



UNIVERSIDAD NACIONAL DEL ALTIPLANO
FACULTAD DE INGENIERÍA MECÁNICA ELÉCTRICA,
ELECTRÓNICA Y SISTEMAS
ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA



PROTOTIPO DE UN SISTEMA DE RECONOCIMIENTO
FACIAL PARA EL CONTROL BIOMÉTRICO EN EL
COLEGIO APLICACIÓN DE LA UNIVERSIDAD
NACIONAL DEL ALTIPLANO PUNO -2019

TESIS

PRESENTADA POR:

Bach. ANGEL BETO MAMANI AQUINO

Bach. RHOAN CHRISTIAN CANAHUIRE QUISPE

PARA OPTAR EL TÍTULO PROFESIONAL DE:

INGENIERO ELECTRÓNICO

PUNO – PERÚ

2022



DEDICATORIA

Con estima y respeto a mis queridos padres, quienes con su apoyo y consejos supieron guiarme en el transcurso de mi vida y apoyarme incondicionalmente sobre todas las cosas en mi formación profesional. Finalmente dedico este trabajo a aquellos compañeros que no se encuentran ya con nosotros, ya que el destino desvaneció sus sueños y los llevo a aquel camino sin retorno.

Angel Beto Mamani Aquino



DEDICATORIA

Dedico este proyecto a mis padres FLORENCIO CANAHUIRE MENDOZA y ROSA QUISPE DE CANAHUIRE como también a mis hermanos RONI, MARIZOL, RAFAEL y DEYSI, ya que, con su incondicional apoyo y ejemplos dignos de superación y entrega dentro de mi formación académica, así como integral, eh logrado que este esfuerzo se culmine con éxito, puesto que siempre han estado a mi lado en los peores y bonitos momentos de mi vida

Rhoan Christian Canahuire Quispe



AGRADECIMIENTO

En primer lugar: A nuestros padres por su confianza, asertividad y esfuerzo incansable quienes nunca dudaron en brindarnos su apoyo incondicional a lo largo de toda nuestra vida y formación académica. A mis docentes a quienes debo gran parte de mis conocimientos, gracias a su paciencia y enseñanza. Finalmente un eterno agradecimiento a esta prestigiosa universidad la cual abrió sus puertas a jóvenes como nosotros, preparándonos para un futuro competitivo y formándonos como personas de bien.

Angel Beto Mamani Aquino

En primer lugar, agradezco a mi casa de estudios superior, la Universidad Nacional del Altiplano Puno quien me abrió las puertas el día que decidí estudiar. Agradezco a los docentes de mi carrera profesional De Ingeniería Electrónica quienes fueron los pilares en la formación académica durante mi vida universitaria, a mis jurados por guiarme y apoyarme en la culminación de este presente trabajo de investigación. A mis padres, familiares, amigos quienes estuvieron siempre a mi lado alentándome y aun amigo en especial que se alegraría de verme titulado pero por cosas del destino partió al infinito antes de tiempo, Gracias Rocky.

Rhoan Christian Canahuire Quispe



ÍNDICE GENERAL

DEDICATORIA

AGRADECIMIENTO

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

ÍNDICE ACRÓNIMOS

RESUMEN 14

ABSTRACT..... 15

CAPITULO I

INTRODUCCIÓN

1.1. OBJETIVO DE LA INVESTIGACIÓN..... 21

1.1.1. Objetivo general 21

1.1.2. Objetivos específicos 21

CAPITULO II

REVISIÓN DE LITERATURA

2.1. ANTECEDENTES DE LA INVESTIGACIÓN 22

2.1.1. Antecedentes internacionales: 22

2.1.2. Antecedente nacionales: 23

2.2. MARCO TEÓRICO 24

2.2.1. El reconocimiento facial 24

2.2.2. Visión artificial..... 25

2.2.3. La tecnología 2d: 28

2.2.4. Python: 28

2.2.5. Microsoft visual studio: 29



2.2.6.	Librerías de software (bibliotecas):.....	29
--------	---	----

CAPITULO III

MATERIALES Y MÉTODOS

3.1.	MATERIALES.....	36
3.1.1.	Hardware	36
3.1.2.	Software	39
3.2.	MÉTODO	42
3.2.1.	Diseño de investigación	42
3.2.2.	Nivel de la investigación.....	42
3.2.3.	Población de la investigación.....	43
3.2.4.	Ámbitos de estudio investigación.....	43
3.2.5.	Recolección de datos.....	44
3.2.6.	Contratación de hipótesis	45

CAPITULO IV

RESULTADOS Y DISCUSIÓN

4.1.	INTERPRETACIÓN DE RESULTADOS	60
4.1.1.	Experiencia en el desarrollo de software	60
4.1.2.	Código para la creación de base de datos.....	60
4.1.3.	Eliminar grupo de personas.....	61
4.1.4.	Registrar persona.....	61
4.1.5.	Código de registro de personas	62
4.1.6.	Inicio de registro de personas.....	63
4.1.7.	Ingreso de datos para pruebas del prototipo.....	63
4.1.8.	Ventana de confirmación inicio de toma de muestras	64
4.1.9.	Ventana emergente inicio de toma de muestras.....	64



4.1.10.	Toma de muestras de imagen.....	65
4.1.11.	Finalización de toma de muestra.....	66
4.1.12.	Reenvió de imágenes tomadas	66
4.1.13.	Finalización de envío de muestras a la app azure microsoft:.....	67
4.1.14.	Finalizado toma de datos.....	68
4.1.15.	Finalización del registro	68
4.2.	EJECUCIÓN DEL PROGRAMA.....	69
4.2.1.	Ejecutamos el programa de reconocimiento	70
4.2.2.	Inicio de reconocimiento.....	70
4.2.3.	Muestras	71
4.2.4.	Detección.....	72
4.2.5.	Detección de rostros	72
4.2.6.	Cambio de Ángulo	73
4.2.7.	Reconocimiento de rostros	74
4.2.8.	Reconocimiento.....	74
4.2.9.	Finalización	75
4.3.	DISEÑO DE UN PROTOTIPO DE SISTEMA DE RECONOCIMIENTO FACIAL QUE FACILITE Y AHORRE TIEMPO AL INGRESO.....	76
4.3.1.	Discusión.....	79
4.3.2.	Contrastación de hipótesis.....	80
4.4.	DISEÑO DE UN PROTOTIPO DE SISTEMA DE RECONOCIMIENTO FACIAL PARA EL CONTROL BIOMÉTRICO QUE PERMITA LA SISTEMATIZACIÓN DE LA ASISTENCIA.....	83
4.4.1.	Discusión.....	83
4.4.2.	Contrastación de hipótesis.....	84



V. CONCLUSIONES.....	88
VII. RECOMENDACIONES	89
VII. REFERENCIAS BIBLIOGRÁFICAS.....	90
ANEXOS.....	95
ANEXO 1	95
ANEXO 2.....	97

Área : robótica

Tema : inteligencia artificial

Fecha de sustentación 11 de febrero de 2022.



ÍNDICE DE FIGURAS

Figura N° 1: Logo de sistema PYTHON.....	28
Figura N° 2: Logo de Microsoft visual studio.....	29
Figura N° 3: Logo de Librería Open CV.....	31
Figura N° 4: Logo de Librería Dlib.....	32
Figura N° 5: Logo de biblioteca NumPy.....	34
Figura N° 6: Logo de Software Multiplataforma CMake.....	35
Figura N° 7: Ordenador Portátil.....	36
Figura N° 8: Arduino Uno.....	37
Figura N° 9: Cámara IQ.....	38
Figura N° 10: Cable periférico para arduino uno.....	39
Figura N° 11: Vista satelital de colegio de Aplicación de la Universidad Nacional del Altiplano.....	44
Figura N° 12: Sistema de usuario administrado.....	46
Figura N° 13: Capas del sistema.....	49
Figura N° 14: Arquitectura.....	52
Figura N° 15: Funcionamiento del sistema.....	53
Figura N° 16: Código de base de datos.....	53
Figura N° 17: Código reinicio de sistema.....	54
Figura N° 18: Formulario de datos estudiante.....	55
Figura N° 19: Ventana de confirmación.....	55
Figura N° 20: Ventana de confirmación e inicio.....	56
Figura N° 21: Toma de muestras.....	56
Figura N° 22: Toma de muestras.....	57
Figura N° 23: Creacion de ID.....	57
Figura N° 24: Confirmación de finalización.....	58



Figura N° 25: Inicio de reconocimiento facial.	58
Figura N° 26: Proceso de reconocimiento.	59
Figura N° 27: Ventana de pruebas.	59
Figura N° 28: Creación base de datos	60
Figura N° 29: Creación base de datos	61
Figura N° 30: Eliminar person group	61
Figura N° 31: Registro.	62
Figura N° 32: Código de registro	62
Figura N° 33: Registro de personas	63
Figura N° 34: Ingreso de datos ensayo de prototipo	63
Figura N° 35: Confirmación de muestras	64
Figura N° 36: Ventana de inicio	64
Figura N° 37: Captura de muestras	65
Figura N° 38: Toma de muestras captadas	65
Figura N° 39: Toma de muestras	66
Figura N° 40: Reenvió de muestras	66
Figura N° 41: Envío de datos.	67
Figura N° 42: Envío de datos.	67
Figura N° 43: Finalización de toma de datos	68
Figura N° 44: Ventana de finalización del registro.	69
Figura N° 45: Inicio de ejecución de código.	69
Figura N° 46: Ejecución de código de reconocimiento.	70
Figura N° 47: Inicio de reconocimiento.	71
Figura N° 48: Ventana de inicio de captura.	71
Figura N° 49: Toma de muestras.	72
Figura N° 50: Etapa de detección	72



Figura N° 51: Detección.....	73
Figura N° 52: Cambio de Ángulo.....	74
Figura N° 53: Reconocimiento.....	74
Figura N° 54: Reconocimiento.....	75
Figura N° 55: Finalización.	75
Figura N° 56: Diagrama de lista de reconocimiento facial	76
Figura N° 57: Diagrama de relación de asistentes y el tiempo de registro.....	78
Figura N° 58: Diagrama general de reconocimiento facial.	79
Figura N° 59: Calculo para Z crítico.	81
Figura N° 60: Prueba de hipótesis campana de Gauss	82
Figura N° 61: Calculo para Z crítico.	86
Figura N° 62: Prueba de hipótesis campana de Gauss	87



ÍNDICE DE TABLAS

Tabla N° 1: Detalle de usuario 01.....	46
Tabla N° 2: Detalle de usuario 02.....	47
Tabla N° 3: Restricciones del sistema.	47
Tabla N° 4: Lista de reconocimiento facial.	76
Tabla N° 5: Relación de asistentes y el tiempo de registro.	77
Tabla N° 6: Resumen general de reconocimiento facial.	78
Tabla N° 7: Datos para la contratación de hipótesis.....	80
Tabla N° 8: Cuadro de prueba de hipótesis.	82
Tabla N° 9: Lista de control biométrico para la asistencia.....	83
Tabla N° 10: Datos para la contratación de hipótesis.....	84
Tabla N° 11: Cuadro de prueba de hipótesis.	86



ÍNDICE ACRÓNIMOS

- USB : Universal Serial Buss
- CMAKE : Cross Platform Make
- BSD : Berkeley Software Distribution



RESUMEN

El presente proyecto estuvo abocado al diseño de un prototipo de un sistema de reconocimiento facial usando el lenguaje de programación PYTHON, para el control biométrico en el colegio Aplicación de la Universidad Nacional del Altiplano. Esta investigación por sus características se considera de tipo aplicativo y experimental. Con la intención de integrar la tecnología de reconocimiento de rostros con el aplicativo Microsoft Azure, para lograr ello se requerirá de las librerías como Open CV. Es por ello, que se utilizó la tecnología de Microsoft Cognitive para la comparación de rostros, y de esta forma el proceso se analizó con una mayor eficiencia. Asimismo, se realizó un prototipo de interfaz para el usuario, teniendo como resultado que el control de reconocimiento facial que facilita y ahorre tiempo al ingreso, en la cual se puede observar que en el registro N° 13 se tiene una duración de tiempo de 88.60 segundos en condición de reconocido, así mismo en el reconocimiento N° 01 se presente un tiempo menor al resto con 5.50 segundo, finalmente se concluye la investigación indicando que el sistema de reconocimiento facial cumple con el control adecuado con respecto al tiempo y el registro de participante e enviar a una base de datos de la cantidad de participantes a la que se le aplica.

PALABRAS CLAVE: Reconocimiento facial, Python, control biométrico.



ABSTRACT

The present project was devoted to the design of a prototype of a facial recognition system using the PYTHON programming language, for biometric control in the Application school of the National University of the Altiplano. Due to its characteristics, this research is considered to be of an applicative and experimental type. With the intention of integrating face recognition technology with the Microsoft Azure application, to achieve this, libraries such as Open CV will be required. For this reason, Microsoft Cognitive technology was used to compare faces, and in this way the process was analyzed with greater efficiency. Likewise, a user interface prototype was made, resulting in the facial recognition control that facilitates and saves time at entry, in which it can be seen that record No. 13 has a time duration of 88.60 seconds in recognized condition, likewise in recognition No. 01 a time less than the rest with 5.50 seconds is presented, finally the investigation is concluded indicating that the facial recognition system complies with the adequate control with respect to time and registration of participant and send to a database of the number of participants to which it applies.

Keywords: Facial recognition - PYTHON, biometric control.



CAPITULO I

INTRODUCCIÓN

La presente tesis titulada “Diseñar un prototipo de sistema de reconocimiento facial para el control biométrico en el colegio Aplicación de la Universidad Nacional del Altiplano”. Se desarrolló para brindar una alternativa más eficiente al software de reconocimiento facial del mercado existente en la actualidad.

El presente trabajo pretende desarrollar un prototipo de sistema de reconocimiento facial para control de ingreso al colegio Aplicación de la Universidad Nacional del Altiplano y de esta manera recabar una base de datos del ingreso de personas a la Institución Educativa.

El uso de la programación PYTHON será el medio para el desarrollo del software a desarrollar, así también el proceso de reconocimiento se verificará en la plataforma Visual Studio para realizar las pruebas de funcionamiento del sistema.

La investigación consta de 4 capítulos desarrollados de la siguiente manera.

El capítulo I referido al problema de la investigación en que se aborda la descripción y las causas del problema, bajo el enfoque de la metodología, así mismo se desarrolló la introducción de nuevas tecnologías en la actualidad, de la misma forma la justificación del porque es importante el desarrollo de un prototipo de reconocimiento facial.

El capítulo II en este capítulo describimos la teoría en donde se pudo apreciar las bases teóricas que hicieron comprensible las distintas investigaciones desarrolladas relacionadas al reconocimiento facial, verificando los aspectos básicos del



reconocimiento de rostros en donde se vio cada una de las pautas que implica el desarrollo de este software de reconocimiento de rostros.

El capítulo III se hizo una descripción del tipo de investigación realizada para el desarrollo del tema, de cómo fueron procesados y analizados los datos, el ámbito del estudio, la metodología del desarrollo del sistema.

En el capítulo IV se abordó el análisis e interpretación en base a los resultados obtenidos de las pruebas de simulación, así mismo se incluyó las conclusiones, recomendaciones, y bibliografías de la investigación.

PLANTEAMIENTO DEL PROBLEMA.

FORMULACIÓN DEL PROBLEMA.

Durante mucho tiempo se tiene descuidado el control de ingreso de personas al interior del colegio Aplicación de la Universidad Nacional del Altiplano, quedando como único registro los apuntes que se realizan al inicio de la sesión de cada clase por el docente, dicho control de asistencia es rudimentario para la época actual, se debe de utilizar las herramientas que tenemos a la mano para lograr la eficiencia y mejorar la seguridad respecto al control e identificación de esta entidad escolar, por tal sentido el reconocimiento de rostros automático será una posible solución para tal problema, la presente investigación realizará el desarrollo de software de reconocimiento facial utilizando el lenguaje de programación PYTHON para el colegio Aplicación de la Universidad Nacional del Altiplano.

HIPÓTESIS DE LA INVESTIGACIÓN.

HIPÓTESIS GENERAL.

Es posible diseñar un prototipo de sistema de reconocimiento facial para el control biométrico en el colegio Aplicación de la Universidad Nacional del Altiplano.



HIPÓTESIS ESPECÍFICA

- El diseño del prototipo de sistema de reconocimiento facial para el control biométrico utilizando las librerías de OpenCV y la app Microsoft Azure garantizara la seguridad al ingreso del colegio Aplicación de la Universidad Nacional del Altiplano.
- El diseño de un prototipo de sistema de reconocimiento facial para el control biométrico utilizando las librerías de OpenCV y la app Microsoft AZURE facilitara el ahorro de tiempo en un 90% al ingreso del colegio Aplicación de la Universidad Nacional del Altiplano.
- El diseño de un prototipo de sistema de reconocimiento facial para el control biométrico, que permita la sistematización en un 95% de la asistencia en el colegio Aplicación de la Universidad Nacional del Altiplano.

JUSTIFICACIÓN DEL PROBLEMA.

En las instituciones educativas se tienen como régimen la asistencia puntual del alumnado, docentes y personal administrativo que influyen en el desenvolvimiento óptimo de las actividades durante la semana, para lo cual la presente investigación se rige a este hecho de la asistencia puntual y responsabilidad, estas son normas de convivencia en las instituciones que se debe cumplir a carta cabal, seguidamente el director de la institución es el directo responsable de hacer cumplir la normativa expuesta como el control de acceso y la asistencia, se debe llevar de manera individual durante los días de la semana del periodo escolar, en diversas instituciones el parte diario de ingreso de personas y docentes es suscrito de forma manual; por lo que en ocasiones no se lleva seguimiento de su correcta suscripción y funcionamiento, por lo que al tener este



inconveniente se pueden suscitar fraudes y equivocaciones en el funcionamiento de este sistema establecido que es el registro general.

Este tipo de investigación es de suma importancia para el control adecuado en las instituciones públicas y privadas, las grandes potencias lo están estudiando para su implementación ya sea en seguridad, salud, educación y militar, investigadores y desarrolladores de empresas privadas como Google, Microsoft, IBM entre otros son quienes desarrollan softwares cada vez más interesantes para reconocimiento facial, objetos entre otros, y además ofrecen computación en la nube (del inglés Cloud Computing) por medio de este captan la atención de empresas y el público, ofreciéndoles servicios gratuitos como (reconocimiento facial, reconocimiento de objetos entre otros servicios) algunos servicios de manera gratuita y otros pagando según al volumen de datos utilizados. Por ejemplo, las empresas que dan servicio de reconocimiento facial mediante sus APIS o SDK son: IBM con su Cloud Computing Bluemix y mediante este se puede hacer uso de su API Watson, por parte de Microsoft está el software Azure, mediante este se puede utilizar el API Face, y de Google esta Google cloud este ofrece su API Visión. Si se hace uso de este servicio (reconocimiento facial) estas empresas cobran según la cantidad de imagen procesada, por ejemplo, IBM permite crear un modelo personalizado gratuito utilizando hasta 5000 imágenes de entrenamiento que permite clasificar hasta 7.500 imágenes por mes utilizando cualquier modelo preformado o personalizado con un límite de 250 imágenes por día (Cáceres, 2018).

Investigadores y científicos se encuentran optimizando los algoritmos de reconocimiento facial así como (Tesillo, 2016) afirma que la inteligencia artificial (IA) y redes neuronales es la que se utiliza en entrenamiento para el reconocimiento de objetos, personas buscando el tiempo de respuesta más corto posible.



Este tipo de herramientas se utilizan con la finalidad de desarrollar prototipos que con el tiempo se verán empleados en las personas con interfaces sencillas, así también tenga aplicaciones en la seguridad para así dar la escalabilidad del prototipo existente en la red de algoritmos realizados en código abierto como Open CV, las que fueron detalladas en la investigación de la tesis.

Este estudio será el futuro de la identificación portátil con el desarrollo posterior, para ser conectados a diferentes bases de datos del estado peruano que están ubicadas en entidades municipales, policiales, puestos de salud, colegios y escuelas. Es por eso que esta investigación está dirigida al desarrollo que implica el control biométrico para mejorar el nivel de seguridad en el acceso, para mejorar la eficiencia del registro de asistencia del personal que ingresa a instituciones con alto índice de fluctuación.



1.1.OBJETIVO DE LA INVESTIGACIÓN

1.1.1. Objetivo general

Diseñar un prototipo de software para el sistema de reconocimiento facial y control biométrico al ingreso del colegio de Aplicación de la Universidad Nacional del Altiplano.

1.1.2. Objetivos específicos

- Diseñar un prototipo de software de reconocimiento facial para el control biométrico que permita mejorar la seguridad al ingreso del colegio de Aplicación de la Universidad Nacional del Altiplano.
- Diseñar un prototipo de sistema de reconocimiento facial que facilite y ahorre tiempo al ingreso del colegio de Aplicación de la Universidad Nacional del Altiplano.
- Diseñar un prototipo de sistema de reconocimiento facial para el control biométrico que permita la sistematización de la asistencia en el colegio de Aplicación de la Universidad Nacional del Altiplano.



CAPITULO II

REVISIÓN DE LITERATURA

2.1.ANTECEDENTES DE LA INVESTIGACIÓN

2.1.1. Antecedentes internacionales:

En la tesis Titulado “Diseño de un Software Piloto de Reconocimiento Facial para el Control de Asistencia en la Escuela de Telecomunicaciones de la Universidad de Carabobo” siguiente trabajo presenta el diseño de un software piloto que optimiza el registro de asistencia de la Escuela de Telecomunicaciones de la Universidad de Carabobo, brindando un mayor nivel fiabilidad por medio de la detección y reconocimiento facial de todo el personal que labora en el lugar, para ello, con la ayuda de la librería OpenCV se usó la tecnología de visión artificial, captura y comparación de imágenes, procesamiento digital de la mismas y de esta forma identificar a las personas. Además, con el uso del lenguaje de programación PYTHON se realizó una interfaz gráfica de usuario para la administración del software realizando pruebas para evaluar la eficiencia y eficacia del mismo (Mosquera & Romero , 2016).

En la tesis titulado “Reconocimiento facial por el método de eigenfaces” este trabajo se presenta una aplicación de detección facial por Eigenfaces que usa Microsoft Visual Studio con diversas herramientas de programación como C# y librerías de la plataforma Emgu CV asociadas a OpenCV. El desarrollo de la aplicación se codifico en dos partes: una de registro y la otra de reconocimiento. El programa está cargado en una computadora de escritorio con sistema operativo Windows 8 y se usa la webcam integrada. En la etapa de registro el individuo se presenta ante la cámara para tomarle tres fotografías en diferentes instantes. En la etapa de reconocimiento el usuario se presenta ante la cámara y el sistema hace una comparación con todos los registros existentes en la



base de datos indicando si el usuario está registrado. Tanto en la etapa de registro como en la etapa de reconocimiento se crean otros formularios que interconectan ambas, utilizando templates del explorador de soluciones por HaarCascade, CANNY_PRUNING y el objetivo EigenObjectRecognizer. El proyecto se sometió a la parte experimental como un universo del tamaño de diez usuarios, en el que ocho usuarios son hombres y dos mujeres, haciendo diez pruebas por usuario obteniendo una matriz de confusión con resultados del 100% de reconocimiento incluso con usuarios no registrados (Caballero, Reyes, Sánchez, & Ríos, 2017).

2.1.2. Antecedente nacionales:

En la tesis titulado "Aplicación móvil de reconocimiento facial en personas con trabajo previos de abuso sexual en la provincia de Andahuaylas, Apurímac - 2018", para optar de Ingeniero de Sistemas en la Universidad Nacional José María Arguedas, Apurímac. El desarrollo del sistema plantea la inexistencia de un sistema de reconocimiento para identificar a las personas que cometen estos actos delictivos en la ciudad de Andahuaylas en donde los acusados casi en su mayoría salen sin represalias de la justicia, entonces el objetivo principal es el de realizar un proceso de reconocimiento facial para identificar a los individuos que cuenten con antecedentes delictivos mediante el uso de un aplicativo instalado en un dispositivo móvil, ya con esta justificación la investigación se manejó con esta tecnología ya que está siendo de mucha efectividad en otros países además de mejorar los procesos de investigación analizada. Con la tecnología de desarrollo se tomó una muestra de 30 individuos y el mismo número de reportes fotográficos para realizar una sesión de pruebas del para detectar fallas en el aplicativo por el que el resultado fue el 100% de las nuestras fueron identificadas de manera eficiente y el 93% tiene consistencia y solo el 7% por factores externos ajenos al software no se generó el reconocimiento. En conclusión, el proyecto desarrollado es que el prototipo



realiza el reconocimiento sin ser afectado por factores externos, adicionalmente realizaron en este proyecto la implementación de una base de datos al sistema para realizar mejor su efectividad para la búsqueda de posibles sospechosos. (Mariño, 2018).

