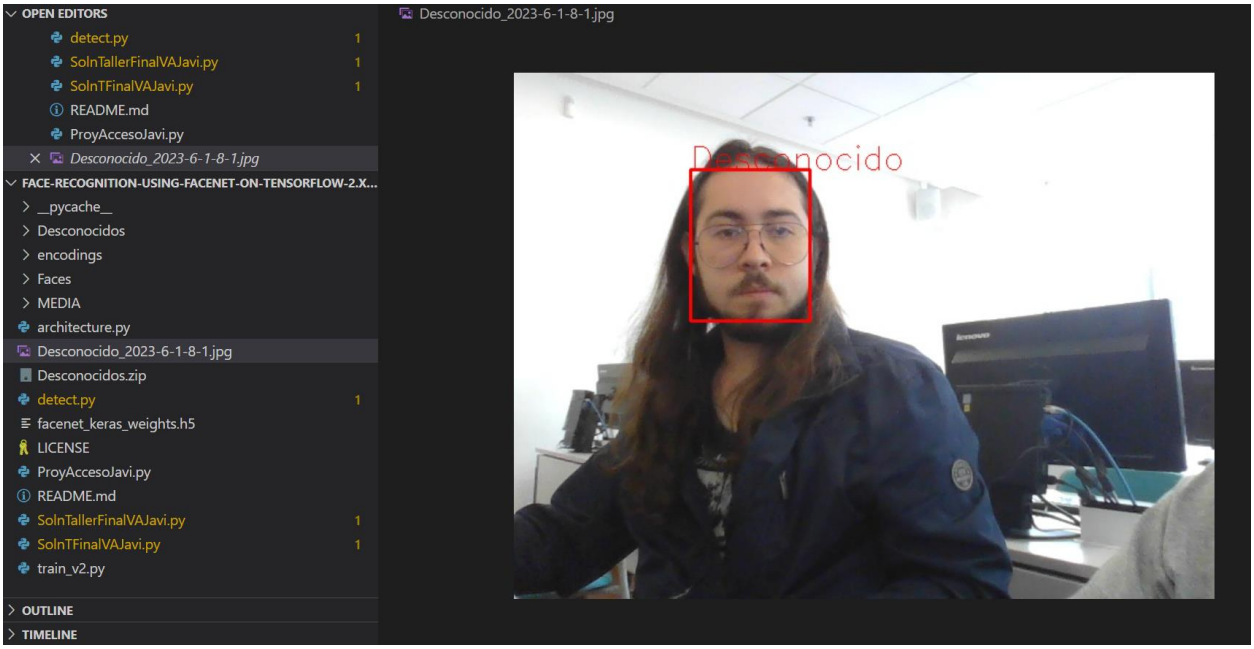
Figura 12 – Archivo "ProyAccesoJavi.py" - 3

ProyAccesoJavi.py > ...

```
39  if nombre == 'Javi':
40      print("Bienvenido ",nombre)
41
42  elif nombre == 'Desconocido':
43      detecto=f"Desconocido_{ano}-{mes}-{dia}-{hora}-{minuto}"
44      nombreArchivo=f"{detecto}.jpg"
45
46      # Guardar la imagen, que se encuentra en la variable foto
47      guardarFoto(foto,nombreArchivo)
48      #Realizar el registro en el NotePad
49      actualizarRegistro(detecto)
50
51  else :
52      print("No se detecto a ninguna persona y se intento acceder")
53      detecto=f"NoSeDetecto_{ano}-{mes}-{dia}-{hora}-{minuto}"
54      nombreArchivo=f"{detecto}.jpg"
55
56      # Guardar la imagen, que se encuentra en la variable foto
57      guardarFoto(foto,nombreArchivo)
58      #Realizar el registro en el NotePad
59      actualizarRegistro(detecto)
```

