

FACULTAD DE INGENIERÍA

Escuela Académico Profesional de Ingeniería de Sistemas e Informática

Tesis

**Reconocimiento facial para la identificación de los
alumnos en exámenes finales en la modalidad presencial
de la Universidad Continental - Huancayo, 2021**

Doris Ida Galindo Taype
Sandy Judith Huaranga Gallardo
Gladys Luz Samaniego Canales

Para optar el Título Profesional de
Ingeniero de Sistemas e Informática

Huancayo, 2021

Repositorio Institucional Continental
Tesis digital



Esta obra está bajo una Licencia "Creative Commons Atribución 4.0 Internacional" .

AGRADECIMIENTOS

Agradecer al ingeniero Daniel Gamarra Moreno, docente de la E.A.P de Ingeniería de Sistemas e Informática por su dedicación y apoyo brindado incondicionalmente para finalizar de manera satisfactoria esta tesis.

DEDICATORIA

A Dios, por permitirme llegar hasta esta etapa de mi formación profesional y a mis padres, por su apoyo incondicional en cada proceso.

Doris Ida Galindo Taype

Le dedico el presente trabajo de investigación a Dios por haber permitido llegar hasta aquí, a los ingenieros y a mi familia por ser el soporte y ayuda que me han brindado durante la elaboración de este proyecto y motivarme a continuar mejorando cada día.

Sandy Judith Huaranga Gallardo

El presente proyecto es dedicado a mi familia, a mis padres por todo el apoyo brindado en todo el proceso de mi formación profesional.

Gladys Luz Samaniego Canales

ÍNDICE DE CONTENIDO

AGRADECIMIENTOS.....	II
DEDICATORIA.....	III
ÍNDICE DE CONTENIDO.....	IV
ÍNDICE DE FIGURAS.....	XI
RESUMEN.....	XIV
ABSTRACT.....	XV
INTRODUCCIÓN.....	XVI
CAPÍTULO I PLANTEAMIENTO DEL ESTUDIO.....	18
1.1. Planteamiento y formulación del problema	18
1.2. Formulación del problema.....	21
1.3. Objetivo general y objetivos específicos	22
1.4. Justificación e importancia	22
1.5. Limitaciones y alcance del proyecto.....	23
CAPÍTULO II	25
MARCO TEÓRICO	25
2.1. Antecedentes del problema	25
2.1.1. Antecedentes internacionales	25
2.1.2. Antecedentes nacionales	28
2.2. Bases teóricas	30
2.2.1. Suplantación de identidad.....	30
2.2.2. Tipos de suplantación	30
2.2.3. Identidad.....	31
2.2.4. Identificación.....	31
2.2.5. Casos de suplantación de identidad	31
2.2.6. Inteligencia artificial.....	31
2.2.7. Áreas de aplicación de la IA.....	32
2.2.8. Aprendizaje automático (Machine Learning)	34
2.2.9. Aprendizaje profundo.....	36
2.2.10. Reconocimiento facial.....	36
2.2.11. Como se inició el reconocimiento facial	37
2.2.12. Como funciona el reconocimiento facial.....	38

2.2.13. Redes neuronales convolucionales (CNN)	39
2.2.14. Estructura de las redes neuronales convolucionales	40
2.2.15. Deep Learning o aprendizaje profundo	40
2.2.16. Sistema de reconocimiento facial	41
2.2.17. Instalación de cámara para reconocimiento facial.....	42
2.3. Definición de términos básicos.....	42
CAPÍTULO III METODOLOGÍA.....	45
3.1. Metodología aplicada para el desarrollo de la solución	45
2.3.1. Flujo de trabajo	45
3.1.1. Realizar explícitas las políticas de proceso.....	48
3.2. Desarrollo del sistema de escritorio mediante Kanban.....	49
3.2.1. Visualizar el flujo de trabajo	49
3.2.2. Limitar el trabajo en progreso (WIP)	49
3.2.3. Gestionar el flujo	50
3.2.4. Hacer políticas explícitas.....	50
3.2.5. Implementar circuitos de retroalimentación	51
3.2.6. Mejorar colaborativamente.....	51
3.2.7. Roles de Kanban	51
3.3. Metodología del reconocimiento facial	51
3.3.1. Fase del entrenamiento	52
3.3.2. La fase de ejecución	56
3.4. La matriz de confusión	59
3.4.1. Verdadero positivo (TP)	61
3.4.2. Verdadero negativo (TN).....	61
3.4.3. Falso positivo (FP): error de tipo 1	61
3.4.4. Falso negativo (FN): error de tipo 2	61
3.4.5. Métricas de la matriz de confusión	62
3.5. Alcance del proyecto.....	62
CAPÍTULO IV ANÁLISIS Y DISEÑO DE LA SOLUCIÓN.....	63
4.1. Identificación de los requerimientos	63
4.1.1. Usuarios que intervienen en el sistema.....	63
4.1.2. Historias de usuarios	64
4.1.3. Requerimientos funcionales.....	65
4.1.4. Especificaciones de los requerimientos	71