En la tesis titulado “detección de alumnos mediante mecanismos proactivos para el control de asistencia para la facultad de ingeniería y arquitectura de la universidad san Martín”, Utiliza la comunicación por bluetooth y la plataforma Android para generar una aplicación que sondea en intervalos de tiempo determinado a todos los equipos que se encuentran dentro de su alcance y permite el monitoreo constante de los personas en el salón de clases, este proyecto está en la etapa de un prototipo con la implementación de un software desarrollado en la plataforma Android para el personal docente y el estudiante. Este sistema sería de gran utilidad ya que se lograría acoplar al sistema de dicha universidad así mostrar el itinerario del estudiante así pueda generar un monitoreo al momento de la clase. Los resultados que muestran e en el proyecto demuestra una reducción de un 84.69%, en un salón formado por 108 personas, en el monitoreo de ingreso al aula. Dicho estudio fue realizado en una casa superior del país, en donde se suscita problemáticas similares a la nuestra, en mayor razón cuando el conjunto de personas supera las 50 personas. (Sandoval & Estela Villena, 2015).

2.2.MARCO TEÓRICO

2.2.1. El reconocimiento facial

En la vida cotidiana normalmente nos sorprendemos de nuestra capacidad para reconocer rostros, incluso de personas a las cuales no vemos desde hace años. Distintas edades, razas, lejos o cerca, con mala o buena iluminación, sin importar la circunstancia, siempre tenemos la capacidad de reconocer un rostro en base a nuestra experiencia previa o recuerdos.



Sin embargo, esta problemática no es tan simple cuando se intenta realizar por medios computacionales, utilizando hardware y software que tienen limitaciones y que dependen de la calidad de información previamente ingresada respecto del rostro que se desea identificar y de la metodología utilizada.

Dado lo anterior, el reconocimiento facial forma parte del área de estudio denominada Biometría, cuyo propósito consiste en el desarrollo de métodos automatizados para la identificación o verificación de personas mediante características físicas o de comportamiento.

2.2.2. Visión artificial

Según (Larcher, BIASONI, Cattaneo, Ruggeri, & Herrera, 2011) La visión artificial tiene como objetivo la recolección de información a partir de muestras de imágenes, con el empleo de un equipo de cómputo. Un software de reconocimiento artificial trabaja sobre una imagen de una realidad que le proporciona información sobre brillo, colores, formas, etc. Estas imágenes suelen estar estáticas, escenas tridimensionales o imágenes en movimiento.

2.2.2.1. Mediciones Manuales Por Bledsoe (1960):

Muchos dirían que el padre del reconocimiento facial fue Woodrow Wilson Bledsoe. Trabajando en la década de 1960, Bledsoe desarrolló un sistema que podía clasificar fotos de rostros a mano utilizando lo que se conoce como una tableta RAND, un dispositivo que las personas podían usar para ingresar coordenadas horizontales y verticales en una cuadrícula utilizando un lápiz óptico que emitía pulsos electromagnéticos. El sistema podría usarse para registrar manualmente las ubicaciones de coordenadas de varias características faciales, incluidas los ojos, la nariz, la línea del cabello y la boca.

Estas métricas podrían insertarse en una base de datos. Luego, cuando el sistema recibió una nueva fotografía de un individuo, fue capaz de recuperar la imagen de la base de datos que más se parecía a esa persona. En ese momento, el reconocimiento facial por desgracia estaba limitado severamente por la tecnología de la era y el poder de procesamiento de la computadora. Sin embargo, fue un primer paso importante para probar que el reconocimiento facial era una biometría viable. (Mao & Valencia, 2021).

2.2.2.2. Mayor Precisión Con 21 Marcadores Faciales (1970):

En la década de 1970, Goldstein, Harmon y Lesk fueron capaces de agregar una mayor precisión a un sistema de reconocimiento facial manual. Utilizaron 21 marcadores subjetivos específicos, incluidos el grosor del labio y el color del cabello, para identificar caras automáticamente. Al igual que con el sistema de Bledsoe, la biometría real tuvo que ser aún computada manualmente.

2.2.2.3. Eigenfaces (Finales De 1980 – Principios De 1990):

En 1988, Sirovich y Kirby comenzaron a aplicar el álgebra lineal al problema del reconocimiento facial. Lo que se conoció como el enfoque Eigenface comenzó como una búsqueda de una representación de baja dimensión de las imágenes faciales. Sirovich y Kirby pudieron demostrar que el análisis de las características de una colección de imágenes faciales podría formar un conjunto de características básicas. También pudieron demostrar que se necesitaban menos de cien valores para codificar con precisión una imagen de cara normalizada.

En 1991, Turk y Pentland ampliaron el enfoque Eigenface descubriendo cómo detectar rostros dentro de las imágenes. Esto condujo a las primeras instancias de reconocimiento facial automático. Su enfoque se vio limitado por factores tecnológicos y



ambientales, pero fue un gran avance en la demostración de la viabilidad del reconocimiento facial automático.

2.2.2.4.Base De Datos Forenses De Aplicación De La Ley (2009):

En 2009, la Oficina del Alguacil del Condado de Pinellas creó una base de datos forense que permitía a los oficiales acceder a los archivos de fotos del Departamento de Seguridad Vial y Vehículos Motorizados (DHSMV) del estado. Para 2011, unos 170 diputados habían sido equipados con cámaras que les permitían tomar fotografías de sospechosos que podrían cotejarse con la base de datos. Esto resultó en más arrestos e investigaciones criminales de lo que hubiera sido posible de otro modo.

2.2.2.5.Medios Sociales (2010-Presente):

A partir de 2010, Facebook comenzó a implementar la funcionalidad de reconocimiento facial que ayudó a identificar a las personas cuyas caras pueden aparecer en las fotos que los usuarios de Facebook actualizan a diario. Si bien la función fue instantáneamente controvertida con los medios de comunicación, lo que generó una gran cantidad de artículos relacionados con la privacidad, a los usuarios de Facebook en general no pareció importarles. Al no tener un impacto negativo aparente en el uso o la popularidad del sitio web, más de 350 millones de fotos se cargan y etiquetan mediante el reconocimiento facial cada día.

2.2.2.6.Iphone x (2017):

Apple lanzó el iPhone X en 2017, anunciando el reconocimiento facial como una de sus características principales. El sistema de reconocimiento facial en el teléfono se usa para la seguridad del dispositivo. El nuevo modelo de iPhone se vendió casi al instante, lo que demuestra que los consumidores ahora aceptan el reconocimiento facial como el nuevo estándar de oro para la seguridad.

2.2.3. La tecnología 2d:

La tecnología 2D funciona bien en condiciones estables y de buena iluminación, como es el caso en el control de pasaportes. Sin embargo, es menos efectiva en espacios más oscuros y no puede ofrecer buenos resultados cuando los sujetos se mueven. Es fácil de engañar con una fotografía.

2.2.4. Python:

PYTHON es un lenguaje de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código. Se trata de un lenguaje de programación multiparadigma, ya que soporta parcialmente la orientación a objetos, programación imperativa y en menor medida, programación funcional. Es un lenguaje interpretado, dinámico no compilado.

Es administrado por la PYTHON Software Foundation License. El cual posee una licencia de código abierto, denominada “PYTHON Software Foundation License”

Figura N° 1: Logo de sistema PYTHON



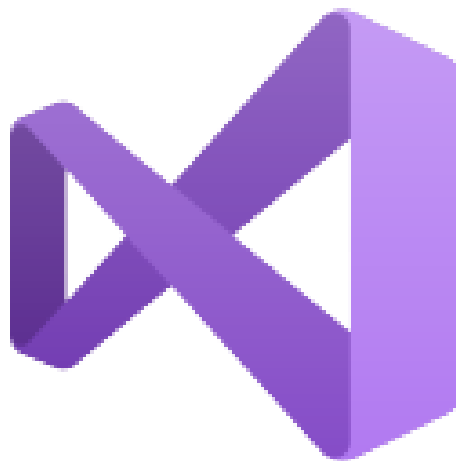
Fuente: (<https://es.wikipedia.org/wiki/Archivo:PYTHON-logo-notext.svg>)

2.2.5. Microsoft visual studio:

Microsoft Visual Studio es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para Windows y Mac OS X. Es compatible con múltiples lenguajes de programación, tales como C++, C#, Visual Basic .NET, F#, Java, PYTHON, Ruby y PHP, al igual que entornos de desarrollo web, como ASP.NET MVC, Django, etc., a lo cual hay que sumarle las nuevas capacidades en línea bajo Windows Azure en forma del editor Mónaco.

Visual Studio permite a los desarrolladores crear sitios y aplicaciones web, así como servicios web en cualquier entorno compatible con la plataforma .NET (a partir de la versión .NET 2002). Así, se pueden crear aplicaciones que se comuniquen entre estaciones de trabajo, páginas web, dispositivos móviles, dispositivos embebidos y videoconsolas, entre otros (Jaliri, y otros, 2019).

Figura N° 2: Logo de Microsoft visual studio



Fuente: (https://es.wikipedia.org/wiki/Microsoft_Visual_Studio)

2.2.6. Librerías de software (bibliotecas):

Las librerías de software son desarrolladores de software que trabajan en programación orientada a objetos que generan un sin número de bibliotecas las cuales generan ayuda en el momento de la creación de algoritmos para otras aplicaciones.



El beneficio más importante de las librerías de software es el reciclado que permite adaptar algoritmos ya utilizados con anterioridad y adaptar códigos ya existentes, los cuales ya se ven mejorados con los trabajos de depuración de los desarrolladores encargados de crear dicha librería (Aguilar 2008) afirma que en la actualidad tenemos en desarrollo un sinnúmero de bibliotecas específicas para cada tipo de aplicación en los distintos lenguajes de programación, por ejemplo uno de las más comunes son las bibliotecas del lenguaje C, tales entregan ficheros de cabecera y bibliotecas con métodos o rutinas que se implementa en acciones comunes.

2.2.6.1.Open CV:

OpenCV es una biblioteca libre de visión artificial originalmente desarrollada por Intel. OpenCV significa Open Computer Vision (Visión Artificial Abierta). Desde que apareció su primera versión alfa en el mes de enero de 1999, se ha utilizado en una gran cantidad de aplicaciones, y hasta 2020 se la sigue mencionando como la biblioteca más popular de visión artificial.¹ Detección de movimiento, reconocimiento de objetos, reconstrucción 3D a partir de imágenes, son sólo algunos ejemplos de aplicaciones de OpenCV.

Su popularidad se debe a que es:

- II. Libre, publicada bajo licencia BSD, que permite que sea usada libremente para propósitos comerciales y de investigación.
- III. Multiplataforma, para los sistemas operativos GNU/Linux, Mac OS X, Windows y Android, y para diversas arquitecturas de hardware como x86, x64 (PC), ARM (celulares y Raspberry Pi).
- IV. Documentada y explicada: la organización tiene una preocupación activa de mantener la documentación de referencia para desarrolladores lo más completa y actualizada posible, ejemplos de uso de sus funciones y tutoriales accesibles al

público no iniciado en visión artificial, además de difundir y fomentar libros y sitios de formación.

El proyecto pretende proporcionar un entorno de desarrollo fácil de utilizar y altamente eficiente. Esto se ha logrado realizando su programación en código C y C++ optimizados, aprovechando además las capacidades que proveen los procesadores multinúcleo. OpenCV puede además utilizar el sistema de primitivas de rendimiento integradas de Intel, un conjunto de rutinas de bajo nivel específicas para procesadores Intel (IPP).

Figura N° 3: Logo de Librería Open CV



Fuente: (<https://es.wikipedia.org/wiki/OpenCV>)

2.2.6.2.DLIB:

Dlib es una biblioteca de software multiplataforma de propósito general escrita en el lenguaje de programación C ++. Su diseño está fuertemente influenciado por ideas de diseño Tkinter

Tkinter es un binding de la biblioteca gráfica Tcl/Tk para el lenguaje de programación PYTHON. Se considera un estándar para la interfaz gráfica de usuario (GUI) para PYTHON y es el que viene por defecto con la instalación para Microsoft Windows.

Hoy en día hay otras alternativas disponibles como PYTHON, PyQt o PySide y PyGTK que cumplen con todos los estándares de componente por contrato e ingeniería de software basada en componentes. Por lo tanto, es, ante todo, un conjunto de componentes de software independientes. Es un software de código abierto publicado bajo una licencia de software Boost.

Desde que comenzó el desarrollo en 2002, Dlib ha crecido para incluir una amplia variedad de herramientas. A partir de 2016, contiene componentes de software para trabajar con redes, subprocessos, interfaces gráficas de usuario, estructuras de datos, álgebra lineal, aprendizaje automático, procesamiento de imágenes, minería de datos, análisis XML y de texto, optimización numérica, redes bayesianas y muchas otras tareas. En los últimos años, gran parte del desarrollo se ha centrado en la creación de un amplio conjunto de herramientas estadísticas de aprendizaje automático y en 2009 se publicó Dlib en la Revista de investigación sobre aprendizaje automático. [2] Desde entonces, se ha utilizado en una amplia gama de dominios.

Figura N° 4: Logo de Librería Dlib



Fuente: (<https://en.wikipedia.org/wiki/Dlib>)



2.2.6.3. NUMPY

NumPy (pronunciado /'nʌmpai/ (numpai) o, a veces /'nʌmpi/ 34 (numpi) es una biblioteca para el lenguaje de programación PYTHON que da soporte para crear vectores y matrices grandes multidimensionales, junto con una gran colección de funciones matemáticas de alto nivel para operar con ellas. El precursor de NumPy, Numeric, fue creado originalmente por Jim Hugunin con contribuciones de varios otros desarrolladores. En 2005, Travis Oliphant creó NumPy incorporando características de la competencia Numarray en Numeric, con amplias modificaciones. NumPy es un software de código abierto y cuenta con muchos colaboradores.

Una forma de superar esos defectos es a través de la detección de prueba de vida. Esos sistemas buscarán indicadores de una imagen no en vivo, como características inconsistentes entre el primer plano y el fondo. Pueden pedir al usuario que parpadee o se mueva. Son necesarios para vencer a los delincuentes que intentan engañar a los sistemas de reconocimiento facial mediante el uso de fotografías o máscaras.

Otro avance clave es la 'red neuronal convolucional profunda'. Se trata de un tipo de máquina de aprendizaje en la que un modelo encuentra patrones en los datos de la imagen. Despliega una red de neuronas artificiales que imita el funcionamiento del cerebro humano. En efecto, la red se comporta como una caja negra. Se le dan valores de entrada cuyos resultados aún no se conocen. Luego realiza comprobaciones para garantizar que la red esté produciendo el resultado esperado. Cuando este no es el caso, el sistema realiza ajustes hasta que se configura correctamente y puede producir sistemáticamente los resultados esperados.

Hoy en día, los procesos previamente avanzados están llegando a los dispositivos del mercado masivo. Por ejemplo, Apple usa tecnología de cámara 3D para la función térmica basada en infrarrojo Face ID en su iPhone X. Las imágenes térmicas IR mapean

los patrones de los rostros derivados principalmente del patrón de los vasos sanguíneos superficiales debajo de la piel.

Apple también envía el patrón capturado del rostro a un "enclave seguro" en el dispositivo. Eso garantiza que la autenticación se realice localmente y que Apple no pueda acceder a los patrones.

Figura N° 5: Logo de biblioteca NumPy



Fuente: (<https://es.wikipedia.org/wiki/numpy>)

2.2.6.4.CMAKE:

Es una herramienta multiplataforma de generación o automatización de código. El nombre es una abreviatura para "cross platform make" (make multiplataforma); más allá del uso de "make" en el nombre, CMake es una suite separada y de más alto nivel que el sistema make común de Unix, siendo similar al auto tools.

CMake es un software libre y de código CMake es una herramienta multiplataforma de generación o automatización de código. El nombre es una abreviatura para "cross platform make" (make multiplataforma); más allá del uso de "make" en el nombre, CMake es una suite separada y de más alto nivel que el sistema make común de Unix, siendo similar a las autotools.

CMake es un software libre y de código abierto multiplataforma para gestionar la automatización de la construcción del software utilizando un método independiente del compilador. Soporta jerarquías de directorios y aplicaciones que dependen de múltiples bibliotecas. Se utiliza en conjunción con entornos de construcción nativos como Make, Qt Creator, Ninja, Xcode de Apple, y Microsoft Visual Studio. Tiene dependencias mínimas, requiriendo sólo un compilador C++ en su propio sistema de construcción. o abierto multiplataforma para gestionar la automatización de la construcción del software utilizando un método independiente del compilador. Soporta jerarquías de directorios y aplicaciones que dependen de múltiples bibliotecas. Se utiliza en conjunción con entornos de construcción nativos como Make, Qt Creator, Ninja, Xcode de Apple, y Microsoft Visual Studio. Tiene dependencias mínimas, requiriendo sólo un compilador C++ en su propio sistema de construcción.

Figura N° 6: Logo de Software Multiplataforma CMake



Fuente: (<https://cmake.org/download/>)

CAPITULO III

MATERIALES Y MÉTODOS

3.1.MATERIALES

3.1.1. Hardware

3.1.1.1.Ordenador Portátil

CARACTERÍSTICAS:

- Modelo: HP Star Wars edición especial
- Procesador: Intel Core i5-6200U 2.3 Ghz
- RAM: 6 GB
- Tipo de sistema: Sistema Operativo de 64 bits Windows 10

Figura N° 7: Ordenador Portátil.



Fuente: (<https://support.hp.com/py-es/product/star-wars-special-edition-15-an000-notebook/9033381>)

3.1.1.2.Ordenador Arduino1.

- Microcontrolador: ATmega328P
- Voltaje de funcionamiento: 5V

- Voltaje de entrada (recomendado): 7-12V
- Voltaje de entrada (limite): 6-20V
- Pins I/O digitales: 14 (6 con salida PWM)
- Pins I/O digitales PWM: 6
- Pins de entrada analógica: 6
- Memoria Flash: 32 KB (ATmega328P) de los cuales 0.5 KB son para el bootloader
- SRAM: 2 KB (ATmega328P)
- EEPROM: 1 KB (ATmega328P)
- Velocidad de reloj: 16 MHz
- Longitud: 68.6 mm
- Anchura: 53.4 mm
- Peso: 25 g.

Figura N° 8: Arduino Uno



Fuente: (<https://paraarduino.net/modelos-de-placas-arduino#arduinounor3>)

3.1.1.3. Cámara NEST CAM IQ.

- Tamaño: 12,4 cm alto, 7,4 cm largo, 7,4 cm profundidad PESO 357 g.
- Cámara: Sensor de 1/2,5" y 8 megapíxeles (4K), zoom digital 12X, HDR.
- Campo De Visión 130°.
- Visión Nocturna: LED infrarrojos de alta potencia (940nm).

- Vídeo: Hasta 1080p con 30fps.
- Audio: Altavoz y 3 micrófonos.
- Conectividad: Wi-Fi 802.11a/b/g/n/ac (2,4 GHz o 5GHz), Bluetooth de bajo consumo (BLE).

Figura N° 9: Cámara IQ



Fuente: (<https://www.xataka.com/otros-dispositivos/nest-cam-iq-analisis-el-ojo-que-todo-lo-ve-y-al-que-no-le-gustan-los-gatos>)

3.1.1.1.4 Cable USB Arduino Uno R3

Descripción

Cable USB para Arduino tipo A – B. Incluye conector macho tipo “A” en un extremo y tipo “B” en el otro.

Este cable te permitirá conectar tu tarjeta de desarrollo a la computadora para descargar programas y/o realizar la comunicación serial. Es especialmente útil para usarse cuando desarrollas programas en computadoras portátiles pues solo tiene 20 cm de longitud, lo que evita que utilice espacio en el área de trabajo.

Este cable es ideal para utilizarse con las siguientes tarjetas:

- Arduino Uno R3.
- Arduino Mega 2560.

- Tarjeta genérica compatible con Arduino Uno R3.
- Tarjeta genérica compatible con Arduino Mega 2560.
- Cualquier tarjeta de desarrollo que tenga un conector tipo B hembra.

Figura N° 10: Cable periférico para arduino uno



Fuente: (<https://www.hifisac.com/shop/product/ard-cable-cable-corto-usb-para-arduino-uno-y-mega-2521#attr=>).

3.1.2. Software

3.1.1.2.1 Microsoft Visual Studio.

Última versión estable: 2019 versión 16.10.3 (info) 29 de junio de 2021 (18 días).

Dirección web: <https://visualstudio.microsoft.com/es/>.

Microsoft Visual Studio es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para Windows y macOS. Es compatible con múltiples lenguajes de programación, tales como C++, C#, Visual Basic .NET, F#, Java, PYTHON, Ruby y PHP, al igual que entornos de desarrollo web, como ASP.NET MVC, Django, etc., a lo cual hay que sumarle las nuevas capacidades en línea bajo Windows Azure en forma del editor Mónaco.

Visual Studio permite a los desarrolladores crear sitios y aplicaciones web, así como servicios web en cualquier entorno compatible con la plataforma .NET (a partir de



la versión .NET 2002). Así, se pueden crear aplicaciones que se comuniquen entre estaciones de trabajo, páginas web, dispositivos móviles, dispositivos embebidos y videoconsolas, entre otros. (Studio, 2021)

3.1.1.2.2 PYTHON.

Sistema de tipos: Fuertemente tipado, dinámico.

Paradigma: Multiparadigma: orientado a objetos, imperativo, funcional, reflexivo.

Última versión estable: 3.9.6 (28 de junio de 2021 (15 días))

Diseñador: Guido van Rossum.

Influido por: ABC, ALGOL 68, C, Haskell, Icon, Lisp, Modula-3, Perl, Smalltalk, Java.

Implementations': CPYTHON, IronPYTHON, Jython, PYTHON for S60, PyPy, ActivePYTHON, Unladen Swallow.

PYTHON es un lenguaje de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código. Se trata de un lenguaje de programación multi paradigma, ya que soporta parcialmente la orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, dinámico y multiplataforma.

Es administrado por la PYTHON Software Foundation. Posee una licencia de código abierto, denominada PYTHON Software Foundation License. (Python, 2021).

3.1.1.2.3 OpenCV|

Licencia: BSD

Tipo de programa: software libre y de código abierto.



Programado en: C++

Desarrollador: Intel Corporation.

Lanzamiento inicial: junio de 2000.

OpenCV es una biblioteca libre de visión artificial originalmente desarrollada por Intel. OpenCV significa Open Computer Vision (Visión Artificial Abierta). Desde que apareció su primera versión alfa en el mes de enero de 1999, se ha utilizado en una gran cantidad de aplicaciones, y hasta 2020 se la sigue mencionando como la biblioteca más popular de visión artificial.¹ Detección de movimiento, reconocimiento de objetos, reconstrucción 3D a partir de imágenes, son sólo algunos ejemplos de aplicaciones de OpenCV.

Su popularidad se debe a que es:

- Libre, publicada bajo licencia BSD, que permite que sea usada libremente para propósitos comerciales y de investigación.
- Multiplataforma, para los sistemas operativos GNU/Linux, Mac OS X, Windows y Android, y para diversas arquitecturas de hardware como x86, x64 (PC), ARM (celulares y Raspberry Pi).
- Documentada y explicada: la organización tiene una preocupación activa de mantener la documentación de referencia para desarrolladores lo más completa y actualizada posible, ejemplos de uso de sus funciones y tutoriales accesibles al público no iniciado en visión artificial, además de difundir y fomentar libros y sitios de formación.

El proyecto pretende proporcionar un entorno de desarrollo fácil de utilizar y altamente eficiente. Esto se ha logrado realizando su programación en código C y C++ optimizados, aprovechando además las capacidades que proveen los procesadores multi



núcleo. OpenCV puede además utilizar el sistema de primitivas de rendimiento integradas de Intel, un conjunto de rutinas de bajo nivel específicas para procesadores Intel (OpenCV, 2021).

3.2. MÉTODO

3.2.1. Diseño de investigación

La presente investigación es considerada de diseño experimental. (Monje, 2016) Indica que la investigación experimental es cualquier investigación realizada con un enfoque científico, donde un conjunto de variables se mantienen constantes, mientras que el otro conjunto de variables se miden como sujeto del experimento. La investigación experimental es uno de los métodos de investigación cuantitativa principales.

En el área de la investigación tecnológica son habituales los estudios que describen la frecuencia de presentaciones de error y fallas en las pruebas que se realiza al crea un software, para evitar cometer el mismo error continuo se tiene que hacer un estudio descriptivo antes de dar con el problema.

Estos diseños llegan a incluir una o más variables independientes y una o más dependientes. Asimismo, pueden utilizar prepruebas y pospruebas para analizar la evolución de los grupos antes y después del tratamiento experimental. Desde luego, no todos los diseños experimentales utilizan preprueba; aunque la posprueba sí es necesaria para determinar los efectos de las condiciones experimentales (Sampieri, Fernades Collazo, & Pilar Baptista, 2014)

3.2.2. Nivel de la investigación.

La investigación explicativa se encarga de buscar el porqué de los hechos mediante el establecimiento de relaciones causa-efecto. En este sentido, los estudios explicativos pueden ocuparse tanto de la determinación de las causas (investigación post



facto), como de los efectos (investigación experimental), mediante la prueba de hipótesis. Sus resultados y conclusiones constituyen el nivel más profundo de conocimientos (Fidias G, 2016).