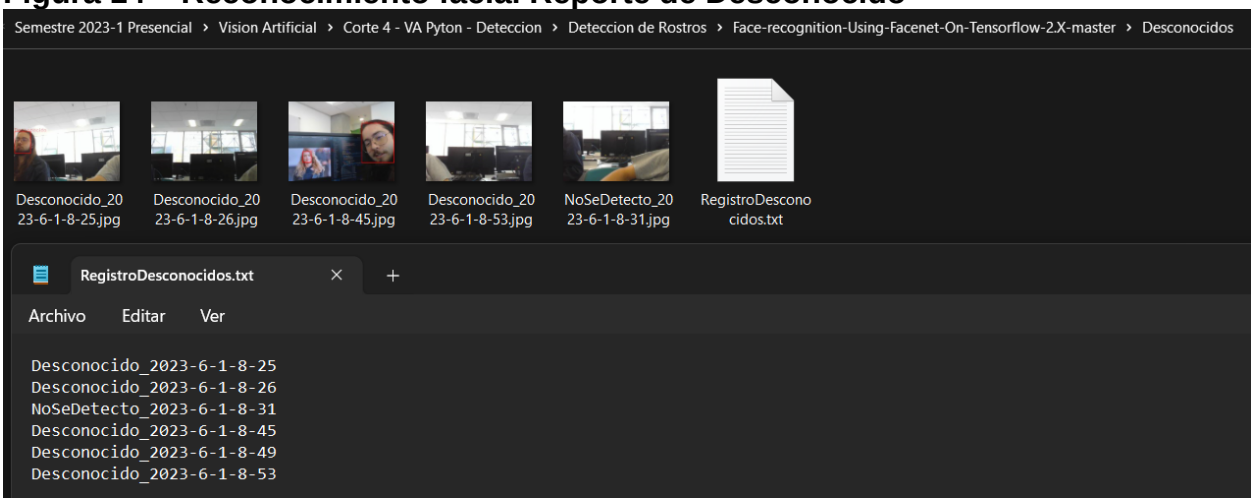
Fuente: Creación propia

Figura 13 – Reconocimiento facial Desconocido



Fuente: Creación propia

Figura 14 – Reconocimiento facial Reporte de Desconocido



Fuente: Creación propia

6. Análisis de costos

Para calcular el costo de desarrollo del aplicativo, venta y gestión del proyecto, se necesitan los siguientes recursos:

- 1 Gerente de servicios de información, el cual se encargara de la venta y negociación, así como de la gestión del proyecto para que este se lleve a cabo en el transcurso de 1 mes.

- 1 Desarrollador de software, que construya el aplicativo y lo integre con el sistema operativo del cliente.

- 1 Diseñador y administrador de bases de datos, el cual se encargara principalmente de la conexión y administración de la base de datos en el servidor web de la compañía, en el cual se almacenaran las imágenes para cada una de las compañías, garantizara que todo este configurado correctamente para la entrega al cliente, y después de la entrega del aplicativo funcionando, la compañía que lo adquiera tomara la administración de la base de datos.

- 1 Analista, el cuál realizará pruebas al aplicativo y garantizara que todo funcione y se vea correctamente.

Los siguientes salarios para cada uno de los recursos, se obtienen del análisis realizado por (WageIndicator, 2023), donde todos los perfiles cuentan con 4 años de experiencia.

Teniendo en cuenta que el aplicativo y venta se llevaran a cabo en 1 mes, la solución tendría un costo de $21’000.000, tal como se detalla en la Tabla 5.

Importante mencionar, que en otros gastos se incluirán temas de patentes, que como lo menciona el Ministerio del Interior de Colombia en una de sus páginas oficiales, “La Oficina de Registro de la Dirección Nacional de Derecho de Autor, presta el servicio gratuito de registro de obras literarias y artísticas, entre ellas el soporte lógico o software” (Dirección Nacional de Derecho de Autor, 2023), por consiguiente el único costo será el envío por correo certificado por medio de Servientrega Efectivo S.A.

Otro punto importante que se incluye en otros gastos es el de si en dado caso, se llegara a necesitar realizar el pago por el permiso del uso de código de uso libre de reconocimiento facial (paquetes y librerías desarrolladas en python) al tercero que las desarrolló, que fueron necesarias para la construcción del prototipo; pero de no ser posible su utilización, nuestro desarrollador de software creara el software desde cero. Este gasto es algo incierto por lo que se tendría que llegar a negociar con el tercero (ya que es de uso libre, pero al utilizarlos para generar una utilidad económica esto puede llegar a afectar), por lo tanto, se le asigna un valor de $1’000.000, para cubrir este tipo de gastos en dado caso que se llegaran a presentar.

De acuerdo con la Tabla 5, si se quisiera llegar a tener una rentabilidad del 20%, el aplicativo debería venderse en $25’200.000 COP (pesos colombianos). Pero como se explicará a continuación existe otro método que nos permite evaluar, cual es el precio real del software.