El nivel de investigación se refiere a la profundidad del conocimiento que se busca lograr con la investigación, por tanto, busca la generación de conocimiento con aplicación directa a los problemas de la sociedad o el sector productivo. Esta se basa fundamentalmente en los hallazgos tecnológicos de la investigación básica, ocupándose del proceso de enlace entre la teoría y el producto.

La presente investigación siendo un prototipo busca abrir un camino para un nuevo método de diseñar un prototipo de sistema de reconocimiento facial para el control biométrico en el colegio Aplicación de la Universidad Nacional del Altiplano.

3.2.3. Población de la investigación.

La población de investigación no se refiere siempre a la definición de tipo demográfico: conjunto de personas ubicadas en una área geográfica determinada, claramente definida (Monje, 2016).

La población de la investigación son 15 personas que participaron voluntarios en el reconocimiento facial, para que posteriormente se aplique en el colegio Aplicación de la Universidad Nacional del Altiplano Puno.

3.2.4. Ámbitos de estudio investigación.

El ámbito de estudio se refiere a la descripción de la población de estudio en la que se va realizar.

De acuerdo a (Sabino, 1996). La población de estudio (muestra) estuvo dada por un grupo de participantes seleccionados aleatoriamente, con lo que se realizó la prueba.

En tal sentido se aplicará como prueba en 15 personas que participan de manera voluntaria, para determinar la eficiencia del software, para posteriormente aplicarse en el colegio Aplicación de la Universidad Nacional del Altiplano Puno

3.2.4.1. Ubicación de la investigación.

La investigación se desarrollará en el distrito de Puno de la provincia de Puno del departamento de Puno para colegio Aplicación de la Universidad Nacional del Altiplano Puno, el cual se encuentra en el interior de la Universidad Nacional Del Altiplano Puno.

Figura N° 11: Vista satelital de colegio de Aplicación de la Universidad Nacional del Altiplano



Fuente: Google maps

3.2.5. Recolección de datos

3.2.5.1. Técnicas

Las técnicas de investigación comprenden un conjunto de procedimientos organizados sistemáticamente que orientan al investigador en la tarea de profundizar en el conocimiento y en el planteamiento de nuevas líneas de investigación. Pueden ser

utilizadas en cualquier rama del conocimiento que busque la lógica y la comprensión del conocimiento científico de los hechos y acontecimientos que nos rodean (Maya, 2014).

3.2.5.2. Instrumentos

Los instrumentos de acopio de datos se denominan a todos los instrumentos que pueden servir para medir las variables, recopilar información con respecto a ellas o simplemente observar su comportamiento (Mejía, 2005).

3.2.6. Contratación de hipótesis

Comprobar una hipótesis significa someterla a contrastación de una realidad. Es decir, el investigador tiene que someter a prueba aquello que ha enunciado en su hipótesis, y para ello ha de establecer, mediante alguna técnica de contrastación si su hipótesis concuerda o no con los datos empíricos. En una prueba de ajuste la hipótesis nula establece que una variable X tiene una cierta distribución de probabilidad con unos determinados valores de los parámetros. El tipo de distribución se determina, según los casos, en función de: La propia definición de la variable, consideraciones teóricas al margen de esta y/o evidencia aportada por datos anteriores al experimento actual. (Espinoza, 2018).

Hipótesis nula: X tiene distribución de probabilidad $f(x)$ con parámetros.

Hipótesis alternativa: X tiene cualquier otra distribución de probabilidad.

3.2.6.1.1. Descripción general del sistema

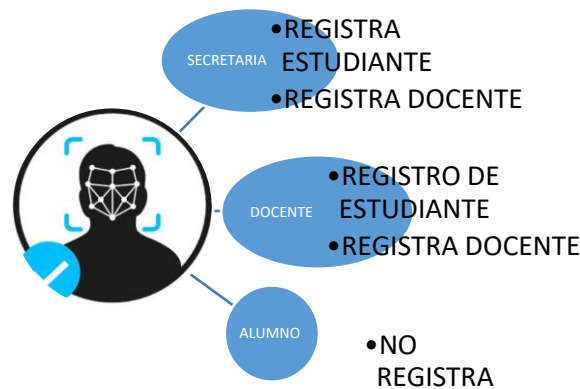
El sistema trabajado, tiene como propósito la verificación de la asistencia de los participantes utilizados en el método de reconocimiento facial; para tener la información integral del sistema al alcance, se describe también las restricciones del sistema y la información en ámbito general, factores supuestos y dependencias que pueden afectar la

elaboración del sistema, además de eso se debe de especificar los requisitos mínimos a contemplar para que el sistema satisfaga un reconocimiento adecuado.

3.2.6.1.2. Prototipo de control de asistencia estudiantil

Será diseñado con una interface visual para que su diseño sea amable para el usuario lo que permitirá el registro de personas con mayor facilidad sin la necesidad de depender de programas alternos instalados.

Figura N° 12: Sistema de usuario administrado



Elaborado por el equipo de trabajo.

3.2.2 DETALLE DE USUARIO:

Tabla N° 1: Detalle de usuario 01.

USUARIO	ADMINISTRADOR DE INSTITUCIÓN.
INFORMACIÓN	conocimientos medios de informática
MANEJO	manejo computacional medio
USO DEL SISTEMA	Identificarse, control y operación del sistema.
USUARIO	secretaria
INFORMACIÓN RELEVANTE AL MANEJO	Administración de archivos y trámites educativos.
USO DEL SISTEMA	Identificarse, creación de base de asistencia de los alumnos, padres, profesores además de la gestión de recursos.

Elaborado por el equipo de trabajo.



3.2.6.1.3. Detalle de usuario

Tabla N° 2: Detalle de usuario 02

USUARIO	DOCENTE
INFORMACIÓN RELEVANTE AL MANEJO	conocimiento básico en informática, manejo de aplicaciones
USO DE SISTEMA	Identificarse, creación de base datos de los alumno manual o automático (reconocimiento facial).

Elaborado por el equipo de trabajo.

3.2.6.1.4. Restricciones del sistema

Tabla N° 3: Restricciones del sistema.

RESTRICCIÓN	CONSIDERACIÓN
LENGUAJE PYTHON Y TECNOLOGÍA JAVA	diseño del sistema cliente servidor
DIFICULTAD PARA EL RECONOCIMIENTO	el sistema debe ser desarrollado de una manera sencilla y de fácil uso por parte del área usuaria
HERRAMIENTA CLOUD	el uso del sistema debe ser restringido para personas ajenas a la administración de la entidad educativa

Elaborado por el equipo de trabajo.

3.2.6.1.5. Requisitos futuros

En el desarrollo futuro se tiene considerado la implementación y desarrollo de una aplicación móvil para cada docente, en la cual se incluya el proceso de suscripción de asistencia mediante el uso de reconocimiento facial así proponer el fácil uso del sistema con el hardware de los dispositivos móviles y así darle una segunda opción en el caso de que no se contara con una computadora o laptop de escritorio en salón donde se desarrollan las sesiones de clase.

3.2.6.2. Factibilidad

3.2.6.2.1. Factibilidad humana



El personal administrativo cuenta con las capacidades informáticas y manejo de internet, por lo tanto, el uso y manejo del sistema no se convertirá en un problema a futuro además se ver la disposición de un manual online donde se verá un resumen del manejo del sistema.

3.2.6.2.2. Factibilidad legal

La entidad educativa, siempre dispone de sesiones virtuales y está en constante evolución tecnológica, por lo cual el sistema que se utilizara entregara un valor agregado a la integridad de procesos educativos que se realiza por el personal docente y no existe restricción legal que impide su realización.

3.2.6.2.3. Factibilidad de sistema

Para llevar el desarrollo e implementación del sistema, es necesario realizar el estudio de viabilidad del sistema, lo cual vera la capacidad técnica, operativa, económica, tecnológica y operacional, con el propósito de conocer si es o no factible realizar el desarrollo.

3.2.6.3. Técnica:

Las técnicas necesarias para el desarrollo e implementación para la aplicación.

- Programación orientada a objetos
- Lenguaje de programación PYTHON
- Manejo de visual studio
- MYSQL
- Conocimientos en cloud computing
- Manejo de cogntive services
- APIs de herramientas (Microsoft services)

3.2.6.4. Operatividad

El sistema será designado para el uso y operación de los administradores de la entidad educativa. El manejo será sencillo e intuitivo, logrando la comprensión relativamente rápida y así poder cubrir las expectativas del manejo.

3.2.6.5. Tecnológica

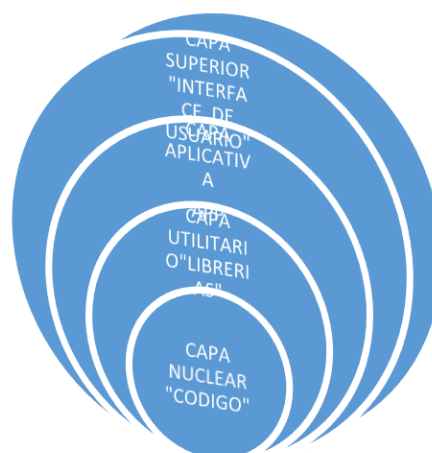
Realizado el análisis de la tecnología utilizar para el desarrollo del sistema de reconocimiento facial se realizó la clasificación enfocada en dos partes hardware y software

- 01 computador de escritorio con las características siguientes
 - Procesador Intel core i7 – 64bits
 - Velocidad de 2.4Ghz
 - Memoria RAM: 6 GB
 - DISCO: 1 TB

3.2.6.6. Elaboración

3.2.6.6.1. Entorno de software

Figura N° 13: Capas del sistema.



Elaborado por el equipo de trabajo.



A de recalcar que el colegio Aplicación de la Universidad Nacional del Altiplano con ubicación en región de Puno provincia Puno distrito de Puno específicamente en la Av. Floral. N° 1153, Puno 21001, el sistema de reconocimiento facial a implementar estará siendo desarrollado en el lenguaje PYTHON, y desarrollado en el software Visual Studio y los requisitos que son necesarios para su funcionamiento. Son la conexión a internet y los apikeys de cognitive services, así mismo el equipo de programación debe de tener una mejor comunicación con el personal que se involucrara de la institución (director, docente, secretaria, estudiante, otros). Para realizar la agrupación de requisitos necesarios y evitar los problemas de desarrollo y validación del sistema desarrollado.

Cada una de las capas ejecuta operaciones diferentes que se semejan continuamente al conjunto de instrucciones de la maquina (sistema operativo).

En la capa externa el componente realiza el trabajo de la interfaz del usuario. En la interna los componentes trabajan la interfaz con el sistema operativo. Las capas que se tienen en la parte intermedia entregan servicios utilitarios y funciones de software para el sistema.

Ingeniería de software, Pressman 2010.

3.2.6.6.2. MYSQL

Es un sistema de base de datos SQL de código libre, desarrollado y con respaldo de Oracle corporation. El sistema permite la creación de base de datos que funcionan con sistemas cliente /servidor e integrados además así mismo tiene funciones multiplataforma (mysql, 2018).

3.2.6.6.3. Microsoft azure

Microsoft Azure (anteriormente Windows Azure y Azure Services Platform) es un servicio de computación en la nube creado por Microsoft para construir, probar,



desplegar y administrar aplicaciones y servicios mediante el uso de sus centros de datos. Proporciona software como servicio (SaaS), plataforma como servicio (PaaS) e infraestructura como servicio (IaaS) y es compatible con muchos lenguajes, herramientas y marcos de programación diferentes, incluidos software y sistemas específicos de Microsoft y de terceros (Conde, 2017).

3.2.6.6.4. Cognitive services

Los servicios cognitivos de Microsoft Azure son un conjunto de API, SDK y servicios personalizables prediseñados que están disponibles para los desarrolladores para hacer que sus aplicaciones sean más inteligentes, atractivas y detectables. Los servicios incluyen inteligencia perceptiva y cognitiva que cubre reconocimiento de voz, reconocimiento de locutor, síntesis neuronal de voz, reconocimiento facial, visión por computadora, OCR / comprensión de formas, procesamiento de lenguaje natural, traducción automática y servicios de decisiones comerciales. La mayoría de las funciones de IA que aparecieron en los propios productos y servicios de Microsoft (Bing, Office, Teams, Xbox y Windows) funcionan con Azure Cognitive Services.

3.2.6.6.5. Servidor cognitive services

En el portal Microsoft Azure existen diferentes servicios de infraestructura y de plataforma para que puedas “montar” los servicios que necesites de manera sencilla, con unos cuantos clics. En tu portal dispones de un botón [New +] y a partir de ahí, sólo tienes que elegir la región de los Datacenter donde estará tu servicio y, a continuación, el tipo de servicio con sus características. Por ejemplo, en el caso de querer crear una máquina virtual podrías seleccionar el tipo de máquina como Windows Server 2016 Datacenter, a continuación, rellenar todas sus características (espacio de disco, RAM...). En este proceso de creación, se define un nombre de usuario y contraseña específica para poder iniciar sesión en esta máquina virtual.

Entre los servicios, dispones de infraestructuras (IaaS: almacenamiento, redes, máquinas virtuales...) y plataformas (PaaS: bases de datos de alta disponibilidad SQL, CMS para desarrollo de web, backend para aplicaciones móviles...). Son compatibles con todo tipo de tecnología: bases de datos Oracle, Linux, php, iOS, My SQL, Android, php.

Estos servicios están garantizados con una disponibilidad del 99.99%, y en caso de fallo en disponibilidad superior, Microsoft se compromete a indemnizar por los daños. Además, cuenta con todas las certificaciones en materia seguridad y protección de datos.

3.2.6.6. Servicios de cloud cognitive

Azure es una nube pública de pago por uso que te permite compilar, implementar y administrar rápidamente aplicaciones en una red global de data centers (centros de datos) de Microsoft Pero atención, “nube pública” no quiere decir que cualquiera pueda acceder a la información de tu negocio, sino que cualquiera con conexión a Internet puede hacer uso de esta nube y que sólo en caso de disponer de las credenciales oportunas, podrá acceder a tu portal de Azure, donde se generan y administran todos los servicios.

3.2.6.7.Arquitectura

Figura N° 14: Arquitectura.



Elaborado por el equipo de trabajo.

Arquitectura del sistema, describe cuatro niveles para el funcionamiento integral del sistema.

3.2.6.8. Proceso de funcionamiento del prototipo

En los siguientes párrafos se representará el proceso de funcionamiento del software para mayor comprensión, dichas ilustraciones se dividirán en etapas y cada uno de ellos se sub divide en dos subtítulos principales descritos para sus generalidades de cada una de sus faces

Figura N° 15: Funcionamiento del sistema



Elaborado por el equipo de trabajo.

3.2.6.8.1. Base de datos

En la siguiente imagen se muestra el código realizado para el proceso de creación de la base de datos del registro realizado por el usuario del sistema.

Figura N° 16: Código de base de datos.

```
01 caso_de_uso_crear_base_de_datos.py > ...  
1 from models import *  
2  
3 # se conecta  
4 db.connect()  
5  
6 # se define las clases de tablas  
7 tables = [Student]  
8  
9 # se crea las tablas  
10 db.create_tables(tables)  
11  
12 # se cierra la base de datos  
13 db.close()  
14
```

Elaborado por el equipo de trabajo.

La base de datos esta creada a partir de la combinación de la librería numpy ya que esta es la encargada de realizar el llamado a los datos para poder verificarlos en formato de tabla.

3.2.6.8.2. Reinicio de base de datos

En la siguiente imagen se muestra el código realizado para el proceso de reinicio de base de datos, ingresados en el anterior paso para realizar una eliminación de los registros ingresados el código es realizado para prevenir el ingreso de datos repetidos y/o realizar la instalación en otro equipo computador sin trasladar los datos anteriormente usados.

Figura N° 17: Código reinicio de sistema

```
delete-pr.py > ...
9  import cognitive_face as CF
10
11  from models import *
12  from global_variables import *
13
14
15
16  CF.Key.set(cfKey)
17  CF.BaseUrl.set(URL)
18
19  # elimina el grupo de personas
20  def eliminarPersonGroup(personGr):
21      CF.person_group.delete(personGr)
22      print("eliminado el person group")
23
24  eliminarPersonGroup(personGroupId)
```

Elaborado por el equipo de trabajo.

3.2.6.8.3. Formulario

En la imagen se muestra el formulario de ingreso de datos creado para el registro de los datos de cada estudiante de quien se desea registrar visualmente para llevar el control de asistencia de los mismos.

Figura N° 18: Formulario de datos estudiante

The image shows a software window titled 'Registro de persona' with a subtitle 'Formulario de registro'. It contains three input fields: 'Nombres', 'Apellidos', and 'Codigo'. Below these fields is a button labeled 'Guardar datos'. The window has standard Windows-style controls (minimize, maximize, close) in the top right corner.

Elaborado por el equipo de trabajo.

3.2.6.8.4. Confirmación:

Ventana de confirmación de ingreso de datos hacia el software para el posterior almacenaje en la base de datos, se debe de tener especial cuidado para no ingresar datos erróneos ya que el sistema no contempla la modificación de los datos previamente ingresados.

Figura N° 19: Ventana de confirmación

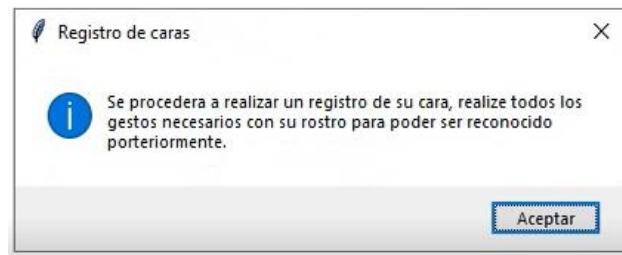
The image shows a confirmation dialog box titled 'Datos a guardar'. It features an information icon (i) on the left. The text inside the dialog reads: 'Nombres : fsjahfashjf', 'Apellidos : ssssssssss', and 'Codigo : 2125876'. At the bottom right, there is a button labeled 'Aceptar'. The dialog has a close button (X) in the top right corner.

Elaborado por el equipo de trabajo.

3.2.6.8.5. Ventana confirmación de registro

En la imagen se verifica el Inicio de toma de muestras con el mensaje “se procederá a realizar un registro de su cara, realizar todos los gestos necesarios con su rostro para poder ser reconocido posteriormente” el mensaje es claro ya que cada muestra tomada será usada como referencia

Figura N° 20: Ventana de confirmación e inicio

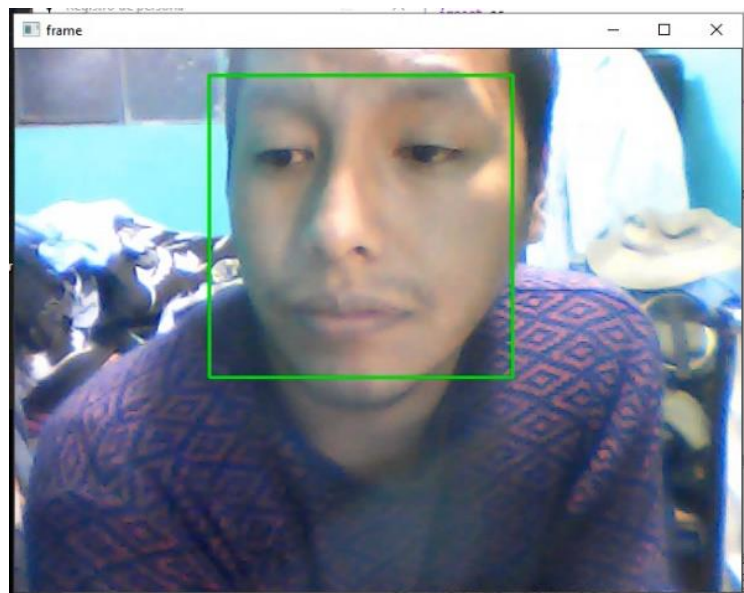


Elaborado por el equipo de trabajo.

3.2.6.8.6. Inicio de proceso de toma de muestra

Mostrada la confirmación e iniciado el proceso de toma de muestra para él envió al servidor, esta ventana está diseñada para mostrarnos frente a la cámara y podamos realizar algunos gestos de diferentes tipos y así tener una base de datos con la variedad de los mismos, consecutivamente se debe de colocar y/o quitar los accesorios utilizados en el rostro sean del tipo aretes, lentes, gorras, todos estos accesorios influirán en el momento del reconocimiento facial una vez iniciado.

Figura N° 21: Toma de muestras.



Elaborado por autores

3.2.6.8.7. Toma de imágenes

En el proceso de toma de muestras se visualiza el envío de cada imagen tomada por la cámara instalada en el computador, como se visualiza en la imagen de la parte inferior. Cada muestra esta nombrada “IMAGE NUM # en el proceso de envío de imagen se codificará cada muestra por el software para su respectiva identificación en servidores de Microsoft Azure

Figura N° 22: Toma de muestras

```
image num 5
image num 6
image num 7
image num 8
image num 9
image num 10
[ WARN:1] global C:\Users\apoveyor\AppData\Local\Temp\1\pip-req-build-kh7iq4w7\opencv\modules\videoio\src\cap_msmf.cpp (434) `anonymous-namespace':::SourceReaderCB::~SourceReaderCB terminating async callback
{'personId': '9e765f47-3b19-4d19-9fb7-05ff2fa18abf'}
person--9bb8ba86-d441-429f-9e45-aca9c9d07647--1.jpg
[]
```

Elaborado por autores

3.2.6.8.8. Envío de muestras:

Este proceso inicia posteriormente a la toma de muestras, se procede a la creación de un ID para cada una de las muestras tomadas por el software para su identificación y se envía a los servidores de Microsoft Azure para su posterior procesamiento de cada una de las imágenes.

Figura N° 23: Creacion de ID

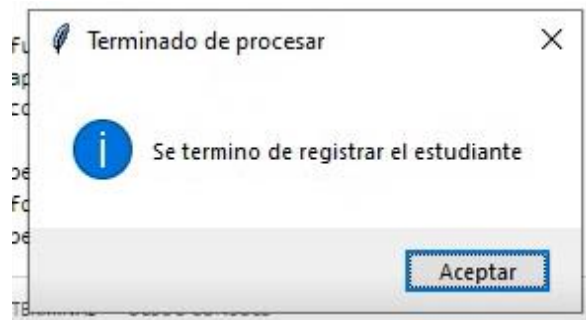
```
{'personId': '9e765f47-3b19-4d19-9fb7-05ff2fa18abf'}
person--9bb8ba86-d441-429f-9e45-aca9c9d07647--1.jpg
[{'faceId': 'd1d26508-d1b6-4671-b73b-7b9fb78209e', 'faceRectangle': {'top': 0, 'left': 0, 'width': 266, 'height': 266}}]
{'persistedFaceId': 'f6a8f7be-7567-4084-9fc9-126a0e3f210c'}
person--9bb8ba86-d441-429f-9e45-aca9c9d07647--10.jpg
[{'faceId': 'd4080f1d-a10e-4a9b-b4ec-4099152aa335', 'faceRectangle': {'top': 0, 'left': 0, 'width': 266, 'height': 266}}]
{'persistedFaceId': '18b547be-0866-4084-a7d6-38c47a90ad1f'}
person--9bb8ba86-d441-429f-9e45-aca9c9d07647--2.jpg
[{'faceId': '0bb7f9ed-a10e-4154-a43b-f00330a0a8f7', 'faceRectangle': {'top': 0, 'left': 0, 'width': 266, 'height': 266}}]
{'persistedFaceId': 'bfdf3da3-ea4c-48af-af85-99edc60730f1'}
person--9bb8ba86-d441-429f-9e45-aca9c9d07647--3.jpg
[{'faceId': '220b1a2b-ef60-4d2e-82ce-017339b714ca', 'faceRectangle': {'top': 0, 'left': 0, 'width': 266, 'height': 266}}]
{'persistedFaceId': 'a20d32f9-202c-4992-8008-9e4041629106'}
person--9bb8ba86-d441-429f-9e45-aca9c9d07647--4.jpg
[]
```

Elaborado por el equipo de trabajo.

3.2.6.8.9. Finalización de registro

La función principal de esta venta es mostrar la finalización del registro del estudiante con éxito, así como él envió de muestras a los servidores, terminado el proceso se procede a continuar el registro del resto del personal.

Figura N° 24: Confirmación de finalización.

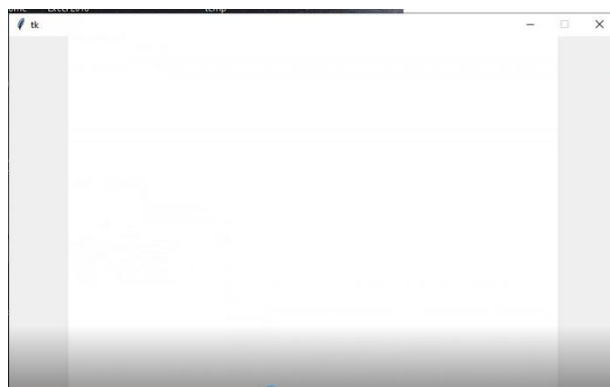


Elaborado por el equipo de trabajo.

3.2.6.8.10. Box de inicio de reconocimiento

En el proceso de reconocimiento facial se diseñó una ventana para mostrar la imagen mostrada por la cámara, antes de inicio la ventana se mostrará en blanco iniciando el código para el reconocimiento.

Figura N° 25: Inicio de reconocimiento facial.



Elaborado por el equipo de trabajo.

3.2.6.8.11. Box proceso de reconocimiento

El proceso de reconocimiento facial inicia con el recuadro verde que aparece en la imagen inferior remarcando el rostro del personal para su reconocimiento, así mismo se muestra los datos aleatorios ingresados en el inicio del registro, para lograr este proceso el software realiza una llamada a los servidores para enviar los datos ingresados.

Figura N° 26: Proceso de reconocimiento.



Elaborado por el equipo de trabajo.

3.2.6.8.12. Ventana de pruebas de reconocimiento

La ventana muestra el proceso de reconocimiento, en vivo se datos fue exitoso más las muestras tomadas para ser comparadas no fueron reconocidas por el servidor por lo tanto el software muestra el aviso siguiente “NO RECONOCIDO”

Figura N° 27: Ventana de pruebas.



Elaborado por el equipo de trabajo.

CAPITULO IV

RESULTADOS Y DISCUSIÓN

4.1.INTERPRETACIÓN DE RESULTADOS

4.1.1. Experiencia en el desarrollo de software

Los resultados obtenidos en las pruebas realizadas con el prototipo de sistema demuestran un margen de error considerable para un sistema que está en etapa de prototipo, el sistema cuenta con un aproximado de 10 segundos para el proceso de reconocimiento hasta su etapa final, los resultados muestran un porcentaje de efectividad bueno para la muestra tomada y cotejada en los servidores de Microsoft Azure.

Los resultados obtenidos en las pruebas y el funcionamiento del software se verifican en el siguiente apartado.

4.1.2. Código para la creación de base de datos

Figura N° 28: Creación base de datos

```
1 from models import *
2
3 # se conecta
4 db.connect()
5
6 # se definen las clases de tablas
7 tablas = [Student]
8
9 # se crea las tablas
10 db.create_tables(tablas)
11
12 # se cierra la base de datos
13 db.close()
14
```

PROCESS OUTPUT TERMINAL DEBUG CONSOLE

```
[WARNING] global C:\Users\lappey\on\appData\Local\Temp\1\gpo-req-bu2a-4n71qk7\openv\modules\desktop\prj\app_name_434\anonymous-namespace\150n
abebecb1--d3arvhebedc0\templating\ajmc_cdl2ack
PS C:\Users\labeled\Desktop\Fack_recogitacion_of_medio & C:\Users\labeled\AppData\Local\Programs\Python\Python38\python.exe C:\Users\labeled\Desktop\Fack_re
ogitacion_of_medio\create_pr.py
exitcode: 0, error: none
PS C:\Users\labeled\Desktop\Fack_recogitacion_of_medio & C:\Users\labeled\AppData\Local\Programs\Python\Python38\python.exe C:\Users\labeled\Desktop\Fack_re
ogitacion_of_medio\Fack_recogitacion_of_medio.py
exitcode: 0, error: none
PS C:\Users\labeled\Desktop\Fack_recogitacion_of_medio & C:\Users\labeled\AppData\Local\Programs\Python\Python38\python.exe C:\Users\labeled\Desktop\Fack_re
ogitacion_of_medio\Fack_recogitacion_of_medio.py
```

Elaborado por el equipo de trabajo.

Figura N° 29: Creación base de datos

```
1 from models import *
2
3 # se conecta
4 db.connect()
5
6 # se define las clases de tablas
7 tablas = [Student]
8
9 # se crea las tablas
10 db.create_tables(tablas)
11
12 # se cierra la base de datos
13 db.close()
14
```

Elaborado por el equipo de trabajo.

4.1.3. Eliminar grupo de personas

Esta etapa para poder realizar el inicio desde cero del sistema se debe de eliminar los grupos existentes en el software para que no existan problemas en la búsqueda de imágenes ya subidas al software o imágenes duplicadas, imágenes desenfocadas e imágenes pixeladas producidas por la baja calidad de las cámaras utilizadas o algún problema físico en la implementación del sistema.

Figura N° 30: Eliminar person group

```
9 import cognitive_face as CF
10
11 from models import *
12 from globus_variables import *
13
14
15 CF.Key.set('key')
16 CF.BaseUrl.set(URL)
17
18 # elimina el grupo de personas
19 def eliminarPersonGroup(personId):
20     CF.person_group.delete(personId)
21     print('Eliminado el person group')
22
23
24 eliminarPersonGroup(personId)
```

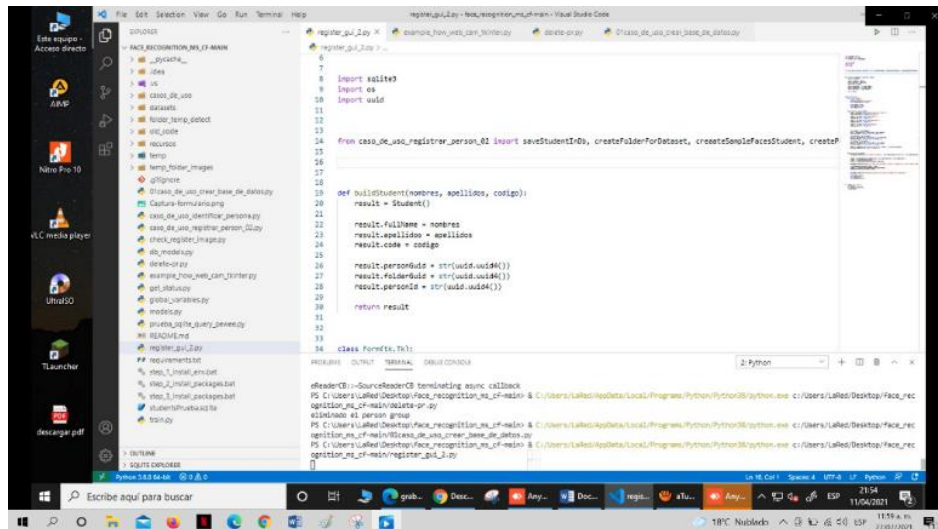
Elaborado por el equipo de trabajo.

4.1.4. Registrar persona

En esta etapa se registra a la persona ingresando los datos solicitados descritos en anexo 4.1.6. Nombre, apellidos, código de estudiante o de ser el caso código de docente

y seguidamente se presione el botón aceptar para inicia el proceso de detección de rostros en sistema en resumen se ingresan los datos requeridos y se procede a la toma de muestras

Figura N° 31: Registro.

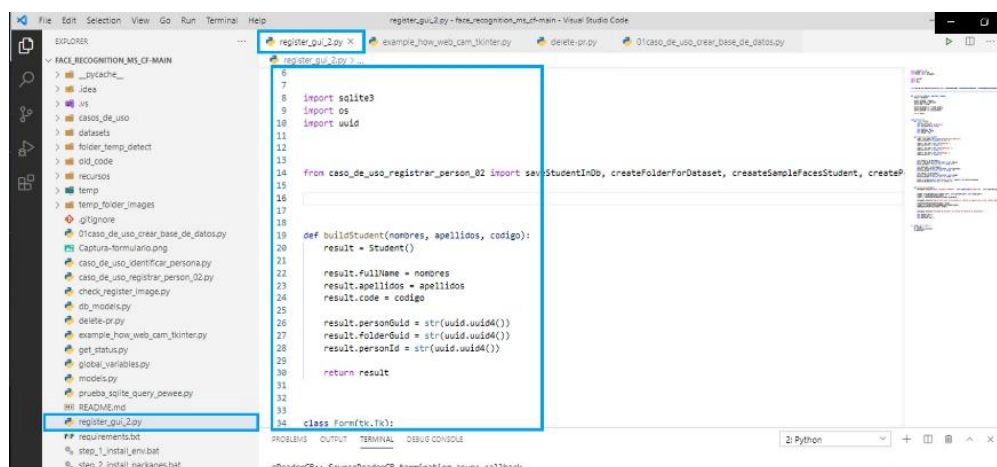


Elaborado por el equipo de trabajo.

4.1.5. Código de registro de personas

En la imagen se describe el algoritmo diseñado para el registro de personas que se emplea en las capas de programación por debajo de la interface gráfica del usuario (cliente registrador)

Figura N° 32: Código de registro

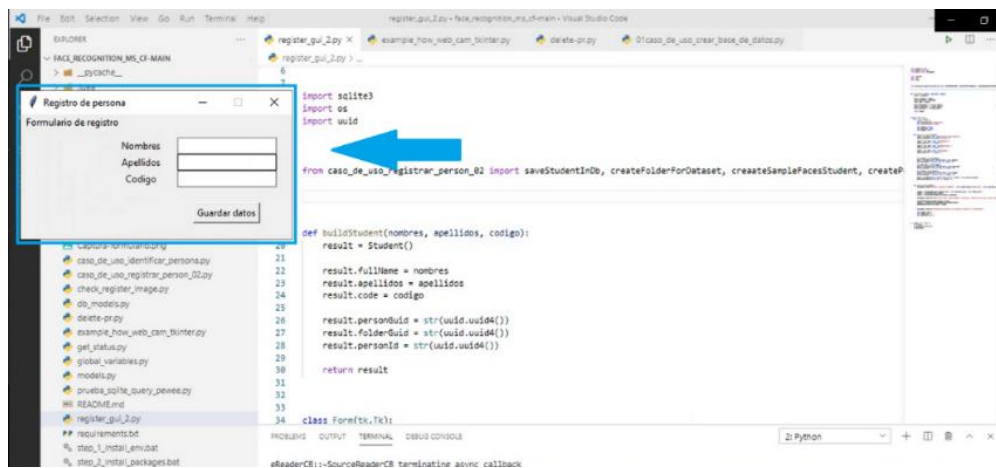


Elaborado por el equipo de trabajo.

4.1.6. Inicio de registro de personas

En esta etapa se puede lograr apreciar la ventana emergente de registro de datos al ejecutar el programa, se iniciará el programa para el ingreso de datos requeridos y la toma de muestras de imagen que se muestra en la captura de pantalla

Figura N° 33: Registro de personas

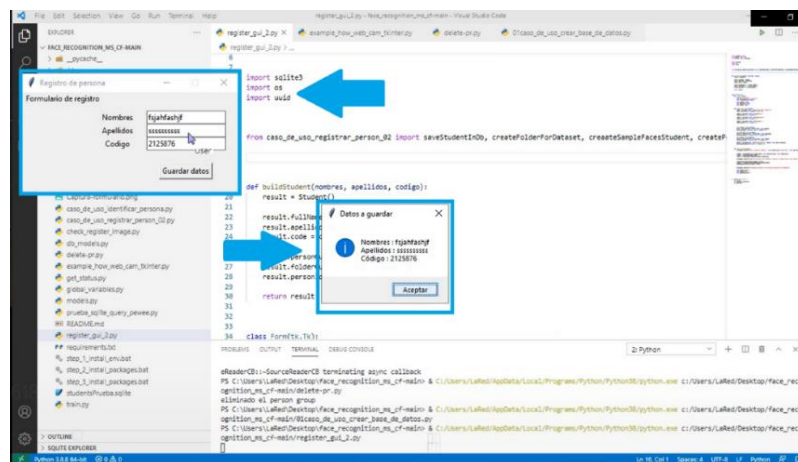


Elaborado por el equipo de trabajo.

4.1.7. Ingreso de datos para pruebas del prototipo

En esta etapa se realiza una prueba de funcionamiento del sistema ingresando datos al azar para revisar la etapa de reconocimiento, almacenamiento de datos así mismo.

Figura N° 34: Ingreso de datos ensayo de prototipo

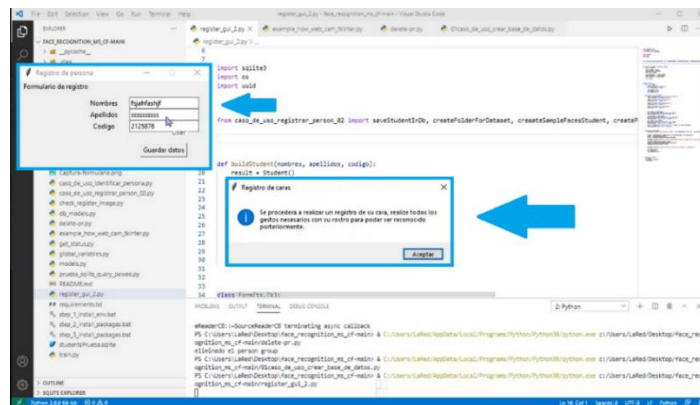


Elaborado por el equipo de trabajo.

4.1.8. Ventana de confirmación inicio de toma de muestras

Esta ventana muestra la confirmación de ingreso de datos seguidamente anuncia el inicio de la toma de muestras para su respectivo envío hacia la plataforma de Microsoft azure.

Figura N° 35: Confirmación de muestras

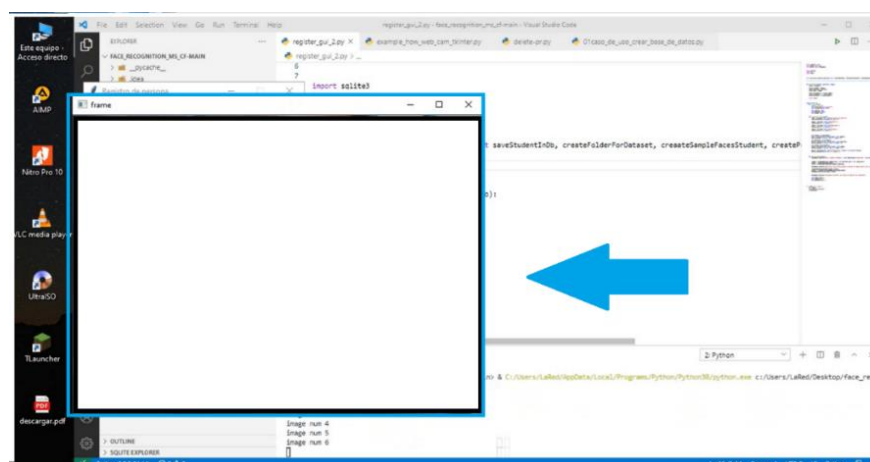


Elaborado por el equipo de trabajo.

4.1.9. Ventana emergente inicio de toma de muestras

Esta ventana tiene la finalidad de realizar la toma de muestras del rostro del estudiante con la cámara de video asignada para el sistema, paralelamente en la toma de muestras aparece el recuadro con márgenes verdes donde se seleccionará el rostro de cada estudiante.

Figura N° 36: Ventana de inicio

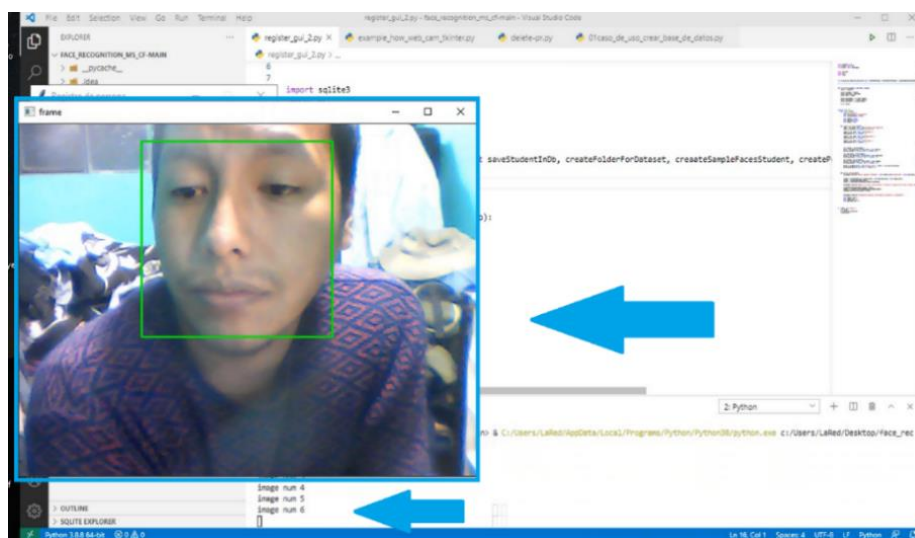


Elaborado por el equipo de trabajo.

4.1.10. Toma de muestras de imagen

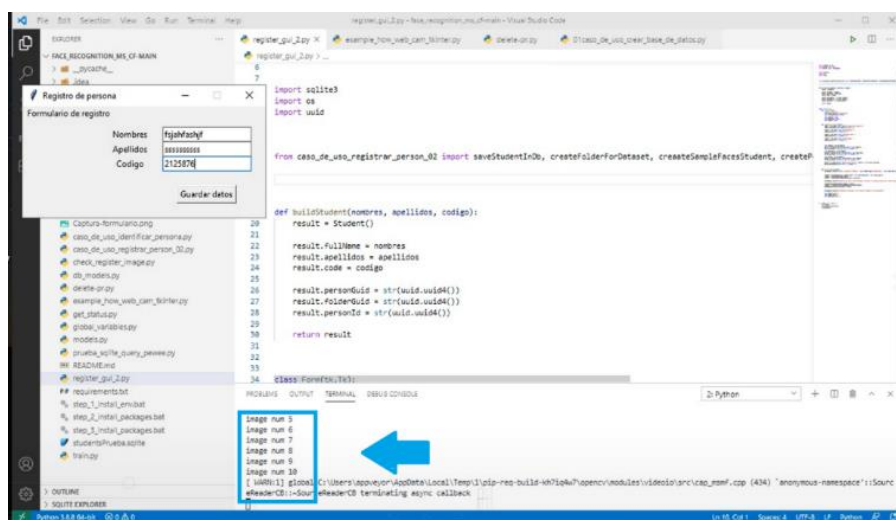
Se toma las muestras del video tomado por la cámara en la parte, inferior del panel, se muestra las muestras tomadas por el software, unos máximos de 10 muestras serán tomadas en cada registro de estudiante y luego serán enviadas para su procesamiento en los servidores de Microsoft azure, la api del mismo (cognitive services) será la llave para el ingreso a los servidores y su entrenamiento en el mismo.

Figura N° 37: Captura de muestras



Elaborado por el equipo de trabajo.

Figura N° 38: Toma de muestras captadas



Elaborado por el equipo de trabajo.

4.1.11. Finalización de toma de muestra

En la imagen muestra el final de la toma de muestra, se muestra que la ventana emergente desaparece para su envío a los servidores a utilizar.

Figura N° 39: Toma de muestras

```
def buildStudent(nombres, apellidos, codigo):  
    result = Student()  
    result.fullName = nombres  
    result.apellidos = apellidos  
    result.code = codigo  
    result.personUuid = str(uuid.uuid4())  
    result.folderUuid = str(uuid.uuid4())  
    result.personId = str(uuid.uuid4())
```

```
Image num 5  
Image num 6  
Image num 7  
Image num 8  
Image num 9  
Image num 10  
[ VMRn:1] global C:\Users\vaspovejor\AppData\Local\Temp\1\pip-req-bu...h72a47\opencl\modules\video\src\caso_menf.cpp (434) 'anonymous-namespace': Sourc  
eHeaderCB::~SourceHeaderCB terminating async callback  
{ "personId": "9a076547-3c21-4d33-9f07-05f762d3e0f7",  
  "person": "9028086-044-420f-9a45-acaf63087647-1.jpg",  
  "faceId": "e2d26006-0206-4671-673b-76f9b762209e", "faceRectangle": {"top": 0, "left": 0, "width": 266, "height": 266}}
```

Elaborado por el equipo de trabajo.

4.1.12. Reenvío de imágenes tomadas

En esta etapa se inicia el envío de imágenes muestreadas por el programa para su procesamiento en los servidores del mismo.

Figura N° 40: Reenvío de muestras

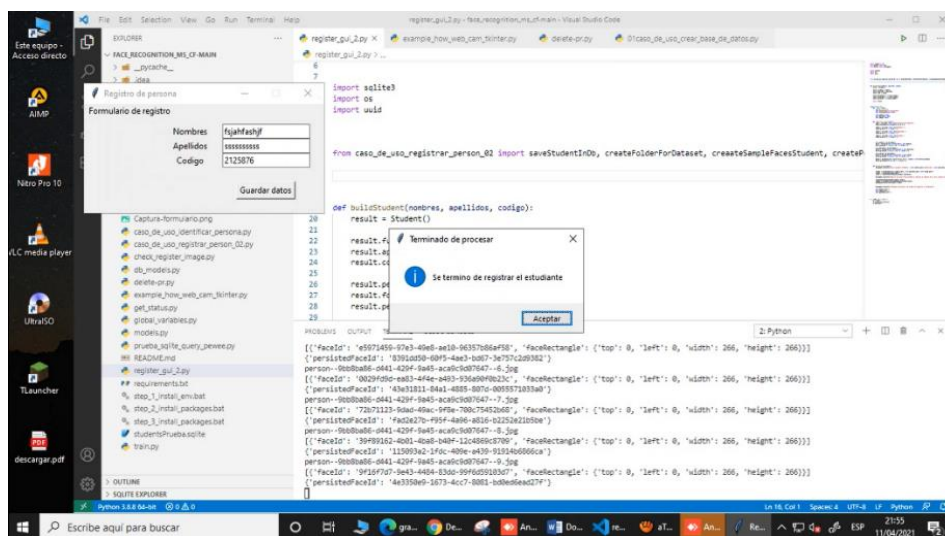
```
Image num 10  
[ VMRn:1] global C:\Users\vaspovejor\AppData\Local\Temp\1\pip-req-bu...h72a47\opencl\modules\video\src\caso_menf.cpp (434) 'anonymous-namespace': Sourc  
eHeaderCB::~SourceHeaderCB terminating async callback  
personId": "9a076547-3c21-4d33-9f07-05f762d3e0f7",  
  person": "9028086-044-420f-9a45-acaf63087647-1.jpg",  
  faceId": "e2d26006-0206-4671-673b-76f9b762209e", "faceRectangle": {"top": 0, "left": 0, "width": 266, "height": 266}}  
Image num 11  
[ VMRn:1] global C:\Users\vaspovejor\AppData\Local\Temp\1\pip-req-bu...h72a47\opencl\modules\video\src\caso_menf.cpp (434) 'anonymous-namespace': Sourc  
eHeaderCB::~SourceHeaderCB terminating async callback  
personId": "9a076547-3c21-4d33-9f07-05f762d3e0f7",  
  person": "9028086-044-420f-9a45-acaf63087647-2.jpg",  
  faceId": "5803071d-e10e-440b-84ec-4809153aa335", "faceRectangle": {"top": 0, "left": 0, "width": 266, "height": 266}}  
Image num 12  
[ VMRn:1] global C:\Users\vaspovejor\AppData\Local\Temp\1\pip-req-bu...h72a47\opencl\modules\video\src\caso_menf.cpp (434) 'anonymous-namespace': Sourc  
eHeaderCB::~SourceHeaderCB terminating async callback  
personId": "9a076547-3c21-4d33-9f07-05f762d3e0f7",  
  person": "9028086-044-420f-9a45-acaf63087647-3.jpg",  
  faceId": "90d7f9e0-a31e-4124-8410-f003380a6677", "faceRectangle": {"top": 0, "left": 0, "width": 266, "height": 266}}  
Image num 13  
[ VMRn:1] global C:\Users\vaspovejor\AppData\Local\Temp\1\pip-req-bu...h72a47\opencl\modules\video\src\caso_menf.cpp (434) 'anonymous-namespace': Sourc  
eHeaderCB::~SourceHeaderCB terminating async callback  
personId": "9a076547-3c21-4d33-9f07-05f762d3e0f7",  
  person": "9028086-044-420f-9a45-acaf63087647-4.jpg",  
  faceId": "90d7f9e0-a31e-4124-8410-f003380a6677", "faceRectangle": {"top": 0, "left": 0, "width": 266, "height": 266}}
```

Elaborado por el equipo de trabajo.