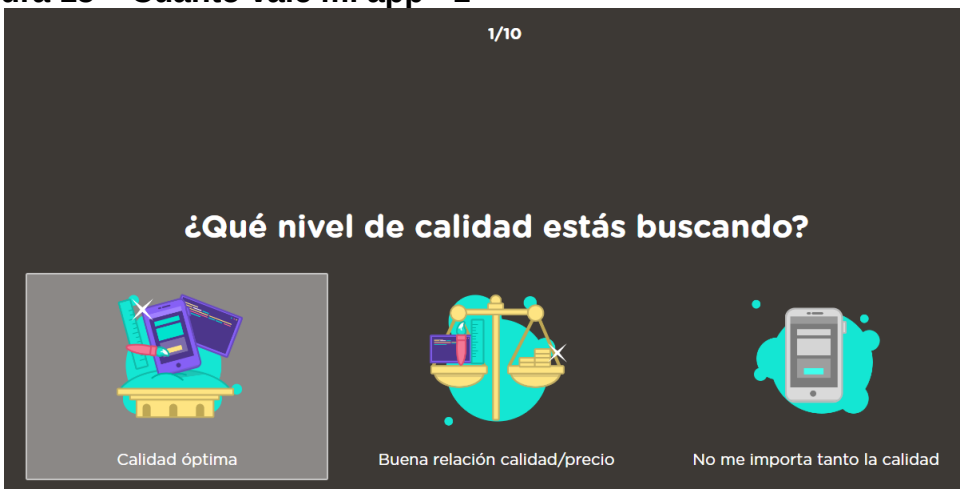
Tabla 5 – Costo de producción Software de Reconocimiento Facial

COSTOS DE PRODUCCION DEL PRODUCTO				
PRODUCTO	Software de reconocimiento facial (desarrollo y venta en 1 mes)			
DESCRIPCION	VALOR UNITARIO	UNIDADES	VALOR TOTAL	TIPO DE COSTO/GASTO
Gerente de servicios de información	$8,000,000	1	$8,000,000	Costo Directo variable
Desarrollador de Software (Full Stack)	$5,000,000	1	$5,000,000	Costo Directo variable
Diseñador de bases de datos	$4,000,000	1	$4,000,000	Costo Directo variable
Tester (Analista)	$3,000,000	1	$3,000,000	Costo Directo variable
Otros Gastos (Permisos)	$1,000,000	1	$1,000,000	Gasto Directo fijo
SUMATORIA DE COSTOS			$21,000,000	
MARGEN DE RENTABILIDAD			20%	
Valor de ganancia			$4,200,000	
PRECIO DE VENTA			$25,200,000	

Fuente: Creación propia

La otra forma de calcular el precio de venta es mediante la plataforma desarrollada por (Yeeply, 2023), teniendo en cuenta las siguientes 10 preguntas:

Figura 15 – Cuanto vale mi app - 1



Fuente: Creación propia

Al ser una aplicativo multiplataforma que va a funcionar tanto en sistema operativo Windows y Mac, seleccionamos la siguiente opción.

Figura 16 – Cuanto vale mi app - 2

Fuente: Creación propia

Al ser un aplicativo automático que funciona por detrás, no se tendrá un diseño exclusivo, lo único que se agregaría en caso de que la doble autenticación no se logré, lo único adicional será incluir el campo para que el usuario ingrese el código de seguridad de 8 dígitos.

Figura 17 – Cuanto vale mi app - 3

Fuente: Creación propia

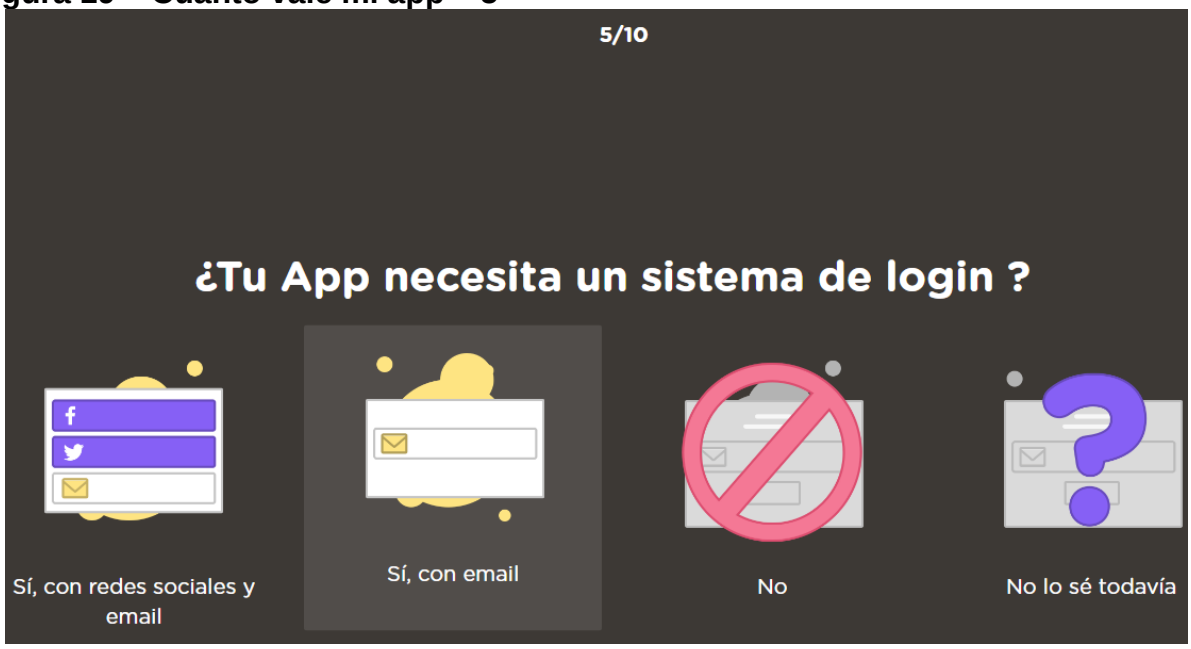
Se le venderá el aplicativo directamente a Microsoft, o a Apple para que lo incluyan en sus sistemas operativos de uso corporativo, o directamente a las compañías directamente que desee incluir está funcionalidad de doble factor de seguridad.