4.1.13. Finalización de envío de muestras a la app azure microsoft:

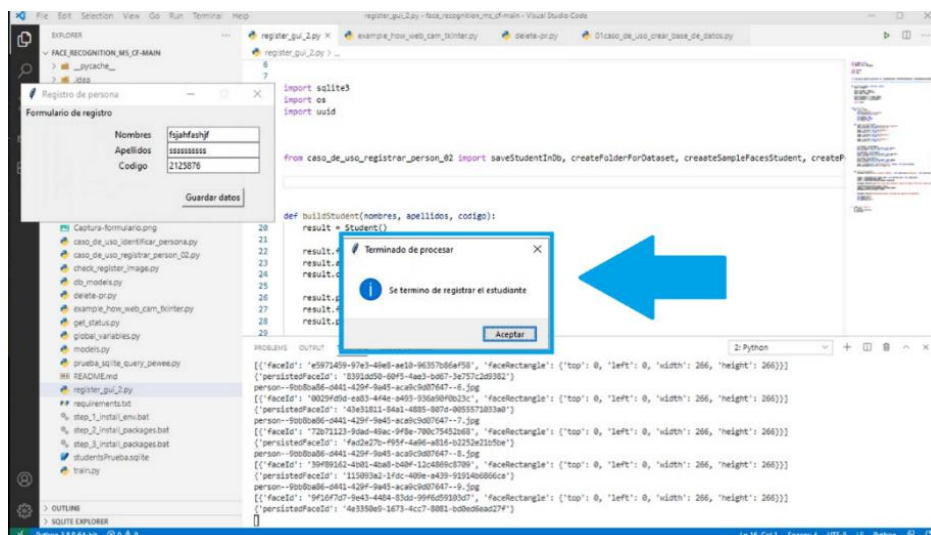
En la imagen se muestra la ventana emergente de confirmación que indica que ha finalizado la carga de imágenes hacia los servidores utilizados.

Figura N° 41: Envío de datos.



Elaborado por el equipo de trabajo.

Figura N° 42: Envío de datos.

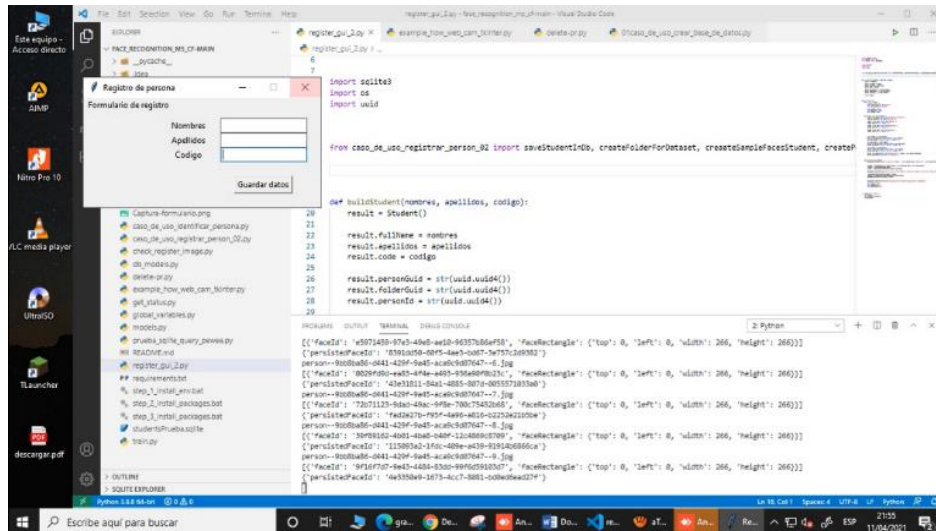


Elaborado por el equipo de trabajo.

4.1.14. Finalizado toma de datos

Como se muestra en la imagen la ventana emergente de registro queda en blanco para recibir los datos del siguiente personal y así sucesivamente para con cada uno de las personas de prueba en esta etapa.

Figura N° 43: Finalización de toma de datos

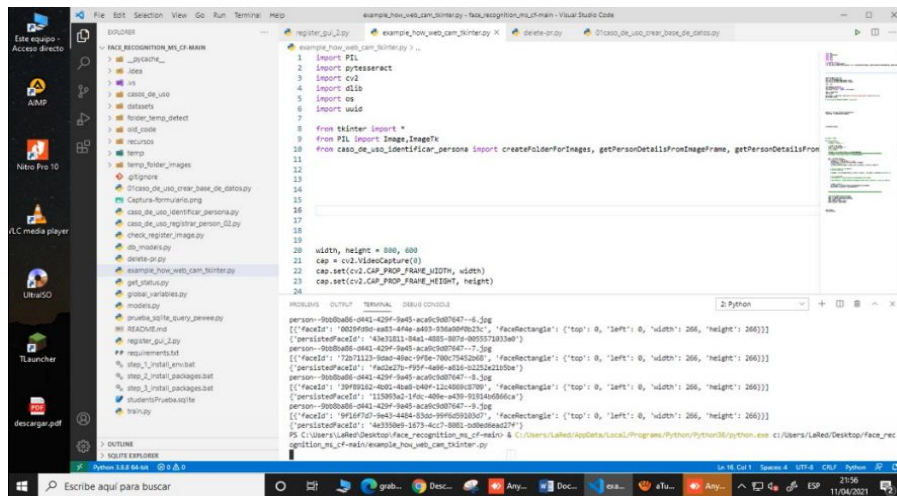


Elaborado por el equipo de trabajo.

4.1.15. Finalización del registro

Seguidamente posterior a la finalización de la primera persona ya se puede lograr a realizar el reconocimiento de la persona registrada así confirmar el registro satisfactorio y el funcionamiento correcto del sistema y el algoritmo.

Figura N° 44: Ventana de finalización del registro.

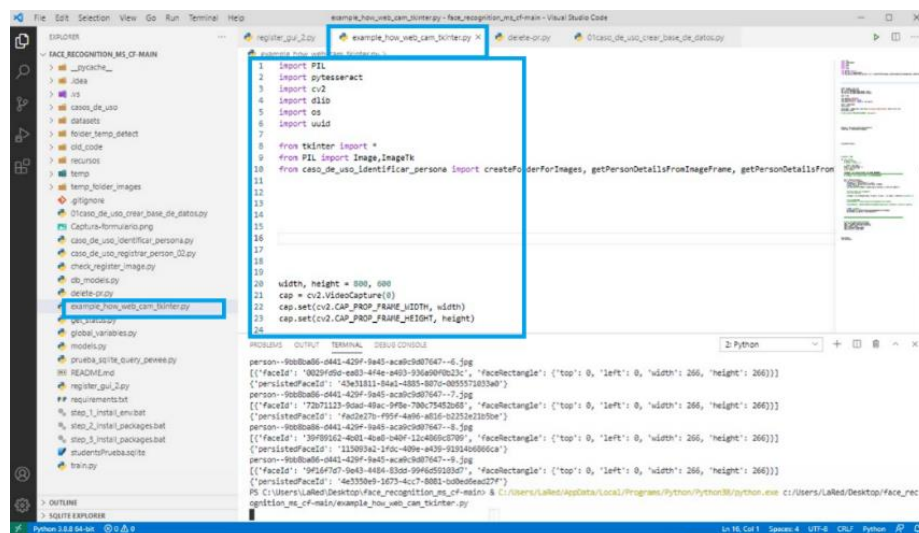


Elaborado por el equipo de trabajo.

4.2. Ejecución del programa

Una vez finalizado la compilación del algoritmo en el sistema se procede a ejecutar la etapa de reconocimiento para con todo el grupo de personas registradas en el sistema con antelación.

Figura N° 45: Inicio de ejecución de código.

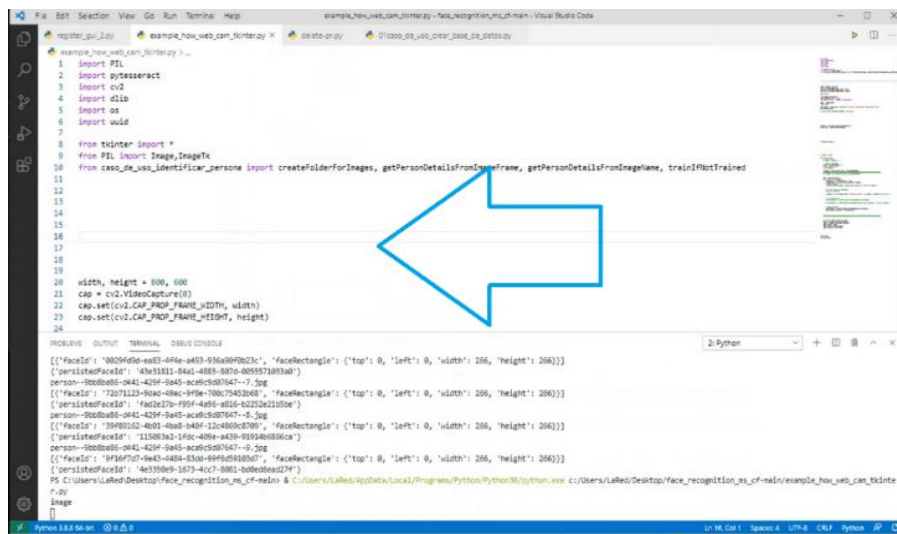


Elaborado por el equipo de trabajo.

4.2.1. Ejecutamos el programa de reconocimiento

Al iniciar el programa de reconocimiento se debe tomar en cuenta que el sistema aún se encuentra en etapa de prototipo por lo cual se tiene en cuenta que pueda lograr que existan algunos errores en dicho sistema como funcionamiento lento del algoritmo etapas que se queden en blanco, así como la probabilidad que el sistema no reconozca al rostro tomado por la cámara.

Figura N° 46: Ejecución de código de reconocimiento.

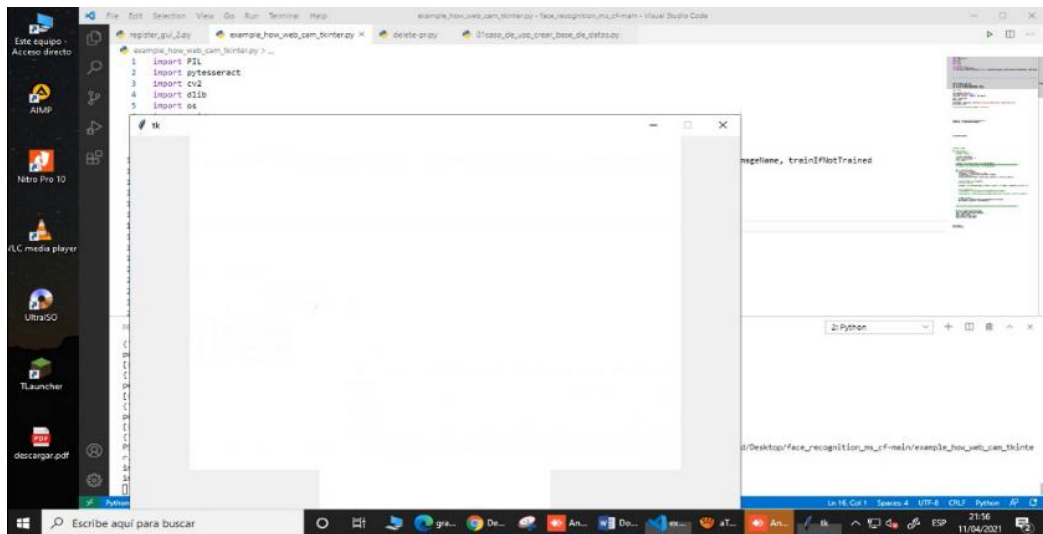


Elaborado por el equipo de trabajo.

4.2.2. Inicio de reconocimiento

La ventana emergente indica la inicialización del sistema por lo cual tomara la salida multimedia más próxima en el computador, el funcionamiento óptimo del sistema depende de muchos factores como son la calidad de video que se tiene para con la cámara que realiza las capturas de imagen así también depende de la robustez de la tarjeta de video y el procesador de la computadora a utilizar.

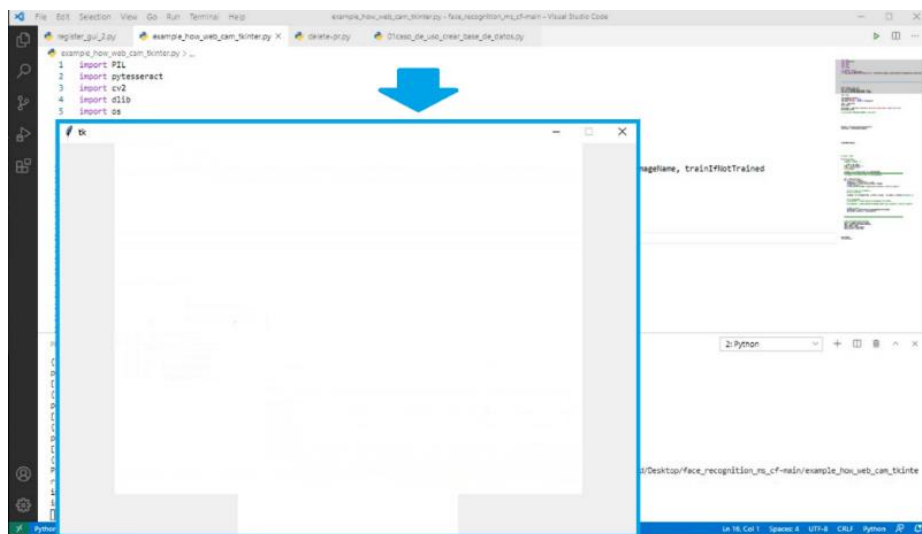
Figura N° 47: Inicio de reconocimiento.



Elaborado por autores

VENTANA: ventana emergente para el inicio de captura de rostros del sistema

Figura N° 48: Venta de inicio de captura.



Elaborado por autores

4.2.3. Muestras

Posterior a la captura de imágenes mediante la cámara de la computadora se realiza el envío de las capturas y al retorno se verifica la detección de rostros automático.

Figura N° 49: Toma de muestras.

```

1 # Importamos PIL para poder tomar fotos
2 import PIL
3 import cv2
4 import cv2.cv2
5 import cv2
6 import cv2
7
8 from tkinter import *
9 from PIL import Image, ImageTk
10 from tkinter import *
11
12 # Importamos cv2 para poder reconocer rostros
13 import cv2
14
15 # Importamos sys para poder salir del programa
16 import sys
17
18 # Importamos os para poder crear carpetas
19 import os
20 width, height = 800, 600
21 cap = cv2.VideoCapture(0)
22 cap.set(cv2.CAP_PROP_FRAME_WIDTH, width)
23 cap.set(cv2.CAP_PROP_FRAME_HEIGHT, height)
24
25 # Importamos sys para poder salir del programa
26 import sys
  
```

Elaborado por autores

4.2.4. Detección

Se verifica la detección de rostros en la plataforma para posteriormente mostrar los resultados del funcionamiento del sistema.

Figura N° 50: Etapa de detección

```

[{"person": "8090a8e-041-439f-94d5-a2a8109767d-7.jpg", "faceId": "72012122-80a6-484c-879c-78615432000", "faceRectangle": {"top": 0, "left": 0, "width": 266, "height": 266}},
{"person": "8090a8e-041-439f-94d5-a2a8109767d-7.jpg", "faceId": "f6a2a70-959f-4a96-af32-0229a23205e", "faceRectangle": {"top": 0, "left": 0, "width": 266, "height": 266}},
{"person": "8090a8e-041-439f-94d5-a2a8109767d-7.jpg", "faceId": "38f0932-40d1-4b4b-94df-22480103709", "faceRectangle": {"top": 0, "left": 0, "width": 266, "height": 266}},
{"person": "8090a8e-041-439f-94d5-a2a8109767d-7.jpg", "faceId": "110919a2-f4c6-d86e-40d8-9109068866e4", "faceRectangle": {"top": 0, "left": 0, "width": 266, "height": 266}},
{"person": "8090a8e-041-439f-94d5-a2a8109767d-7.jpg", "faceId": "912af107-94d5-4448-8302-99902053203", "faceRectangle": {"top": 0, "left": 0, "width": 266, "height": 266}},
{"person": "8090a8e-041-439f-94d5-a2a8109767d-7.jpg", "faceId": "4e32040e-3d73-4c77-88e2-30d6d8a221f", "faceRectangle": {"top": 0, "left": 0, "width": 266, "height": 266}}
  
```

Elaborado por autores

4.2.5. Detección de rostros

Una vez detectado el rostro del individuo el sistema deberá mostrar los datos detectados en la parte inferior de la ventana emergente para su lectura por el personal de la seguridad, para este caso deber de mostrar en la etapa de pruebas de funcionamiento

los datos ingresados aleatoriamente para su verificación de que fueron registrado y asignados a las imágenes correspondientes

Figura N° 51: Detección.

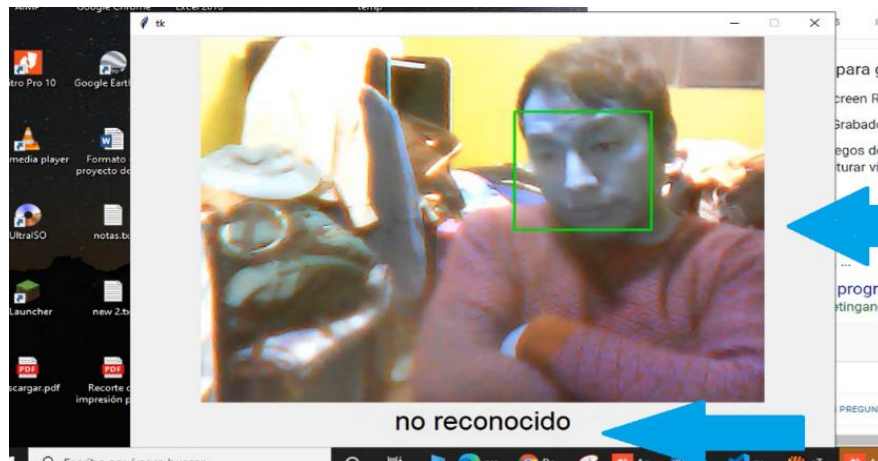


Elaborado por autores

4.2.6. Cambio de Ángulo

En esta etapa de pruebas se muestra el cambio de enfoque de la persona que se encuentra en etapa de toma de muestras, al no recibir la luz ambiente de la forma en que fueron tomadas las imágenes se produce un reconocimiento erróneo por algunos segundos hasta que la luz se asemeje a la toma de muestras inicial, para esta situación se tiene que afirmar que el funcionamiento correcto del sistema se verá afectado el ambiente en el que fueron tomadas las pruebas así también como los accesorios de uso común para el rostro. Como recomendación la toma de imágenes y/o muestras se deberán de hacer en entornos controlados para evitar errores ambientales como el que se presentó en las pruebas de funcionamiento.

Figura N° 52: Cambio de Ángulo

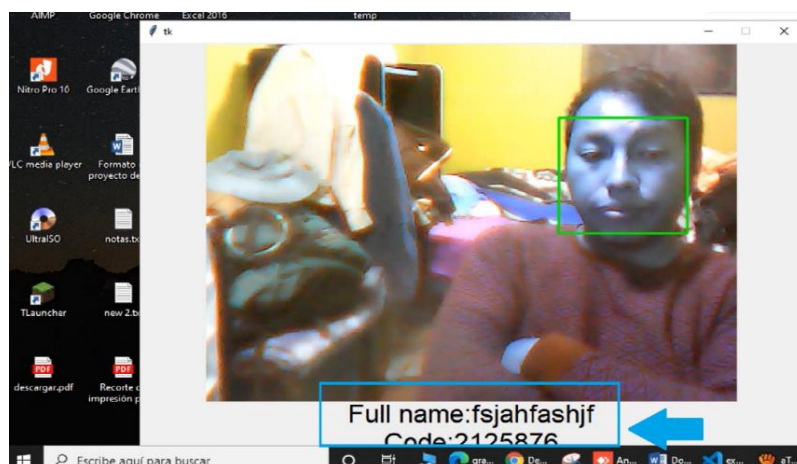


Elaborado por el equipo de trabajo.

4.2.7. Reconocimiento de rostros

Cambiado el enfoque los errores mencionados en la anterior imagen se corrigen ya que se encuentran en una aproximación ambiental aceptable para el sistema por lo cual se verifica los datos aleatorios ingresados en un inicio.

Figura N° 53: Reconocimiento.



Elaborado por el equipo de trabajo.

4.2.8. Reconocimiento

En la etapa de reconocimiento se debe de recalcar que se continúa apareciendo pequeños Flap de la imagen dicha situación debe a que en el momento de las pruebas se presentaron inconvenientes con el tráfico del internet contratado ya que al no ofrecer una

estabilidad de línea balanceada al 50% aproximadamente se verán estos errores más comúnmente de lo que se quiere para nuestro sistema.

Figura N° 54: Reconocimiento

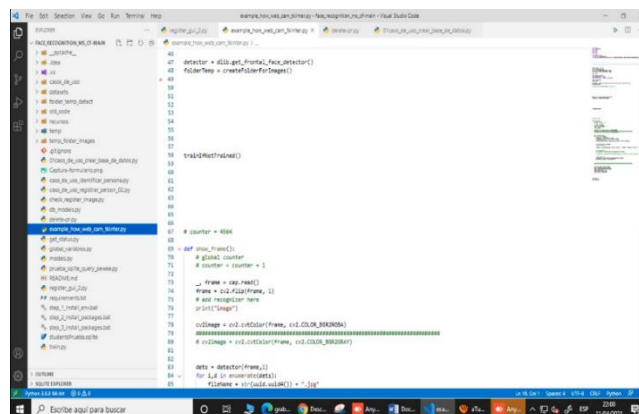


Elaborado por equipo de trabajo.

4.2.9. Finalización

Una vez finalizado el proceso de identificación por el personal de control de acceso el usuario deberá de presionar la tecla Esc y/o cerrar la ventana TK con el botón en la parte superior derecha, de esta manera se cierra el sistema y se procede a guardar los datos automáticamente sin la necesidad de manipular ningún otro objeto o herramienta de la plataforma.

Figura N° 55: Finalización.



Elaborado por el equipo de trabajo.

- V.D.: Sistema de Reconocimiento Facial
- V.I.: Control de acceso de los personas

4.3. Diseño de un prototipo de sistema de reconocimiento facial que facilite y ahorre tiempo al ingreso.

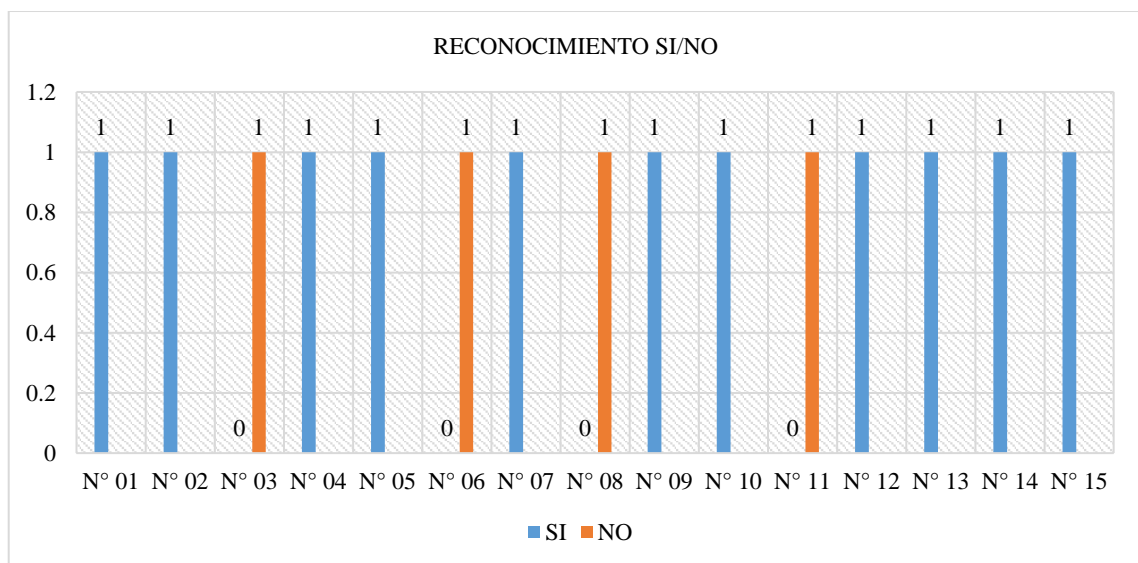
Tabla N° 4: Lista de reconocimiento facial.

Grupo	N° 01	N° 02	N° 03	N° 04	N° 05	N° 06	N° 07	N° 08	N° 09	N° 10	N° 11	N° 12	N° 13	N° 14	N° 15
Si	1	1	0	1	1	0	1	0	1	1	0	1	1	1	1
No			1			1		1			1				

Elaborado por el equipo de trabajo.

En la **Tabla N° 4** se muestra la lista original de 15 personas que accedieron a la prueba de reconocimiento facial, de las cuales 11 personas fueron reconocidas sin ningún tipo de problema, mientras que en 4 personas se presentó un defecto de reconocimiento, el cual representa que el 70% es la eficiencia del sistema de reconocimiento facial que facilita el control de asistencia y el tiempo de ingreso a la institución.

Figura N° 56: Diagrama de lista de reconocimiento facial



Elaborado por el equipo de trabajo.

En la **Figura N° 56**. Se muestra el diagrama de la cantidad de personas que se sometieron a la prueba de reconocimiento facial, en donde se puede observar que se trabajó con un total de 15 personas, de la cual 11 persona fueron reconocidas sin ningún tipo de dificultad, y 4 fueron reconocidos con ciertas dificultades esto por las prendas de vestir que llevaban con ello, así mismo el ángulo de la visión de la cámara cumple un papel fundamental en el reconocimiento facial, las 11 personas sometidas a la prueba de reconocida representan el 73% de reconocimiento sin dificultad y las 4 personas representan el 23% de reconocimiento con dificultad.

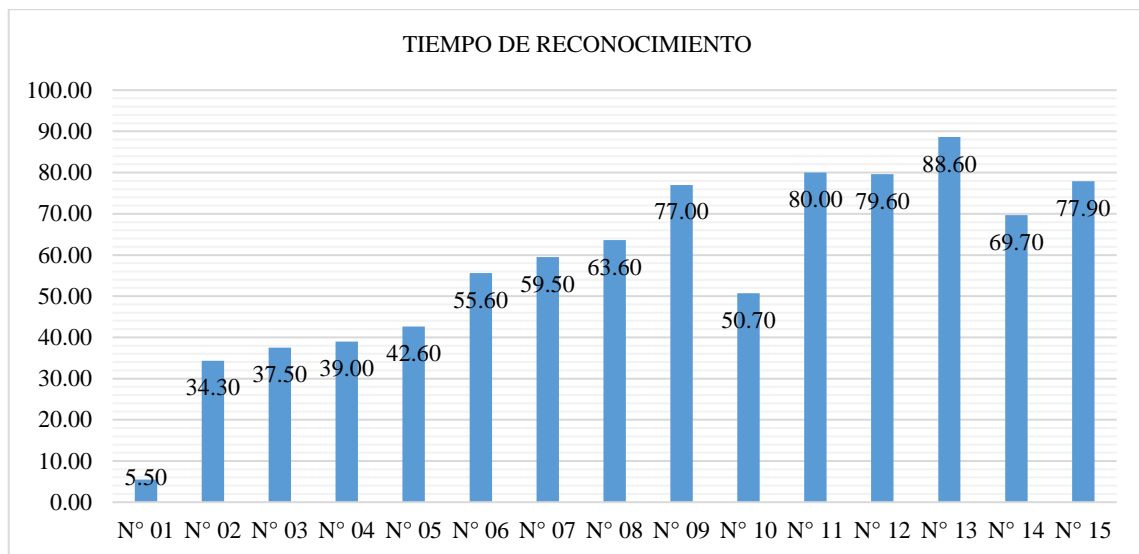
Tabla N° 5: Relación de asistentes y el tiempo de registro.

Grupo	N° 01	N° 02	N° 03	N° 04	N° 05	N° 06	N° 07	N° 08	N° 09	N° 10	N° 11	N° 12	N° 13	N° 14	N° 15
Si	1	1	0	1	1	0	1	0	1	1	0	1	1	1	1
No			1			1		1			1				
Tiempo en segundos	5.50	34.30	37.50	39.00	42.60	55.60	59.50	63.60	77.00	50.70	80.00	79.60	88.60	69.70	77.90

Elaborado por el equipo de trabajo.

En la **Tabla N° 5**. Se muestra la relación el registro de 15 participantes que se sometieron a la prueba de reconocimiento facial, así mismo se observa el tiempo de duración de registro en cada persona. El reconocimiento con el mayor tiempo de duración con el sistema de reconocimiento facial es de 88.60 segundos y el tiempo más corto es de 5.50 segundos, esto facilita la toma de asistencia directamente a la base de datos.

Figura N° 57: Diagrama de relación de asistentes y el tiempo de registro.



Elaborado por el equipo de trabajo.

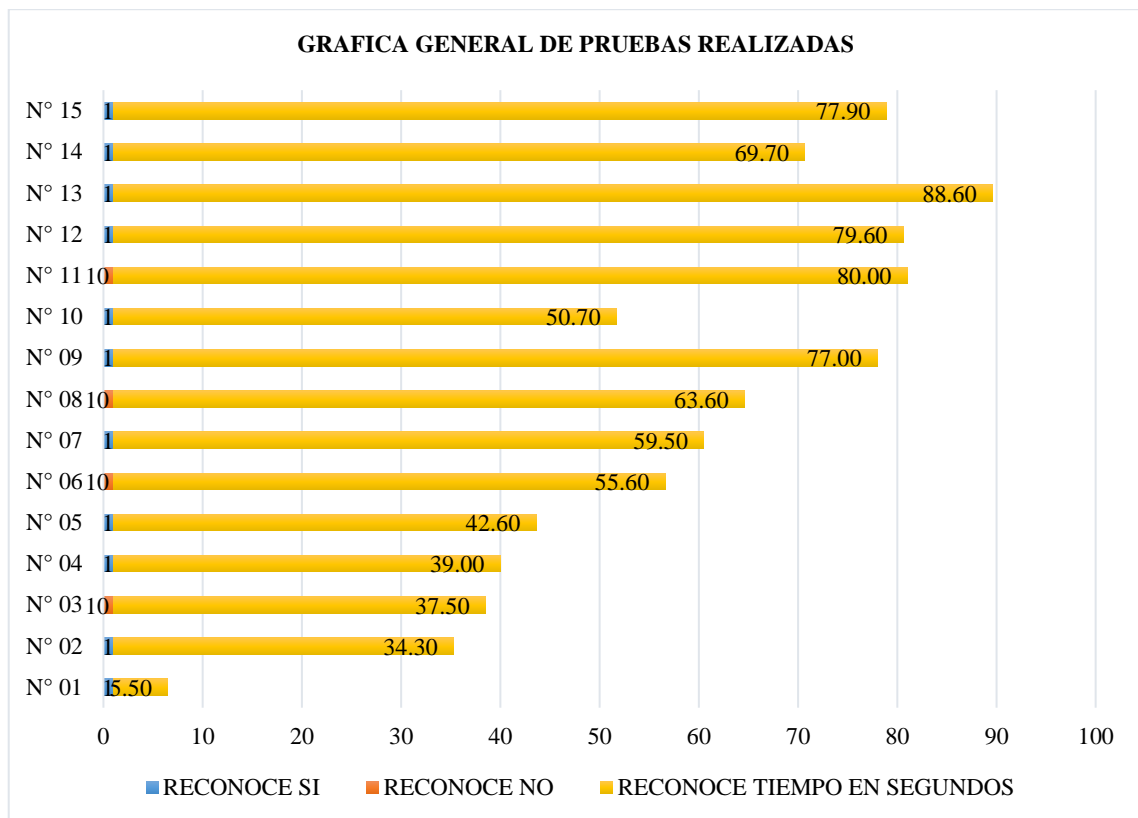
En la **Figura N° 57**. Se muestra el tiempo de reconocimiento de la personas al momento de registrarse para su asistencia, el reconocimiento con mayor tiempo se presenta en el registro N° 13 con 88.60 segundos, y el registro con menor tiempo se presenta en el registro N° 01 con 5.50 segundos, el tiempo de reconocimiento depende mucho de los ángulo de ubicación de la persona frente a la cámara y la iluminación del entorno.

Tabla N° 6: Resumen general de reconocimiento facial.

Grupo	N° 01	N° 02	N° 03	N° 04	N° 05	N° 06	N° 07	N° 08	N° 09	N° 10	N° 11	N° 12	N° 13	N° 14	N° 15
Si	1	1	0	1	1	0	1	0	1	1	0	1	1	1	1
No			1			1		1			1				
Tiempo en segundos	5.50	34.30	37.50	39.00	42.60	55.60	59.50	63.60	77.00	50.70	80.00	79.60	88.60	69.70	77.90
Promedio en segundos	57.41 segundos.														

Elaborado por el equipo de trabajo.

Figura N° 58: Diagrama general de reconocimiento facial.



Elaborado por el equipo de trabajo.

En la **Tabla N° 6** y **Figura N° 58**. Se muestran el resumen general del control de reconocimiento facial que facilite y ahorre tiempo al ingreso, en la cual se puede observar que en el registro N° 13 se tiene una mayor duración de tiempo con 88.60 segundos en condición de reconocido, así mismo en el reconocimiento N° 01 se presenta un tiempo menor al resto con 5.50 segundos.

4.3.1. Discusión

En el diseño de un prototipo de sistema de reconocimiento facial que facilite y ahorre tiempo al ingreso se presenta como resultado que de los 15 registros 11 fue recocida sin ningún tipo de problema y 4 no fueron reconocidos y el tiempo de reconocimiento facial tiene un promedio de 51.41 segundos. En los antecedentes internacionales (Castano & Alonso, 2019), presenta que su tiempo promedio en el



reconocimiento fácil es de 21 segundos, en antecedentes nacionales (Ipanaqué, 2020) indica en su investigación que el tiempo promedio de reconocimiento facial es 79 segundos, según los antecedentes de investigación se tiene diferentes tiempo de reconocimiento facial.

4.3.2. Contratación de hipótesis

El método que se utiliza para la contratación de hipótesis es el método de chi cuadrado el cual sirve para someter a prueba de hipótesis referidas a distribuciones de frecuencias.

Tabla N° 7: Datos para la contratación de hipótesis.

Muestras	Datos
N° 01	5.50
N° 02	34.30
N° 03	37.50
N° 04	39.00
N° 05	42.60
N° 06	55.60
N° 07	59.50
N° 08	63.60
N° 09	77.00
N° 10	50.70
N° 11	80.00
N° 12	79.60
N° 13	88.60
N° 14	69.70
N° 15	77.90
Promedio	57.41

Elaborado por el equipo de trabajo.

Muestra (n): en la presente investigación la muestra que se considera para la contratación de hipótesis son las 15 personas que se sometieron a la prueba

- Varianza (σ^2): la varianza se calcula con la siguiente formula:

$$S^2 = \frac{\sum(x_i - X)^2}{n - 1}$$

$$S^2 = \frac{\sum(5.50 - 57.41)^2 + (34.30 - 57.41)^2 \dots}{15 - 1}$$

- Desviación estándar (σ)

$$S = \sqrt{\frac{\sum(x_i - X)^2}{n - 1}}$$

$$S = \sqrt{\frac{\sum(5.50 - 57.41)^2 + (34.30 - 57.41)^2 \dots}{15 - 1}}$$

- Coeficiente de varianza (μ)

$$\mu = \frac{S}{\text{Promedio}}$$

- Z y Z crítico

Figura N° 59: Calculo para Z crítico.

zi	0,00	0,01	0,02	0,03	0,04	0,05	0,06	0,07	0,08	0,09
1,5	0,9331	0,9344	0,9357	0,9369	0,9382	0,9394	0,9406	0,9417	0,9429	0,9440
1,6	0,9452	0,9463	0,9473	0,9484	0,9494	0,9505	0,9515	0,9525	0,9535	0,9544
1,7	0,9554	0,9563	0,9572	0,9581	0,9590	0,9599	0,9607	0,9616	0,9624	0,9632
1,8	0,9640	0,9648	0,9656	0,9663	0,9671	0,9678	0,9685	0,9692	0,9699	0,9706
1,9	0,9712	0,9719	0,9725	0,9731	0,9738	0,9744	0,9750	0,9755	0,9761	0,9767
2,0	0,9772	0,9777	0,9783	0,9788	0,9793	0,9798	0,9803	0,9807	0,9812	0,9816
2,1	0,9821	0,9825	0,9829	0,9834	0,9838	0,9842	0,9846	0,9849	0,9853	0,9857
2,2	0,9860	0,9864	0,9867	0,9871	0,9874	0,9877	0,9880	0,9883	0,9886	0,9889
2,3	0,9892	0,9895	0,9898	0,9900	0,9903	0,9906	0,9908	0,9911	0,9913	0,9915
2,4	0,9918	0,9920	0,9922	0,9924	0,9926	0,9928	0,9930	0,9932	0,9934	0,9936
2,5	0,9937	0,9939	0,9941	0,9942	0,9944	0,9946	0,9947	0,9949	0,9950	0,9952
2,6	0,9953	0,9954	0,9956	0,9957	0,9958	0,9959	0,9960	0,9962	0,9963	0,9964
2,7	0,9965	0,9966	0,9967	0,9968	0,9969	0,9970	0,9971	0,9971	0,9972	0,9973
2,8	0,9974	0,9975	0,9975	0,9976	0,9977	0,9978	0,9978	0,9979	0,9980	0,9980
2,9	0,9981	0,9981	0,9982	0,9983	0,9983	0,9984	0,9984	0,9985	0,9985	0,9986
3,0	0,9986	0,9986	0,9987	0,9987	0,9988	0,9988	0,9988	0,9989	0,9989	0,9989
3,1	0,9990	0,9990	0,9990	0,9991	0,9991	0,9991	0,9992	0,9992	0,9992	0,9992
3,2	0,9993	0,9993	0,9993	0,9993	0,9994	0,9994	0,9994	0,9994	0,9994	0,9994
3,3	0,9995	0,9995	0,9995	0,9995	0,9995	0,9995	0,9996	0,9996	0,9996	0,9996

Fuente: (Pacori, 2018).

- Z prueba

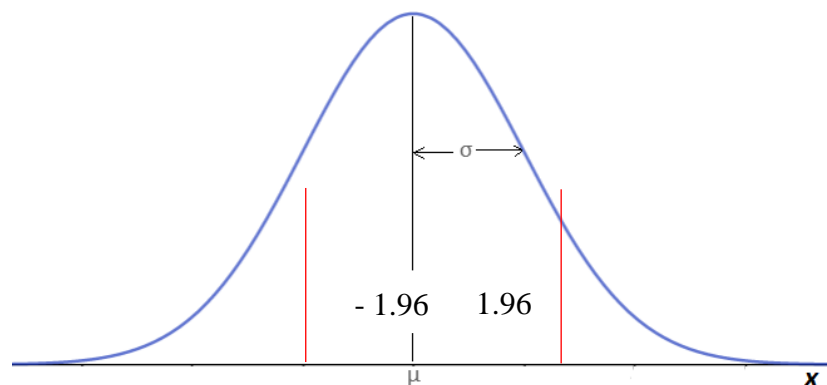
$$Z = \frac{X - \mu}{\frac{S}{\sqrt{n}}}$$

Tabla N° 8: Cuadro de prueba de hipótesis.

Muestra (n)	15.00
Varianza (σ^2)	514.61
Desviación. Estándar (S)	22.68
Coefficiente de varianza (μ)	0.40
Promedio	57.41
Nivel de confianza	0.95
Significancia (α)	0.05
Z	0.98
Z critico	1.96
Z prueba	0.65

Elaborado por el equipo de trabajo.

Figura N° 60: Prueba de hipótesis campana de Gauss



Elaborado por el equipo de trabajo.

En la **Tabla N° 8** y **Figura N° 60**. Se muestra la prueba de hipótesis de la presente investigación en donde se puede afirmar la hipótesis que se planteó, indicando que la Z crítico tiene un valor de 1.96, mientras que la Z de prueba tiene un valor de 0.65 menor a Z crítico, por lo que la hipótesis planteado es afirmativo “el diseño de un prototipo de sistema de reconocimiento facial para el control biométrico utilizando las librerías de OpenCV y la app Microsoft AZURE facilitara el ahorro de tiempo en 90% al ingreso del colegio Aplicación de la Universidad Nacional del Altiplano”

4.4. Diseño de un prototipo de sistema de reconocimiento facial para el control biométrico que permita la sistematización de la asistencia.

Tabla N° 9: Lista de control biométrico para la asistencia.

Grupo	Si	No	Asistencia
N° 01	1		Asistió
N° 02	1		Asistió
N° 03	0	1	No
N° 04	1		Asistió
N° 05	1		Asistió
N° 06	0	1	No
N° 07	1		Asistió
N° 08	0	1	No
N° 09	1		Asistió
N° 10	1		Asistió
N° 11	0	1	No
N° 12	1		Asistió
N° 13	1		Asistió
N° 14	1		Asistió
N° 15	1		Asistió

Elaborado por el equipo de trabajo.

En la **Tabla N° 9**. Se puede visualizar la lista final del registro de asistencia y el tiempo de toma de datos para el registro final de la asistencia de los personas, en donde la cantidad de participante para la presente investigación son 15 personas, de la cual 11 personas fueron reconocidas sin problemas el cual representa 73 %, y se marcó como asistido y las otras 4 personas que no fueron reconocidos por el sistema de reconocimiento facial se marcó como no asistido lo cual representa el 23%.

4.4.1. Discusión

En el diseño de un prototipo de sistema de reconocimiento facial para el control biométrico permitió marcar las asistencia de forma automática, en la cual 14 personas fueron reconocidos y marcados “ASISTIÓ” y las otras 4 personas no fueron reconocidos y fueron marcada como “NO” En el antecedente internacional de (Ortiz, 2014) indica que



el sistema de reconocimiento facial marca de manera automática las asistencia, el área de trabajo, especialidades, reuniones y la importancia del trabajador de empresa en donde se implementó este sistema. En los antecedentes nacionales (Yañez, 2019) indica que se concluye que, el porcentaje de asistencias en el laboratorio de cómputo de la Facultad de Ingeniería Industrial y Sistemas, aumento considerablemente gracias a la ayuda de sistema de reconocimiento facial, debido a que el porcentaje de asistencias antes de implementar el sistema fue de un 67 % de asistencias de personas y después de que se implementó el sistema de reconocimiento facial alcanza el 80 % de asistencias del estudiante. Por lo tanto, se puede afirmar que gracias a la implementación del sistema de reconocimiento facial aumenta el porcentaje de asistencias, por lo que se ve que aumento el 23 % de asistencias de los alumnos, esto se debe al control que se establece al colocar un sistema de reconocimiento facial dentro del laboratorio de cómputo.

4.4.2. Contratación de hipótesis

El método que se utiliza para la contratación de hipótesis es el método de chi cuadrado el cual sirve para someter a prueba de hipótesis referidas a distribuciones de frecuencias.

Tabla N° 10: Datos para la contratación de hipótesis.

Muestras	Datos
N° 01	1
N° 02	1
N° 03	0
N° 04	1
N° 05	1
N° 06	0
N° 07	1
N° 08	0
N° 09	1
N° 10	1
N° 11	0
N° 12	1
N° 13	1



Muestras	Datos
N° 14	1
N° 15	1

Elaborado por el equipo de trabajo.

Muestra (n): en la presente investigación la muestra que se considera para la contratación de hipótesis son las 15 personas que se sometieron a la prueba

- Varianza (σ^2): la varianza se calcula con la siguiente formula:

$$S^2 = \frac{\sum(x_i - X)^2}{n - 1}$$

$$S^2 = \frac{\sum(1 - 0.73)^2 + (1 - 0.73)^2 + (0 - 0.73)^2 \dots}{15 - 1}$$

- Desviación estándar (σ)

$$S = \sqrt{\frac{\sum(x_i - X)^2}{n - 1}}$$

$$S = \sqrt{\frac{\sum(1 - 0.73)^2 + (1 - 0.73)^2 + (0 - 0.73)^2 \dots}{15 - 1}}$$

- Coeficiente de varianza (μ)

$$\mu = \frac{S}{\text{Promedio}}$$

- Z y Z critico

Figura N° 61: Calculo para Z crítico.

zi	0,00	0,01	0,02	0,03	0,04	0,05	0,06	0,07	0,08	0,09
1,5	0,9331	0,9344	0,9357	0,9369	0,9382	0,9394	0,9406	0,9417	0,9429	0,9440
1,6	0,9452	0,9463	0,9473	0,9484	0,9494	0,9505	0,9515	0,9525	0,9535	0,9544
1,7	0,9554	0,9563	0,9572	0,9581	0,9590	0,9599	0,9607	0,9616	0,9624	0,9632
1,8	0,9640	0,9648	0,9656	0,9663	0,9671	0,9678	0,9685	0,9692	0,9699	0,9706
1,9	0,9712	0,9719	0,9725	0,9731	0,9738	0,9744	0,9750	0,9755	0,9761	0,9767
2,0	0,9772	0,9777	0,9783	0,9788	0,9793	0,9798	0,9803	0,9807	0,9812	0,9816
2,1	0,9821	0,9825	0,9829	0,9834	0,9838	0,9842	0,9846	0,9849	0,9853	0,9857
2,2	0,9860	0,9864	0,9867	0,9871	0,9874	0,9877	0,9880	0,9883	0,9886	0,9889
2,3	0,9892	0,9895	0,9898	0,9900	0,9903	0,9906	0,9908	0,9911	0,9913	0,9915
2,4	0,9918	0,9920	0,9922	0,9924	0,9926	0,9928	0,9930	0,9932	0,9934	0,9936
2,5	0,9937	0,9939	0,9941	0,9942	0,9944	0,9946	0,9947	0,9949	0,9950	0,9952
2,6	0,9953	0,9954	0,9956	0,9957	0,9958	0,9959	0,9960	0,9962	0,9963	0,9964
2,7	0,9965	0,9966	0,9967	0,9968	0,9969	0,9970	0,9971	0,9971	0,9972	0,9973
2,8	0,9974	0,9975	0,9975	0,9976	0,9977	0,9978	0,9978	0,9979	0,9980	0,9980
2,9	0,9981	0,9981	0,9982	0,9983	0,9983	0,9984	0,9984	0,9985	0,9985	0,9986
3,0	0,9986	0,9986	0,9987	0,9987	0,9988	0,9988	0,9988	0,9989	0,9989	0,9989
3,1	0,9990	0,9990	0,9990	0,9991	0,9991	0,9991	0,9992	0,9992	0,9992	0,9992
3,2	0,9993	0,9993	0,9993	0,9993	0,9994	0,9994	0,9994	0,9994	0,9994	0,9994
3,3	0,9995	0,9995	0,9995	0,9995	0,9995	0,9995	0,9996	0,9996	0,9996	0,9996

Fuente: (Pacori, 2018).

- Z prueba

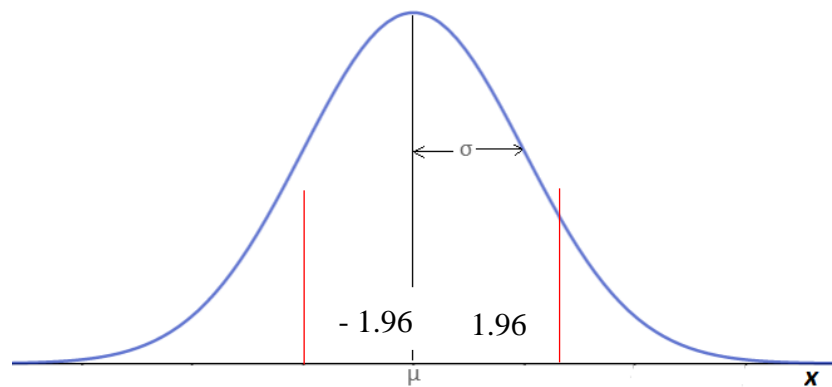
$$Z = \frac{X - \mu}{\frac{S}{\sqrt{n}}}$$

Tabla N° 11: Cuadro de prueba de hipótesis.

Muestra (n)	15,00
Varianza (σ^2)	0,21
Desviación. Estándar (S)	0,46
Coefficiente de varianza (μ)	0,63
Promedio	0,73
Nivel de confianza	0,95
Significancia (α)	0,05
Z	0,98
Z crítico	1,96
Z prueba	0,06

Elaborado por el equipo de trabajo.

Figura N° 62: Prueba de hipótesis campana de Gauss



Elaborado por el equipo de trabajo.

En la **Tabla N° 11** y **Figura N° 62**. Se muestra la prueba de hipótesis de la presente investigación en donde se puede afirmar la hipótesis que se planteó, indicando que la Z crítico tiene un valor de 1.96, mientras que la Z de prueba tiene un valor de 0.06 menor a Z crítico, por lo que la hipótesis planteado es afirmativo “El diseño de un prototipo de sistema de reconocimiento facial para el control biométrico que permita la sistematización un 95% de la asistencia en el colegio Aplicación de la Universidad Nacional del Altiplano”



V. CONCLUSIONES

El desarrollo de software utilizando las librerías de Open CV y la app de Microsoft Azure permitió elaborar un sistema con reconocimiento facial para controlar a los participantes, la cual permitió mejorar la seguridad de ingresar al interior de la institución colegio de Aplicación de la Universidad Nacional del Altiplano en la cual se puede visualizar las imágenes de prueba así mismo se pudo ver que el software de reconocimiento facial reconoce a las personas con seguridad de acuerdo al Ángulo de toma de muestra, el sistema desarrollado para la detección de rostros muestra una eficiencia de tiempo descrita en la sección de pruebas por lo tanto la implementación de dicho software podría entrar a ensayos e implementación en el colegio Aplicación de la Universidad Nacional del Altiplano.

El diseño del prototipo de reconocimiento facial, si facilita el ingreso de personas a la institución tiene un promedio de tiempo de registro e ingreso de 51.41 segundo por cada persona que ingresa a la institución, así mismo esto facilita que la asistencia de las personas se generó de manera automática en la base de datos del software, en donde se pudo observar que de 15 participantes sometidos al reconocimiento facial, 11 participantes fueron reconocidos sin ningún tipo de problemas mientras que los 4 restantes fueron rechazados en primera instancia hasta que se posicionen en el Ángulo adecuado, El uso de software de reconocimiento en las instalaciones del colegio de aplicación propone una solución sistemática relativamente eficiente, reflejando un ahorro de tiempo de clases presenciales.