Figura 18 – Cuanto vale mi app - 4

Fuente: Creación propia

El aplicativo necesitara vincularse con el correo empresarial, para poder realizar el inicio de sesión.

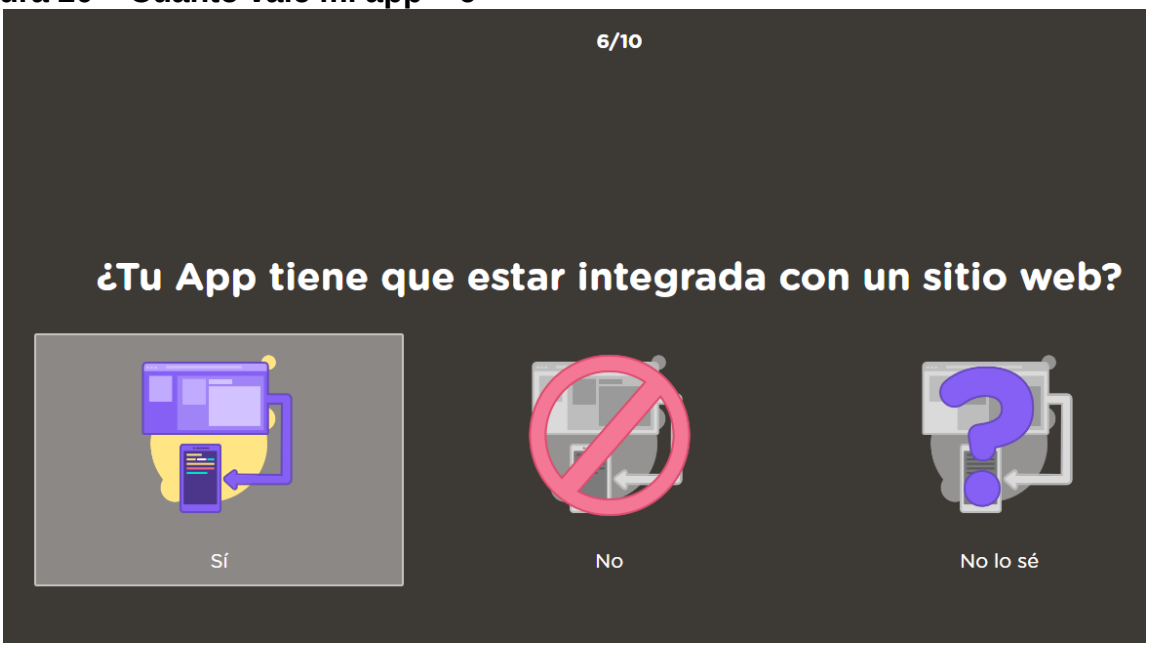
Figura 19 – Cuanto vale mi app – 5



Fuente: Creación propia

En el sitio web, se almacenarán todas las imágenes del rostro de las personas, para garantizar que realmente el usuario que está intentando acceder, realmente es la persona que dice ser.

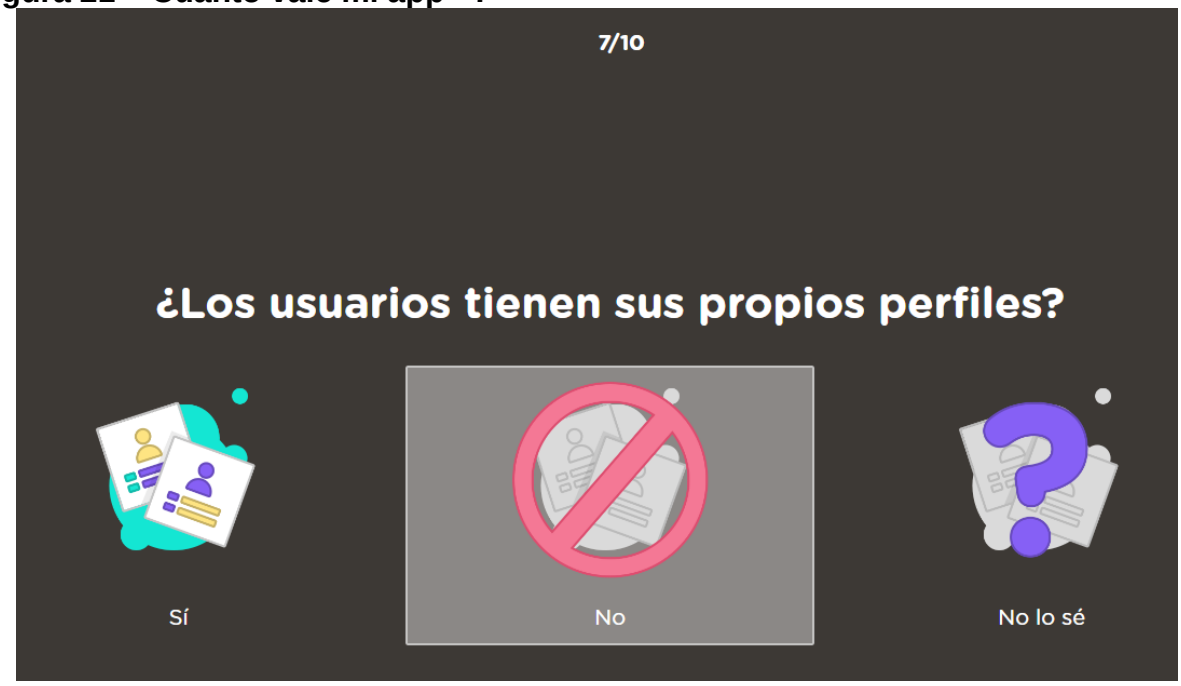
Figura 20 – Cuanto vale mi app – 6



Fuente: Creación propia

El crear un perfil como tal, no va a ser parte del aplicativo.

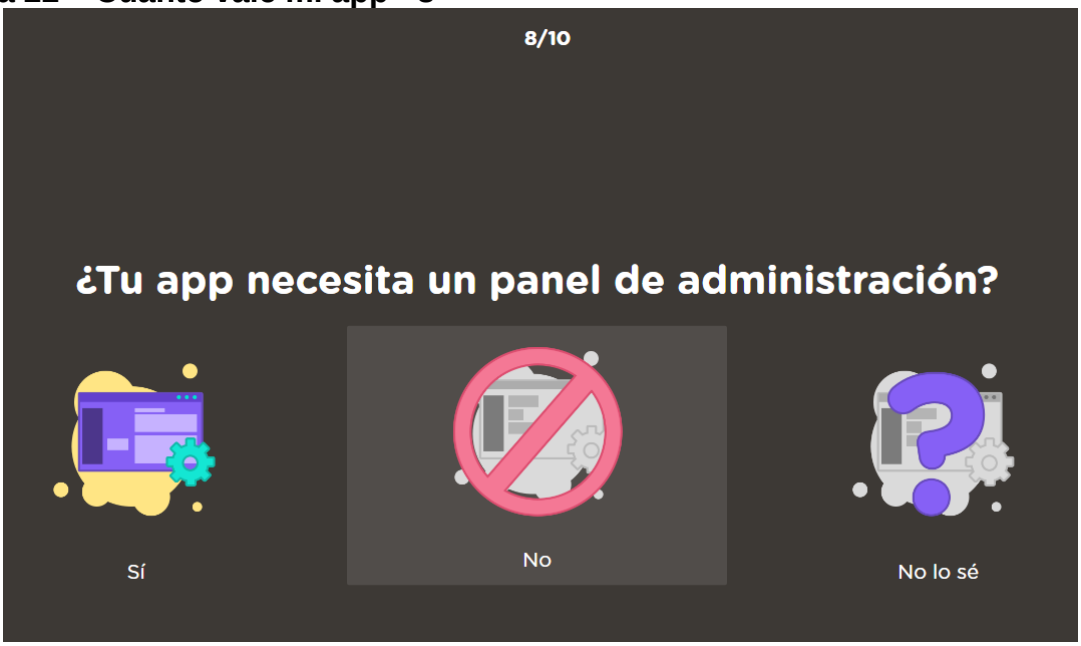
Figura 21 – Cuanto vale mi app - 7



Fuente: Creación propia

No se necesita un panel de administración, ya que todo vendrá incluido en el inicio de sesión del mismo sistema operativo, por lo tanto, todo funcionará en el backend.