El sistema de reconocimiento facial cumple con la función de registrar a todos los participantes que pasan por el reconocimiento facial, en la presente investigación se hizo la prueba con 15 personas en la cual el software reconoció sin problemas a 11 participantes y registro con “asistió” mientras que los 4 participantes fueron rechazados y fueron registrados con “NO”.



VII. RECOMENDACIONES

Se recomienda que para el buen uso del sistema de para una mejora del presente proyecto es necesario el desarrollo de un algoritmo basado en inteligencia artificial para de esta manera el algoritmo pueda realizar más eficientemente el procesamiento de formas, patrones y colores de esta manera se lograría una mayor eficiencia en la detección de rostros, ya que no se ha realizado en condiciones más rígidas, se encontrarían mayores inconvenientes a considerar.

Se recomienda que para el uso de este sistema se debe de implementar una cámara con mayor capacidad, el cual cumplirá un gran papel en el reconocimiento de rostros el cual mejoraría significativamente el sistema debido a que el reconocimiento de rostros depende en gran medida a la calidad de grabación de la misma.

Se recomienda que para el registro de participantes reconocidos por el sistema de reconocimiento facial sean enviados a una base de datos donde puedan acceder las máximas autoridades de la institución, y que si se registra alguna falta de un participante sea notificado de forma inmediata a los responsables de cada participante (personas)



VII. REFERENCIAS BIBLIOGRÁFICAS

- Apaza, R., & Charaja, G. (2013). *Sistema para detección y reconocimiento facial utilizando técnicas híbridas en imágenes y secuencias de video Puno 2013*. Repositorio Institucional UNAP, Puno. Obtenido de <https://renati.sunedu.gob.pe/handle/sunedu/2905700>
- Bunge, M. (1989). *La Ciencia, su método y su filosofía*. Mexico: Ediciones Siglo XX. argentina.
- Caballero, F., Reyes, M., Sánchez, A., & Ríos, C. (2017). *Reconocimiento facial por el método de eigenfaces*. Obtenido de <http://webcache.googleusercontent.com/search?q=cache:q-ImlzQIY-YJ:itcelaya.edu.mx/ojs/index.php/pistas/article/download/1068/878+&cd=1&hl=es&ct=clnk&gl=pe>
- Cáceres, E. (2018). *Aplicación móvil de reconocimiento facial en personas con antecedentes de abuso sexual en la provincia de Andahuaylas, Apurímac - 2018*. Obtenido de <https://renati.sunedu.gob.pe/handle/sunedu/2894847>
- Capacho, J. R., & Nieto Bernal, W. (2017). *Diseño de Base de Datos*. Barranquilla: Universidad del Norte.
- Castano, D. L., & Alonso, J. D. (2019). *Sistema de reconocimiento facial para control de acceso a viviendas*. Obtenido de <https://repository.ucatolica.edu.co/bitstream/10983/24032/1/Final%20Trabajo%20de%20grado.pdf>
- Conde, R. (2017). *The new way to deliver technology*.
- Espinoza, E. (2018). *La hipótesis en la investigación*.



- Fidias G, A. (2016). *El Proyecto de Investigacion*.
- Ipanaqué, J. (2020). *Análisis comparativo de técnicas de reconocimiento facial en ambientes no controlados para optimizar el proceso de registro de personal de la ugel ferreñafe*. Obtenido de <https://repositorio.uss.edu.pe/bitstream/handle/20.500.12802/6964/JOS%C3%89%20JES%C3%9AS%20IPANAQU%C3%89%20CASANOVA.pdf?sequence=1&isAllowed=y>
- Jaliri, M., Ramos , M., Peralta, E., Escamillai, m., Luna , V., Rosales, E., . . . Ramos, A. (2019). *Tecnología Informatica*. Obtenido de https://www.ecorfan.org/spain/researchjournals/Tecnologia_Informatica/vol3num9/Revista_de_Tecnolog%C3%ADa_Inform%C3%A1tica_V3_N9.pdf
- Julían, F. G., Vidal Reyes, M., López Sánchez, A., & Jerónimo Ríos, C. (2017). *RECONOCIMIENTO FACIAL POR EL MÉTODO DE EIGENFACES*. México: Pistas Educativas.
- Larcher, L., BIASONI, E., Cattaneo, C., Ruggeri, A., & Herrera, C. (2011). *Algoritmo para Detección de Bordes y Ulterior Determinación de Objetos en Imágenes Digitales*. Obtenido de <https://cimec.org.ar/ojs/index.php/mc/article/view/3955>
- Mao , W., & Valencia, L. (2021). El uso de tecnología de reconocimiento facial en el proceso de pago para promover la economía de Mexico durante la pandemia. *exploratoris: revista de la realidad global*, 8.
- Mariño, E. L. (2018). *Aplicación móvil de reconocimiento facial en personas con trabajo previos de abuso sexual en la provincia de Andahuaylas*. Andahuaylas .
- Maya, E. (2014). *Métodos y técnicas de investigación*.
- Mejía, E. (2005). *Técnicas e Instrumentos de Investigación*.



- Monje, C. (2016). *Metodología de la investigación cuantitativa y cualitativa. Guía didáctica.*
- Mosquera , V., & Romero, E. (2016). *DISEÑOR DE UN SOFTWARE PILOTODiseño de un software piloto de reconocimiento facial para el control de asistencia en la escuela de telecomunicaciones de la Universidad de Carabobo.* Barbulo. Obtenido de <http://mriuc.bc.uc.edu.ve/handle/123456789/5232>
- Mosquera, V., & Romero , E. (2016). *Diseño de un software piloto de reconocimiento facial para el control de asistencia en la escuela de telecomunicaciones de la universidad de Carabobo.*
- OpenCV. (04 de 07 de 2021). *wikipedia.* Obtenido de wikipedia: <https://es.wikipedia.org/wiki/OpenCV>
- Ortiz, T. (2014). *Sistema de reconocimiento facial automatizado para el control de asistencia de recursos humanos.* Obtenido de <https://dspace.unl.edu.ec/jspui/bitstream/123456789/14237/1/Ortiz%20Pesantes%2c%20Tatiana%20Maricela.pdf>
- Pacori, E. (2018). *Estadística básica e investigación con aplicación SPSS.*
- Padúa, J. (1979). *Técnicas de Investigación Aplicadas a las Ciencias Sociales.* Mexico: fondo de Cultura Económica.
- Padúa, J. (1979). *Técnicas de Invetigación aplicadas a las Ciencias Sociales.* Mexico: Fondo de Cultura Económica .
- Python. (13 de 07 de 2021). *wikipedia.* Obtenido de wikipedia: <https://es.wikipedia.org/wiki/Python>



- Saavedra, D. L., & Alonso Sierra, J. D. (2019). *Sistema de Reconocimiento Facial para Control de Acceso a Viviendas*. Bogota.
- Sabino, C. (1996). *El proceso de investigacion*.
- Sampieri, R. H., Fernades Collazo, C., & Pilar Baptista, L. (2014). *Metodología de la Investigación*. Mexico: MCGRAW-HILL / INTERAMERICANA EDITORES, S.A. DE C.V.
- Sandoval, C. Y., & Estela Villena, C. C. (2015). *Detección de alumnos mediante mecanismos proactivos para el control de asistencia para la facultad de ingeniería y arquitectura de la universidad San Martin*. Lima.
- sousa, k., & Mora, C. (2016). *Sistema de seguridad basado en reconocimiento facial utilizando una Raspberry Pi*[Trabajo especial de Grado, universidad central de venezuela]. UNIVERSIDAD CENTRAL DE VENEZUELA, caracas, venezuela.
- Studio, M. V. (06 de 07 de 2021). *wikipedia*. Obtenido de wikipedia: https://es.wikipedia.org/wiki/Microsoft_Visual_Studio
- Tesillo, C. (2016). *Análisis comparativo de los algoritmos FISHERFACEs y LBPH para el reconocimiento facial en diferentes condiciones de iluminación y pose, Tacna - 2015*. Obtenido de http://repositorio.unjbg.edu.pe/bitstream/handle/UNJBG/2479/954_2016_tesillo_gomez_cm_fain_ingenieria_informatica_y_sistemas.pdf?sequence=1&isAllowed=y
- Turk, M., & Pentland, A. (2016). *Face Recognition Using Eigenfaces*[ensayo; universidad de califonia ,san diego]. *IEEEExplore*. Obtenido de <https://www.cin.ufpe.br/~rps/Artigos/Face%20Recognition%20Using%20Eigenfaces.pdf>



Yañez, M. L. (2019). *Sistema de reconocimiento facial para el control de acceso de estudiantes a los laboratorios de la FIIS-UNAC, 2019*. Obtenido de <https://repositorio.ucv.edu.pe/handle/20.500.12692/44310>



ANEXOS

ANEXO 1

MATRIZ DE CONSISTENCIA

TITULO: DISEÑO DE UN PROTOTIPO DE SISTEMA DE RECONOCIMIENTO FACIAL PARA EL CONTROL BIOMÉTRICO EN EL COLEGIO APLICACIÓN DE LA UNIVERSIDAD NACIONAL DEL ALTIPLANO

PROBLEMA	OBJETIVOS	HIPÓTESIS
<p>PROBLEMA GENERAL: ¿Es necesario el reconocimiento facial para el control biométrico en el ingreso del colegio de Aplicación de la Universidad Nacional del Altiplano Puno 2019?</p>	<p>OBJETIVO GENERAL. •Diseñar un prototipo de software para el sistema de reconocimiento facial y control biométrico al ingreso del Colegio de Aplicación de la Universidad Nacional del Altiplano.</p>	<p>HIPÓTESIS GENERAL. Es posible diseñar un prototipo de sistema de reconocimiento facial para el control biométrico en el Colegio de Aplicación de la Universidad Nacional del Altiplano.</p>
<p>PREGUNTA ESPECIFICA 1 ¿Se mejorará la seguridad con el sistema de reconocimiento facial para el control biométrico en el ingreso del colegio de Aplicación de la Universidad Nacional del Altiplano.</p>	<p>OBJETIVO ESPECIFICO 1 Diseñar un prototipo de software de reconocimiento facial para el control biométrico que permita mejorar la seguridad al ingreso del Colegio de Aplicación de la Universidad Nacional del Altiplano.</p>	<p>HIPÓTESIS ESPECIFICA 1 El diseño del prototipo de sistema de reconocimiento facial para el control biométrico utilizando las librerías de OpenCV y la app Microsoft Azure garantizara la seguridad al ingreso del colegio Aplicación de la Universidad Nacional del Altiplano.</p>
<p>PREGUNTA ESPECIFICA 2 ¿El sistema de reconocimiento facial para el control biométrico mejorará el tiempo de respuesta evitando la aglomeración en el ingreso del colegio de Aplicación de la Universidad Nacional del Altiplano Puno 2019?</p>	<p>OBJETIVO ESPECIFICO 2 Diseñar un prototipo de sistema de reconocimiento facial que facilite y ahorre tiempo al ingreso del Colegio de Aplicación de la Universidad Nacional del Altiplano.</p>	<p>HIPÓTESIS ESPECIFICA 2 El diseño de un prototipo de sistema de reconocimiento facial para el control biométrico utilizando las librerías de OpenCV y la app Microsoft AZURE facilitara el ahorro de tiempo en 90% al ingreso del colegio Aplicación de la Universidad Nacional del Altiplano.</p>
<p>PREGUNTA ESPECIFICA 3 ¿El sistema de reconocimiento facial para el control biométrico permitirá la sistematización de la asistencia del colegio de Aplicación de la Universidad Nacional del Altiplano Puno 2019?</p>	<p>OBJETIVO ESPECIFICO 3 Diseñar un prototipo de sistema de reconocimiento facial para el control biométrico que permita la sistematización de la asistencia en el Colegio de Aplicación de la Universidad Nacional del Altiplano.</p>	<p>HIPÓTESIS ESPECIFICA 3 El diseño de un prototipo de sistema de reconocimiento facial para el control biométrico que permita la sistematización un 95% de la asistencia en el colegio Aplicación de la Universidad Nacional del Altiplano.</p>



Continuación.

VARIABLE	DIMENSIÓN	INDICADOR	METODOLOGÍA
VARIABLE INDEPENDIENTE Prototipo de sistema de reconocimiento facial	Viabilidad	<ul style="list-style-type: none"> - Diseño del prototipo de sistema de reconocimiento facial SÍ permite el control biométrico. - Diseño del prototipo de sistema de reconocimiento facial NO permite el control biométrico. 	TIPO DE INVESTIGACIÓN. Será de tipo descriptivo
VARIABLE DEPENDIENTE Control biométrico para el ingreso	Seguridad	<ul style="list-style-type: none"> - Imagen de rostro. - Distancia entre ojos. - Distancia de labios a mentón. - Márgenes de rostros. - Puntos característicos. 	DISEÑO DE INVESTIGACIÓN. Cuasi experimental.
	Tiempo	<ul style="list-style-type: none"> - Segundos - Minutos 	ENFOQUE. Cuantitativo.
	Sistematización.	<ul style="list-style-type: none"> - Por grado. - Por sección. - Por edad. - Por porcentajes - Por sexo. 	POBLACIÓN. personas del Colegio Aplicación UNA-PUNO. 2019 MUESTRA: Salón de 49 alumnos del tercer grado de secundaria turno mañana y tarde

ANEXO 2

PROCESO DE RECONOCIMIENTO

Código para la creación de base de datos

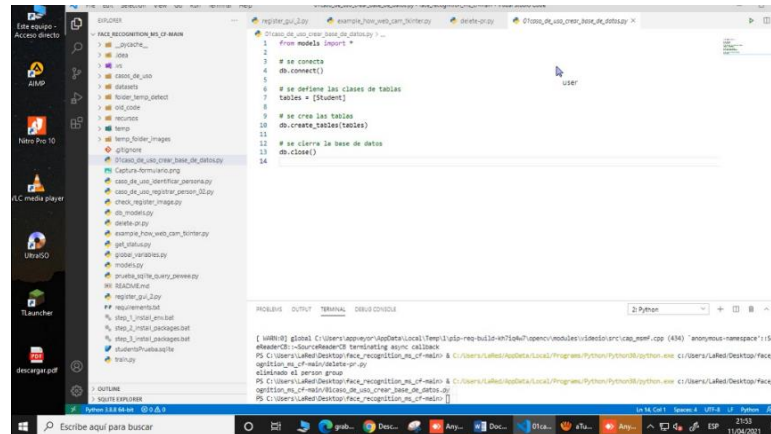


Figura: 1

Carpeta y código para la creación de base de datos

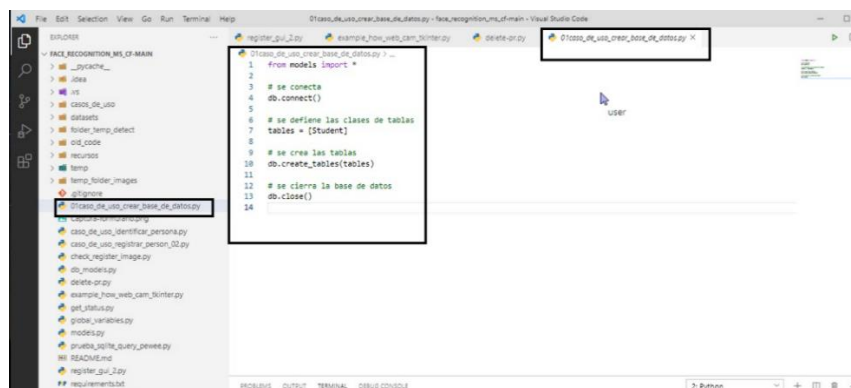


Figura: 2

ELIMINAR PERSON GROUP

Esta etapa para poder realizar el inicio desde cero del sistema se debe de eliminar los grupos existentes en el software para que no existan problemas en la búsqueda de imágenes ya subidas al software o imágenes duplicadas, imágenes desenfocadas e imágenes pixeladas producidas por la baja calidad de las cámaras utilizadas o algún problema físico en la implementación del sistema

```
1 import cv2
2 import numpy as np
3 import face_recognition as face_rec
4
5 # Definición de la clase Persona
6 class Persona:
7     def __init__(self, nombre, apellido, codigo):
8         self.nombre = nombre
9         self.apellido = apellido
10        self.codigo = codigo
11
12    def __str__(self):
13        return f'Persona(nombre={self.nombre}, apellido={self.apellido}, codigo={self.codigo})'
14
15 # Definición de la función para registrar a una persona
16 def registrar_persona(nombre, apellido, codigo):
17     # Creación de un objeto Persona
18     persona = Persona(nombre, apellido, codigo)
19     # Devolución del objeto Persona
20     return [persona]
```

Figura: 3

REGISTRAR PERSONA

En esta etapa se registra a la persona ingresando los datos solicitados descritos en anexo 0-0-0-0 nombre, apellidos, código de estudiante o de ser el caso código de docente y seguidamente se presione el botón aceptar para inicia el proceso de detección de rostros en sistema en resumen se ingresan los datos requeridos y se procede a la toma de muestras

```
1 def registrar_persona(nombre, apellido, codigo):
2     # Cargar la imagen de la persona
3     imagen = cv2.imread('ruta_a_la_imagen.jpg')
4     # Detectar los rostros en la imagen
5     rostros = face_recognition.face_recognition_from_image(imagen)
6     # Crear una lista de rostros conocidos
7     rostros_conocidos = []
8     # Comparar el rostro con los rostros conocidos
9     resultados = face_recognition.compare_faces(rostros_conocidos, rostros)
10    # Devolver el resultado
11    return resultados
```

Figura: 4

CÓDIGO DE REGISTRO DE PERSONAS

En la imagen se describe el algoritmo diseñado para el registro de personas que se emplea en las capas de programación por debajo de la interface gráfica del usuario (cliente registrador)

```
6
7
8 import sqlite3
9 import os
10 import uuid
11
12
13
14 from caso_de_uso_registrar_person_02 import saveStudentInDb, createFolderForDataset, createSampleFacesStudent, create
15
16
17
18
19 def buildStudent(nombres, apellidos, codigo):
20     result = Student()
21
22     result.fullName = nombres
23     result.apellidos = apellidos
24     result.code = codigo
25
26     result.personGuid = str(uuid.uuid4())
27     result.folderGuid = str(uuid.uuid4())
28     result.personId = str(uuid.uuid4())
29
30     return result
31
32
33
34 class Form(tk.Tk):
```

Figura: 5

INICIO DE REGISTRO DE PERSONAS:

En esta etapa se puede lograr apreciar la ventana emergente de registro de datos al ejecutar el programa, se iniciará el programa para el ingreso de datos requeridos y la toma de muestras de imagen que se muestra en la captura de pantalla.

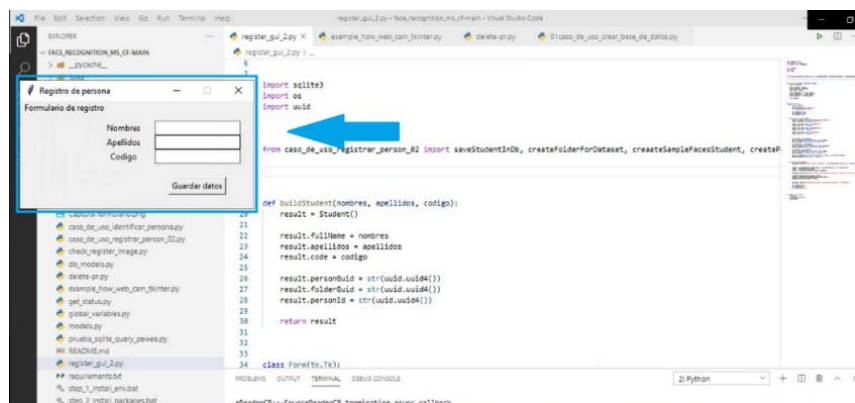


Figura: 6

INGRESO DE DATOS PARA PRUEBAS DEL PROTOTIPO

En esta etapa se realiza una prueba de funcionamiento del sistema ingresando datos al azar para revisar la etapa de reconocimiento, almacenamiento de datos así mismo.

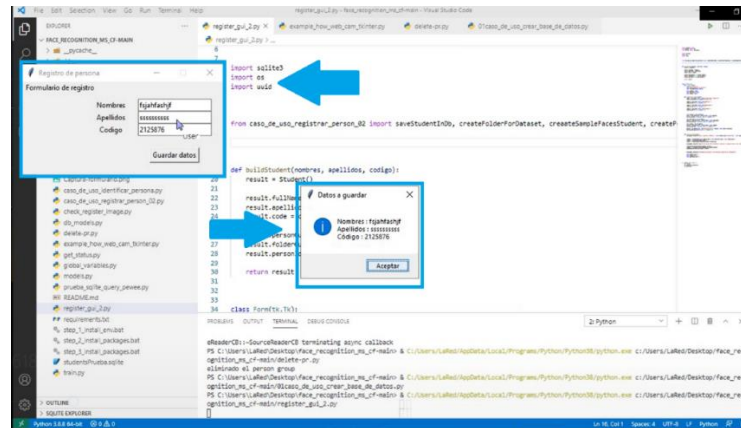


Figura: 7

VENTANA DE CONFIRMACIÓN INICIO DE TOMA DE MUESTRAS:

Esta ventana muestra la confirmación de ingreso de datos seguidamente anuncia el inicio de la toma de muestras para su respectivo envío hacia la plataforma de Microsoft azure.

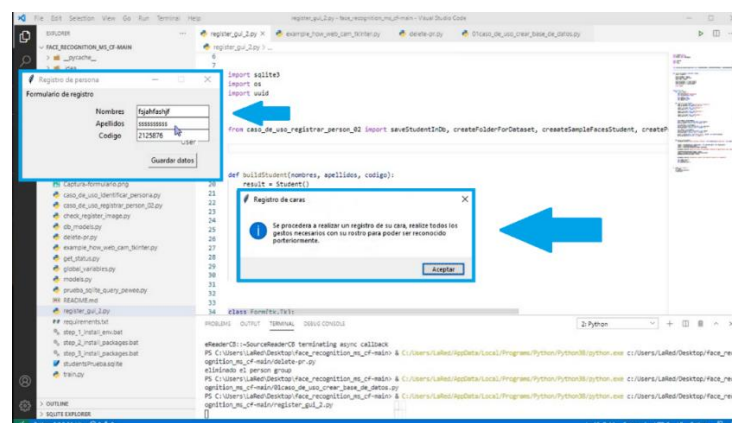


Figura: 8

VENTANA EMERGENTE INICIO DE TOMA DE MUESTRAS:

Esta ventana tiene la finalidad de realizar la toma de muestras del rostro del estudiante con la cámara de video asignada para el sistema, paralelamente en la toma de muestras aparece el recuadro con márgenes verdes donde se seleccionará el rostro de cada estudiante.

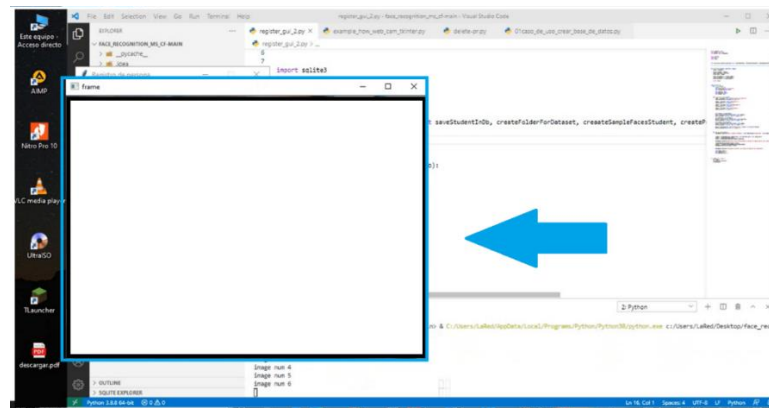


Figura: 9

TOMA DE MUESTRAS DE IMAGEN

Se toma las muestras del video tomado por la cámara en la parte, inferior del panel, se muestra las muestras tomadas por el software, unos máximos de 10 muestras serán tomadas en cada registro de estudiante y luego serán enviadas para su procesamiento en los servidores de Microsoft azure, la api del mismo (cognitive services) será la llave para el ingreso a los servidores y su entrenamiento en el mismo.