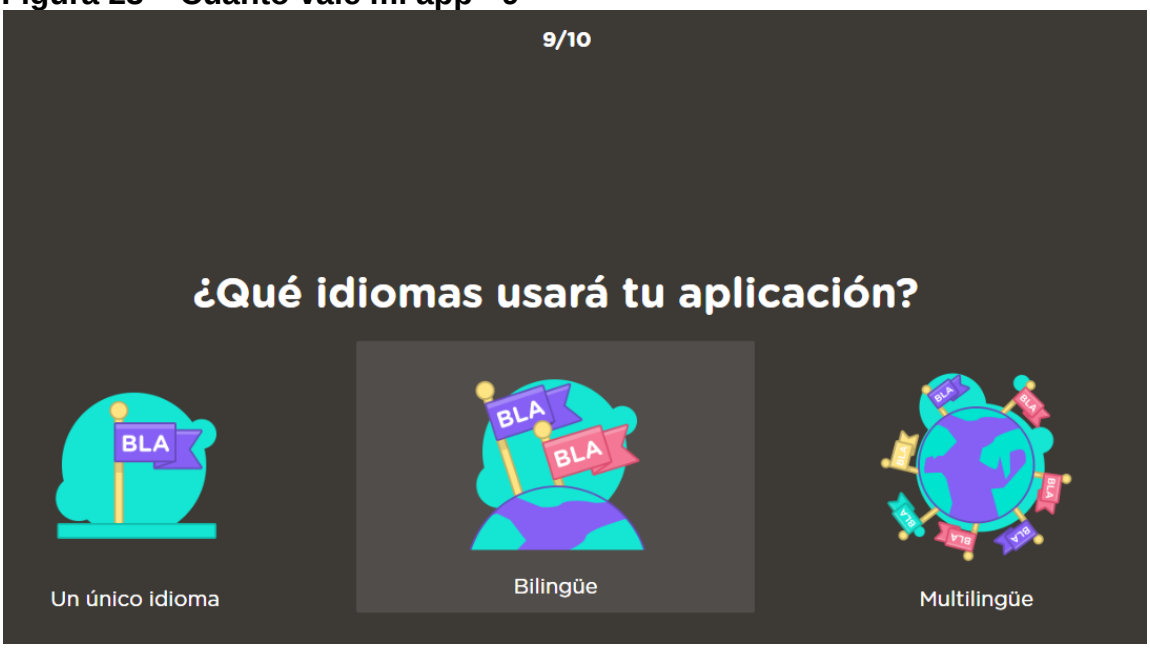
Figura 22 – Cuanto vale mi app - 8



Fuente: Creación propia

Inicialmente el aplicativo se desarrollará en Inglés y Español, pero si es adquirida por Microsoft o macOS de Apple, en una próxima integra se le incluirán todos los idiomas que incluyen estos sistemas operativos

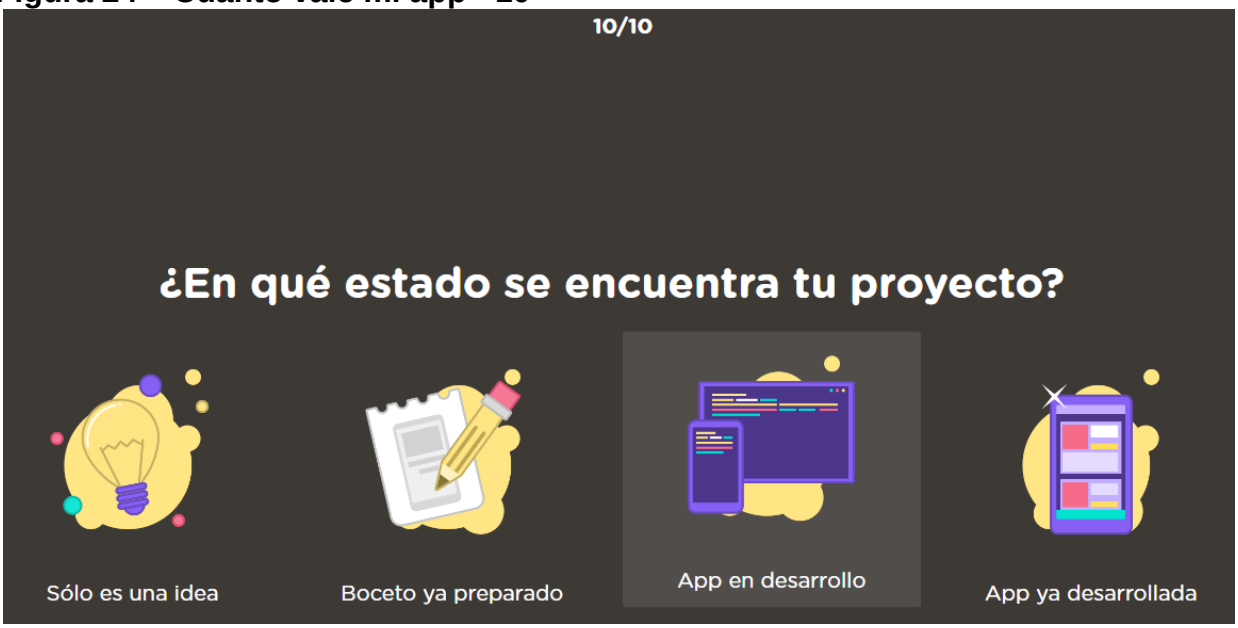
Figura 23 – Cuanto vale mi app - 9



Fuente: Creación propia

Por el momento es un prototipo que, de ser adquirido por alguna marca como Microsoft o Apple, en ese momento se incluirá directamente en sus sistemas operativos en una próxima actualización.

Figura 24 – Cuanto vale mi app - 10



Fuente: Creación propia

Teniendo en cuenta los parámetros mencionados anteriormente, según el reporte de (Yeeply, 2023), el aplicativo tendría un precio de $ 29.250.000 COP (pesos colombianos).

Figura 25 – Cuanto vale mi app - 11

Fuente: Creación propia

Gracias a los análisis realizados y a los 2 resultados obtenidos, tanto de costos asociados al desarrollo del software, como el valor del aplicativo en el mercado colombiano que se explicaron anteriormente, se estima que el software de reconocimiento facial se puede llegar a vender entre $25.200.000 y 29.250.000 pesos colombianos.

7. Plan de implementación

Se presentará el prototipo de reconocimiento facial desarrollado en el presente proyecto, inicialmente a las empresas que cuentan con sistemas operativos, tal como lo son Microsoft y Apple.

Se les venderá el software desarrollado directamente a ellos, con el fin de que estas empresas incluyan este segundo factor de autenticación de reconocimiento facial en sus componentes de inicio de sesión.

Luego de ser incluidos en las nuevas actualizaciones de los sistemas operativos, está opción se les ofrecerá a las compañías que cuentan con computadores y tienen instalado el sistema operativo correspondiente, ya sea de Microsoft o de Apple (macOS) para que la compañía que desee usar esta nueva funcionalidad la incluya en su compañía y la conecte con su servidor, el cual será el mismo en el cuál almacenan los usuarios de sus empleados. En este servidor cada empresa se encargará de almacenar los rostros de cada uno de sus empleados, con el fin de que el aplicativo identifique cuales imágenes se asocian a cuál usuario, y de esta manera el software realizara el reconocimiento de rostro y únicamente se le dará acceso al sistema al usuario que realmente es el que cuenta con los permisos para acceder a ese computador.

8. Conclusiones

- En la investigación realizada se encontraron los 3 principales lenguajes de programación que permitían desarrollar el prototipo de reconocimiento facial, y de estos se concluyó que Python es la mejor opción, debido a que este es sencillo de utilizar y cuenta con librerías de machine learning, inteligencia y visión artificial, los cuales son indispensables para el desarrollo del software de reconocimiento facial.

- El prototipo desarrollado en el presente trabajo permite identificar si la persona que está intentando iniciar sesión en el computador, es realmente el usuario que dice ser. El cual incluye un segundo factor de autenticación, para que luego de ingresar la contraseña, se active automáticamente la cámara e identifique el rostro de la persona y la compare con las imágenes asociadas a ese usuario, en donde si el reconocimiento facial es satisfactorio, el usuario iniciara sesión, pero si en dado caso no reconoce a la persona, el software almacenara la foto de esa persona que intento acceder y que no cumple con el rostro asociado, generando un registro de fotos de los momentos en que se identificó a una persona desconocida intentando acceder al sistema, en donde este registro incluirá la fecha y hora en que se intentó acceder. De igual manera si por causa de problemas con la cámara el usuario autorizado no pudiera validar su reconocimiento facil, existirá otra opción para cumplir con el segundo factor de autenticación, el cual se realizará por medio de la confirmación de un código enviado al número de celular registrado por el empleado.

- El costo total para desarrollar el software es de $ 21’000.000, y estimando una utilidad del 20%, el valor de venta del software sería igual a $25’200.000. Gracias a la página web de de Yeeply en su función del calcula de “cuánto vale mi app”, se obtuvo un precio de $29’250.000, por lo tanto, su precio de venta se negociará en este rango de precios, pero analizando los grandes beneficios que trae este aplicativo en cuanto a seguridad de la información y rapidez de verificación, su valor comercial será de $29’250.000 COP (pesos colombianos).