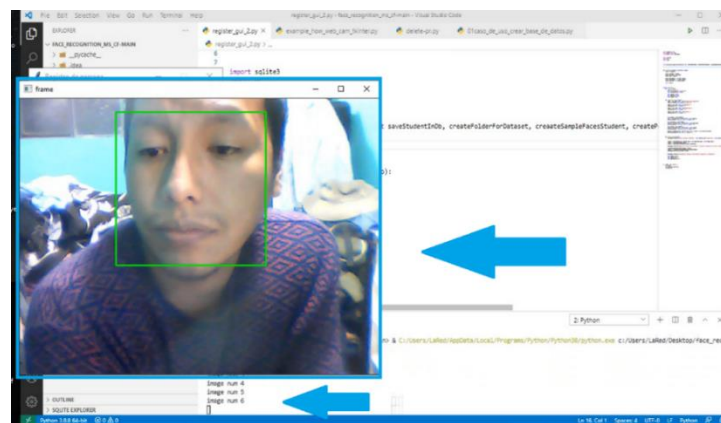


Figura: 10

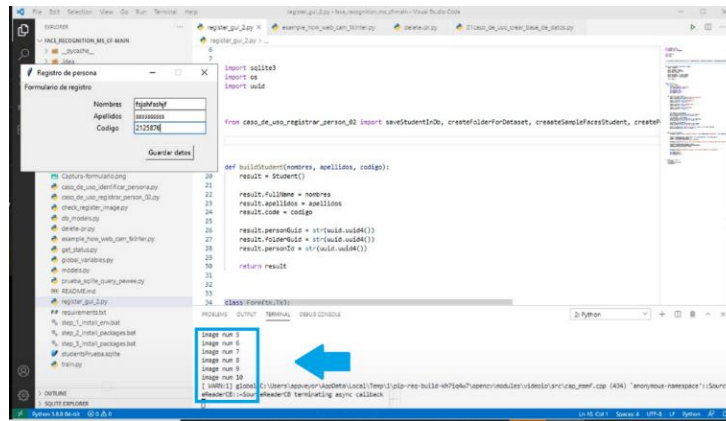


Figura: 11

FINALIZACIÓN DE TOMA DE MUESTRA:

En la imagen muestra el final de la toma de muestra, se muestra que la ventana emergente desaparece para su envío a los servidores a utilizar.

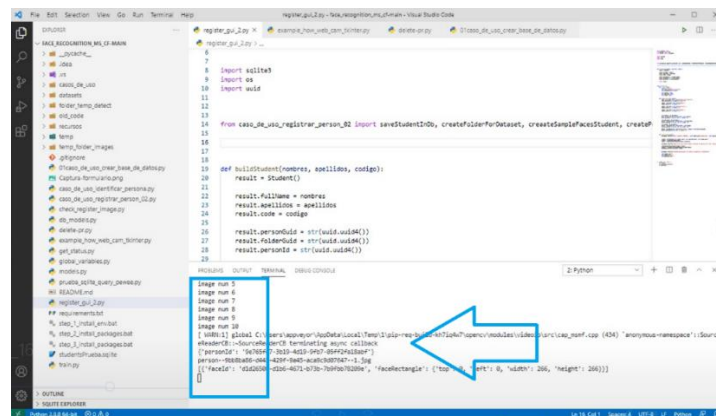


Figura: 12

INICIO DE REENVIO DE IMÁGENES TOMADAS (MUESTRAS TOMADAS HACIA LA APP DE MICROSOFT AZURE

En esta etapa se inicia el envío de imágenes muestreadas por el programa para su procesamiento en los servidores del mismo.

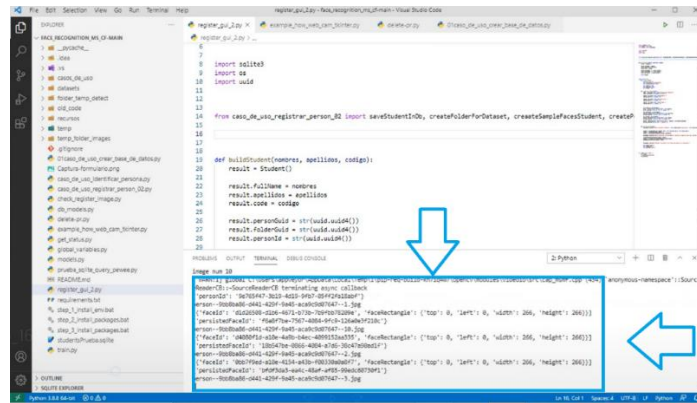


Figura: 13

FINALIZACION DE ENVIO DE MUESTRAS A LAPP AZURE MICROSOFT:

En la imagen se muestra la ventana emergente de confirmación que indica que a finalizado la carga de imágenes hacia los servidores utilizados

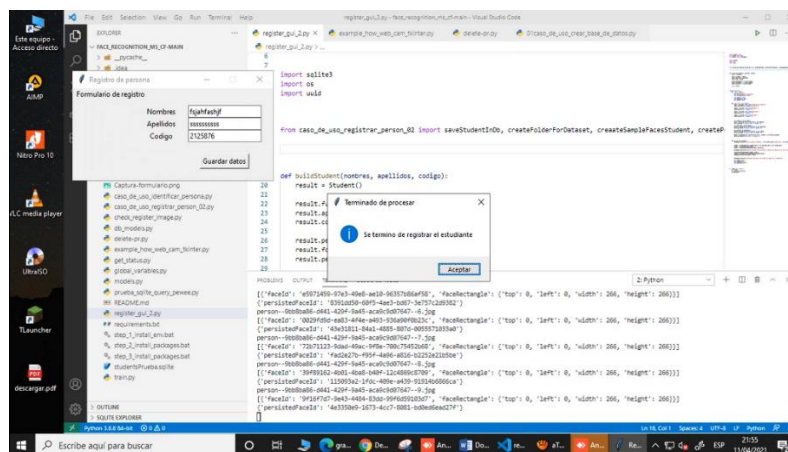


Figura: 14

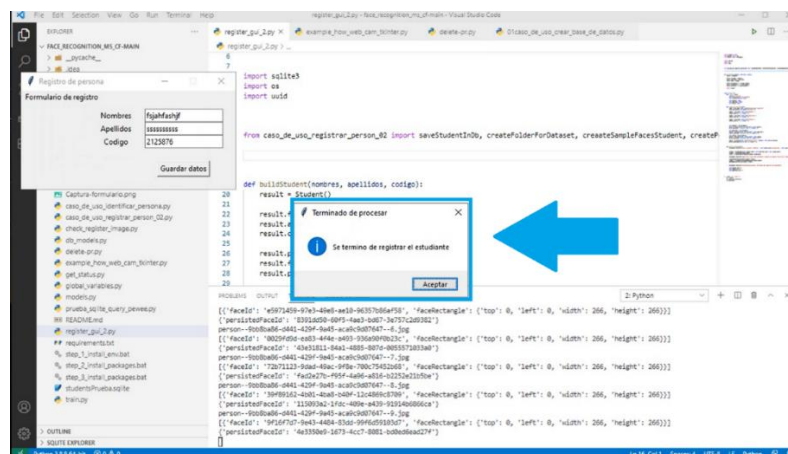


Figura: 15

FINALIZADO REGISTRO DE PERSONA

Como se muestra en la imagen la ventana emergente de registro queda en blanco para recibir los datos del siguiente personal y así sucesivamente para con cada uno de las personas de prueba en esta etapa.

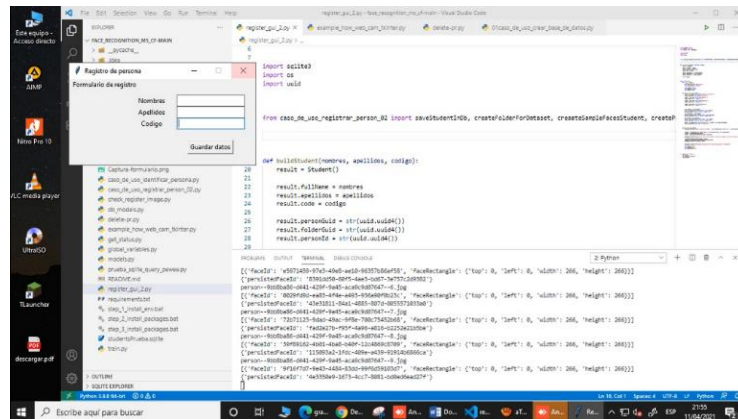


Figura: 16

Imágenes remarcando las casillas del box en blanco

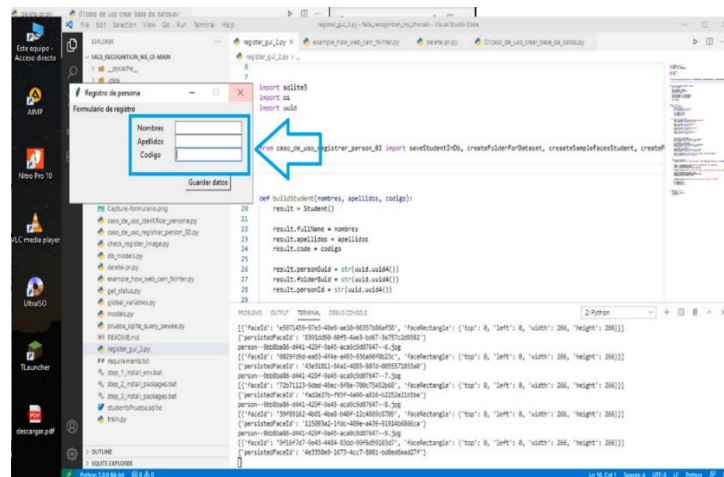


Figura: 17

FINALIZACIÓN DEL REGISTRO:

Seguidamente posterior a la finalización de la primera persona ya se puede lograr a realizar el reconocimiento de la persona registrada así confirmar el registro satisfactorio y el funcionamiento correcto del sistema y el algoritmo.

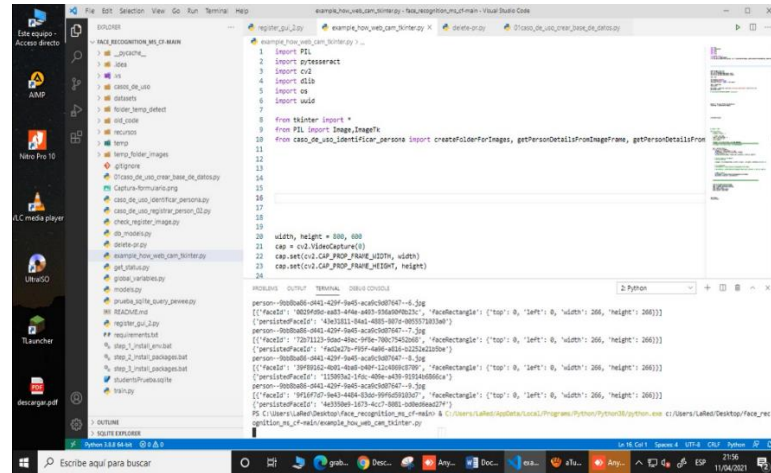


Figura: 18

EJECUCIÓN DEL PROGRAMA

Una vez finalizado la compilación del algoritmo en el sistema se procede a ejecutar la etapa de reconocimiento para con todo el grupo de personas registradas en el sistema con antelación.

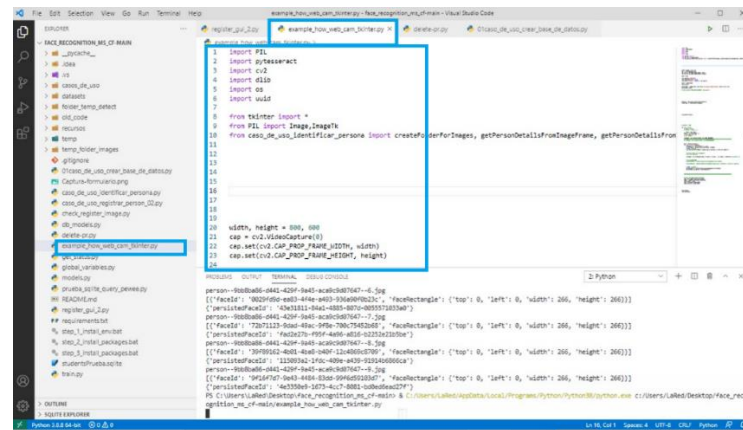


Figura: 19

Imagen detallada del código para la detección de rostros.

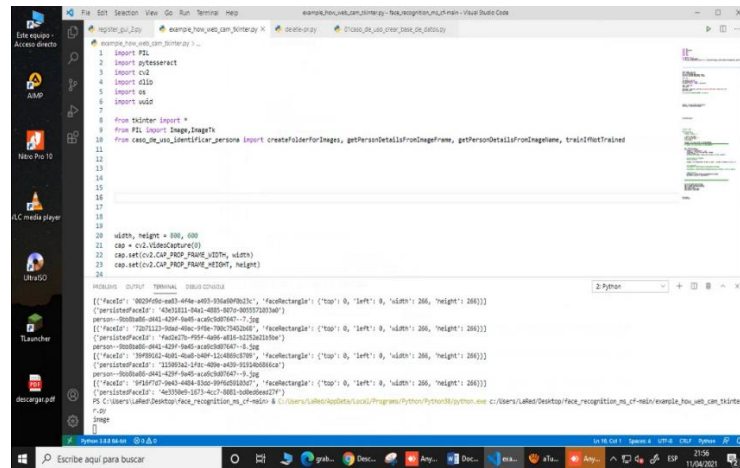


Figura: 20

EJECUTAMOS EL PROGRAMA DE RECONOCIMIENTO

Al iniciar el programa de reconocimiento se debe tomar en cuenta que el sistema aún se encuentra en etapa de prototipo por lo cual se tiene en cuenta que pueda lograr que existan algunos errores en dicho sistema como funcionamiento lento del algoritmo etapas que se queden en blanco, así como la probabilidad que el sistema no reconozca al rostro tomado por la cámara.

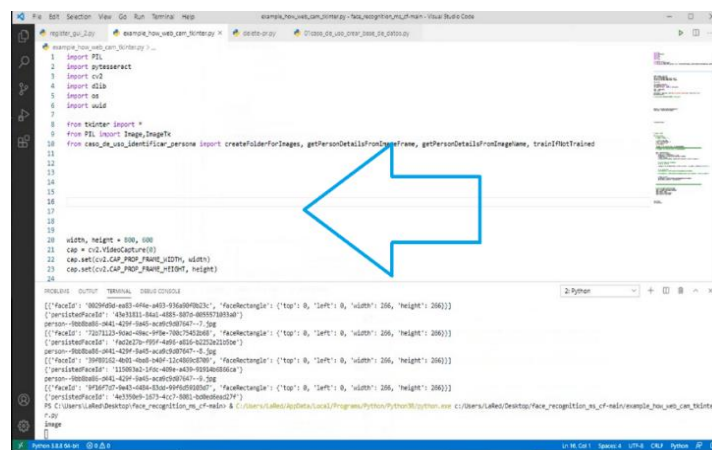


Figura: 21

INICIO DE RECONOCIMIENTO

La ventana emergente indica la inicialización del sistema por lo cual tomara la salida multimedia más próxima en el computador, el funcionamiento óptimo del sistema depende de muchos factores como son la calidad de video que se tiene para con la cámara

que realiza las capturas de imagen así también depende de la robustez de la tarjeta de video y el procesador del pc a utilizar.

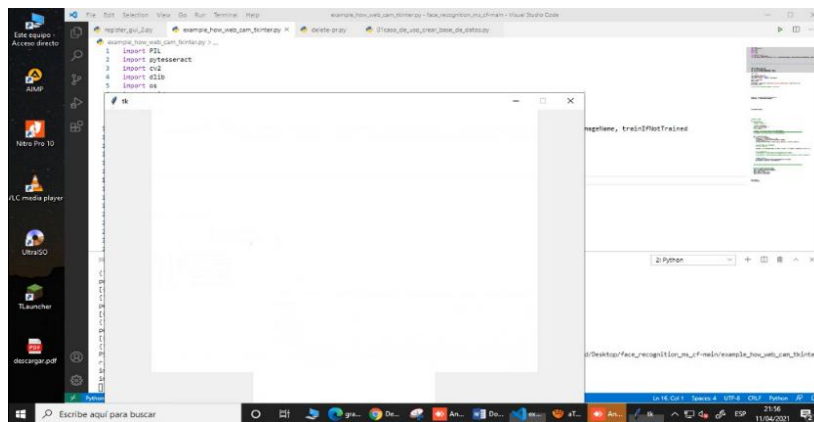


Figura: 22

VENTANA: ventana emergente para el inicio de captura de rostros del sistema

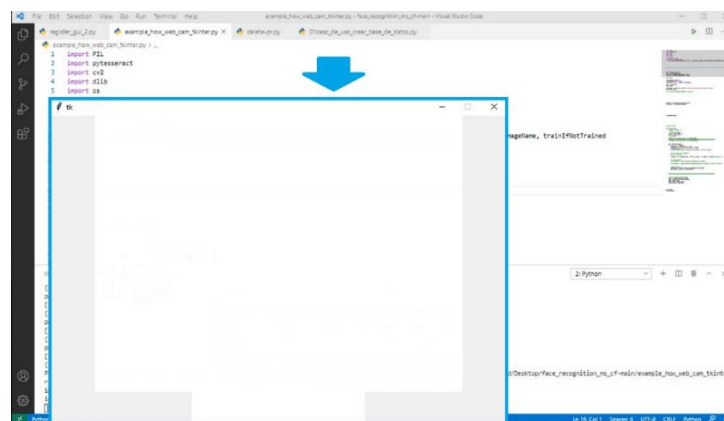


Figura: 23

MUESTRAS

Posterior a la captura de imágenes mediante la cámara de la computadora se realiza el envío de las capturas y al retorno se verifica la detección de rostros automático.

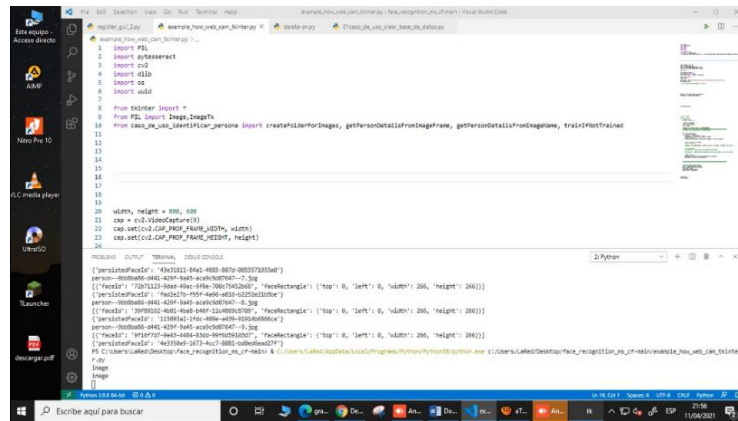


Figura: 24

DETECCIÓN

Se verifica la detección de rostros en la plataforma para posteriormente mostrar los resultados del funcionamiento del sistema.

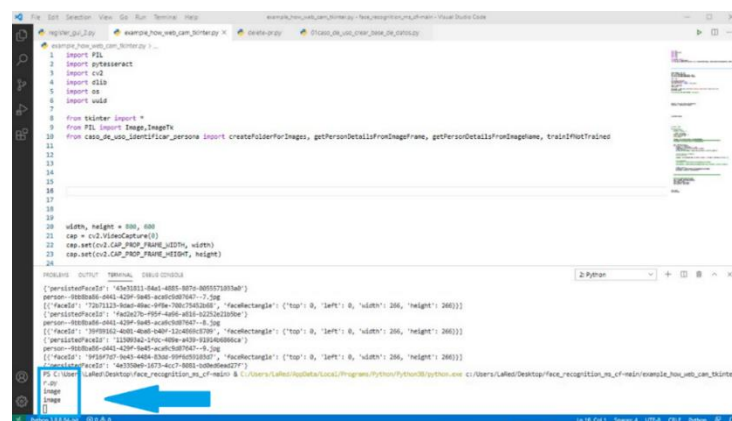


Figura: 25

DETECCIÓN DE ROSTROS

Una vez detectado el rostro del individuo el sistema deberá mostrar los datos detectados en la parte inferior de la ventana emergente para su lectura por el personal de la seguridad, para este caso deber de mostrar en la etapa de pruebas de funcionamiento los datos ingresados aleatoriamente para su verificación de que fueron registrado y asignados a las imágenes correspondientes



Figura: 26

CAMBIO DE ÁNGULO

en esta etapa de pruebas se muestra el cambio de enfoque de la persona que se encuentra en etapa de toma de muestras, al no recibir la luz ambiente de la forma en que fueron tomadas las imágenes se produce un reconocimiento erróneo por algunos segundos hasta que la luz se asemeje a la toma de muestras inicial, para esta situación se tiene que afirmar que el funcionamiento correcto del sistema se verá afectado el ambiente en el que fueron tomadas las pruebas así también como los accesorios de uso común para el rostro. Como recomendación la toma de imágenes y/o muestras se deberán de hacer en entornos controlados para evitar errores ambientales como el que se presentó en las pruebas de funcionamiento.



Figura: 27

RECONOCIMIENTO DE ROSTROS

Cambiado el enfoque los errores mencionados en la anterior imagen se corrigen ya que se encuentran en una aproximación ambiental aceptable para el sistema por lo cual se verifica los datos aleatorios ingresados en un inicio

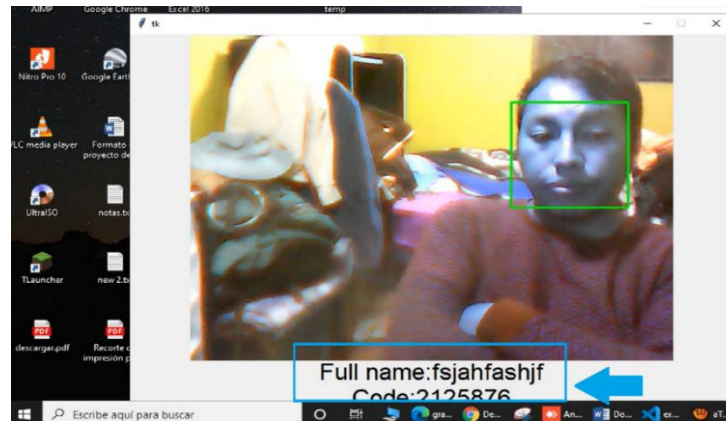


Figura: 28

RECONOCIMIENTO:

En la etapa de reconocimiento se debe de recalcar que se continúa apareciendo pequeños flap de la imagen dicha situación debe a que en el momento de las pruebas se presentaron inconvenientes con el tráfico del internet contratado ya que al no ofrecer una estabilidad de línea balanceada al 50% aproximadamente se verán estos errores más comúnmente de lo que se quiere para nuestro sistema.

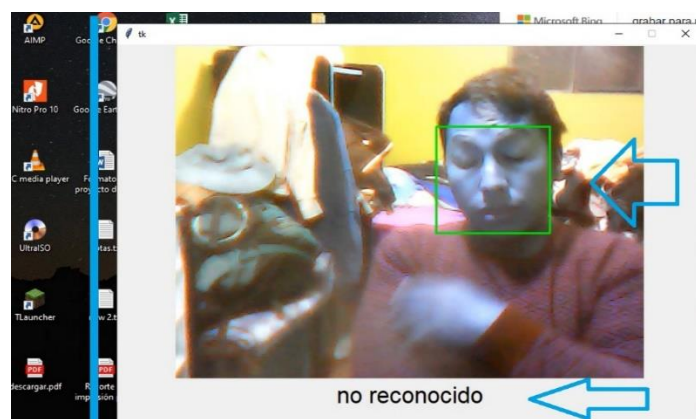


Figura: 29

FINALIZACIÓN

Una vez finalizado el proceso de identificación por el personal de control de acceso el usuario deberá de presionar la tecla Esc y/o cerrar la ventana TK con el botón en la parte superior derecha, de esta manera se cierra el sistema y se procede a guardar los datos automáticamente sin la necesidad de manipular ningún otro objeto o herramienta de la plataforma.

```
#!/usr/bin/env python
import cv2
import tkinter as tk
import numpy as np

# Configuración de la ventana TK
root = tk.Tk()
root.title("Sistema de Reconocimiento Facial")
root.geometry("400x300")

# Etiqueta para mostrar el resultado
label = tk.Label(root, text="Resultado de la identificación")
label.pack()

# Botón para iniciar el proceso
button = tk.Button(root, text="Iniciar")
button.pack()

# Función para capturar y procesar un frame
def capture_frame():
    # Capturar un frame de la cámara
    frame = cv2.cvtColor(cv2.VideoCapture(0).read()[1], cv2.COLOR_BGR2RGB)

    # Mostrar el frame en una ventana separada
    cv2.imshow("Frame", frame)

    # Esperar a que se presione una tecla
    cv2.waitKey(1)

    # Guardar el frame en un archivo
    cv2.imwrite("frame.jpg", frame)

# Función para mostrar el resultado de la identificación
def show_result():
    # Mostrar el resultado de la identificación
    label.config(text="Resultado de la identificación")

# Asignar eventos a los botones
button.config(command=capture_frame)
button.config(command=show_result)

# Ejecutar el bucle principal de TK
root.mainloop()
```

Figura: 30

V.D.: Sistema de Reconocimiento Facial

V.I.: Control de acceso de las personas