9. Referencias

Abhishek, K., Roshan, S., Kumar, P., & Ranjan, R. (2013). A Comprehensive Study on Multifactor Authentication Schemes. En N. Meghanathan, D. Nagamalai, & N. Chaki (Eds.), *Advances in Computing and Information Technology. Advances in Intelligent Systems and Computing* (pp. 561-568). Springer.

Avenía, C. (2017). *Fundamentos de seguridad informática*. https://core.ac.uk/download/pdf/326424171.pdf

CISA. (2021, abril). *Multi-factor authentication*. https://www.cisa.gov/sites/default/files/publications/CISA%20MultiFactor%20Auth%20HDO_040721_508.pdf

Comisión Económica para América Latina y el Caribe (CEPAL). (2021). Tecnologías digitales para un nuevo futuro. *LC/TS.2021/43*, 9. https://repositorio.cepal.org/server/api/core/bitstreams/879779be-c0a0-4e11-8e08-cf80b41a4fd9/content

Dirección Nacional de Derecho de Autor. (2023). *Registro de soporte lógico(software)*. http://derechodeautor.gov.co:8080/software

Fresno, S. (2020). *Implementación de un sistema software de visión artificial para la detección de objetos en movimiento* [Universitat Politècnica de València]. https://riunet.upv.es/bitstream/handle/10251/151678/Fresno%20-%20Implementaci%C3%B3n%20de%20un%20sistema%20software%20de%20visi%C3%B3n%20artificial%20para%20la%20detecci%C3%B3n%20de%20objetos%20....pdf?sequence=1

Ghimire, R. (2021, noviembre 2). *Face-recognition-Using-Facenet-On-Tensorflow-2.X*. https://github.com/R4j4n/Face-recognition-Using-Facenet-On-Tensorflow-2.X

IBM. (2022). *¿Qué es la información de identificación personal (PII)?* https://www.ibm.com/es-es/topics/pii#:~:text=La%20informaci%C3%B3n%20de%20identificaci%C3%B3n%20personal%20(PII)%20es%20cualquier%20informaci%C3%B3n%20relacionada,su%20direcci%C3%B3n%20de%20correo%20electr%C3%B3nico.

IT User. (2023, septiembre 15). *Cómo detectar y proteger rápidamente los datos confidenciales*. https://discoverthenew.ituser.es/seguridad-inteligente/2023/09/como-detectar-y-proteger-rapidamente-los-datos-confidenciales

Misini, E., & Lajçi, U. (2022). *Biometric Authentication*. https://www.researchgate.net/publication/371567274_Biometric_Authentication

Noema, Y. (2021, diciembre 8). *Top 3 Programming Languages For Implementing a Computer Vision System*. imagescv. https://medium.com/imagescv/my-top-3-programming-languages-for-implementing-a-computer-vision-system-2aa0a3ad0a2a

NumPy. (2022). *NumPy Documentation*. https://numpy.org/doc/stable/

OpenCV team. (2023). *OpenCV Face Recognition*. https://opencv.org/opencv-face-recognition/

Parmar, D., & Mehta, B. (2014). *Face Recognition Methods & Applications*. https://www.researchgate.net/publication/260483303_Face_Recognition_Methods_Applications

Rajput, S. (2020, septiembre 1). *Face Detection using MTCNN*. https://medium.com/@saranshrajput/face-detection-using-mtcnn-f3948e5d1acb

Scikit Learn. (2023). *Scikit-learn Machine Learning in Python*. https://scikit-learn.org/stable/

Soriano, M. (2023). *¿Veremos Face ID en los MacBook de 2024?* https://lamanzanamordida.net/noticias/rumores/veremos-face-id-macbook-2024/

Stanford. (2017). *Introduction to Computer Vision* (R. Krishna, Ed.). Stanford University. http://vision.stanford.edu/teaching/cs131_fall1718/files/cs131-class-notes.pdf

Symantec. (2022). *Secure Authentication Anywhere*. https://vip.symantec.com/

Taniai, H. (2018, junio 20). *keras-facenet*. https://github.com/nyoki-mtl/keras-facenet

TensorFlow. (2023). *Crea modelos de aprendizaje automático de nivel de producción con TensorFlow*. https://www.tensorflow.org/?hl=es-419

Verdium. (2019, agosto 13). *Why Windows Hello doesn't work for all organizations*. https://veridiumid.com/microsoft-windows-hello-passwordless-authentication/

WageIndicator. (2023). *Comparador Salarial*. https://tusalario.org/colombia

Yeeply. (2023). *¿Cuánto cuesta desarrollar mi app?* https://www.cuantocuestamiapp.co/