4.1.5. Requerimientos no funcionales	73
4.1.6. Requerimientos no funcionales del <i>software</i>	74
4.1.7. Requerimientos no funcionales de la organización	76
4.2. Análisis de la solución.....	77
4.2.1. Propuesta de la solución.....	77
4.2.2. Análisis de las especificaciones de los requerimientos funcionales ...	78
4.2.3. Análisis técnico y económico	81
4.2.4. Viabilidad de la propuesta.....	84
4.3. Diseño	86
4.3.1. Diseño de la arquitectura del sistema	86
4.3.2. Herramientas	88
4.3.3. Diseño del reconocimiento facial.....	89
4.3.4. Diseño de interfaz y navegación	90
4.3.5. Interfaz de login	90
4.3.6. Pantalla de principal.....	91
4.3.7. Pantalla administrar usuarios.....	92
4.3.8. Formulario registrar usuario	92
4.3.9. Pantalla administrar estudiantes	93
4.3.10. Formulario registrar estudiante	93
4.3.11. Formulario reconocimiento facial	94
4.3.12. Formulario de asistencia.....	94
4.3.13. Diseño de la base de datos	95
CAPÍTULO V CONSTRUCCIÓN.....	97
5.1. Construcción.....	97
5.5.1. Modelamiento de la red neuronal artificial y el Machine Learning	97
5.5.2. Creando una imagen integral.....	98
5.5.3. Extracción de características	100
5.5.4. Escritura de código de red neuronal	101
5.5.5. Interfaz de inicio de sesión.....	104
5.5.6. Interfaz de la pantalla principal	107
5.5.7. Interfaz administrar usuarios.....	108
5.5.8. Interfaz registrar usuario	108
5.5.9. Interfaz administrar estudiantes	109
5.5.10. Formulario registrar estudiante	109

5.5.11. Formulario reconocimiento facial	110
5.5.12. Diagrama de clases	112
5.5.13. Construcción de la base de datos	113
CAPÍTULO VI	117
PRUEBAS Y RESULTADOS	117
6.1. Pruebas y resultados	117
6.1.1. Pruebas del modelo de reconocimiento facial	117
6.2. Pruebas y resultados del modelo de reconocimiento facial	118
6.2.1. Prueba 01: Gladys Samaniego	118
6.2.2. Prueba 02: Doris Galindo	121
6.2.3. Prueba 03: Sandy Huaranga	124
6.2.4. Prueba 04: Joel Cuadros	127
6.2.5. Prueba 05: Jorge Bizarro	129
6.2.6. Resultados	131
6.2.7. Pruebas de software	132
6.2.8. Pruebas de aceptación	132
6.2.9. Plan de pruebas de aceptación	133
6.2.10. Ejecución de resultados	134
6.2.11. Aplicación de la matriz de confusión	142
6.2.12. Aplicación de las métricas de evaluación	143
6.3. Discusión de resultados	145
CONCLUSIONES	148
RECOMENDACIONES	149
REFERENCIAS BIBLIOGRÁFICAS	150
ANEXOS	157

ÍNDICE DE TABLAS

Tabla 1. Porcentaje de casos de suplantación	20
Tabla 2. Frecuencia de casos de suplantación.....	20
Tabla 3. Principales ámbitos de aplicación de los sistemas de inteligencia artificial.....	33
Tabla 4. Políticas DOD y DOR	50
Tabla 5. Usuarios del sistema	64
Tabla 6. Historias de usuarios	64
Tabla 7. Requerimientos funcionales	65
Tabla 8. Requerimientos funcionales – Iniciar sesión.....	66
Tabla 9. Requerimientos funcionales – Gestión de usuarios	66
Tabla 10. Requerimientos funcionales – Registrar estudiante	67
Tabla 11. Requerimientos funcionales – Capturar rostro.....	67
Tabla 12. Requerimientos funcionales – Guardar información	67
Tabla 13. Requerimientos funcionales – Registrar asistencia.....	68
Tabla 14. Requerimientos funcionales – Guardar imagen	68
Tabla 15: Requerimientos funcionales – Reconocimiento facial	68
Tabla 16. Requerimientos funcionales – Reconocimiento facial en tiempo real.....	69
Tabla 17: Requerimientos funcionales – Verificar identidad	69
Tabla 18. Requerimientos funcionales – Permitir ingreso.....	69
Tabla 19. Requerimientos funcionales – Listar usuarios.....	70
Tabla 20. Requerimientos Funcionales – Listar estudiantes.....	70
Tabla 21. Requerimientos funcionales – Manual de usuario.....	70
Tabla 22. Especificaciones de los requerimientos funcionales	71
Tabla 23. Requerimientos no funcionales del software.....	74
Tabla 24. Requerimientos no funcionales – Base de datos	74
Tabla 25. Requerimientos no funcionales – Requerimientos de usabilidad	75
Tabla 26. Requerimientos no funcionales – Requerimientos de rendimiento.....	75

Tabla 27. Requerimientos no funcionales – Requerimientos de seguridad.....	75
Tabla 28. Requerimientos no funcionales – Requerimientos de eficiencia ...	75
Tabla 29. Requerimientos no funcionales – Requerimientos de confidencialidad	75
Tabla 30. Requerimientos no funcionales – Requerimientos de integridad.....	76
Tabla 31. Requerimientos no funcionales – Requerimientos de disponibilidad	76
Tabla 32. Requerimientos no funcionales de la organización	76
Tabla 33. Requerimiento funcional 001, Iniciar Sesión	78
Tabla 34. Requerimiento funcional 002, Registrar usuario	79
Tabla 35. Requerimiento funcional 003, Registrar estudiante.....	79
Tabla 36. Requerimiento funcional 004, Capturar rostro del estudiante.....	79
Tabla 37. Requerimiento funcional 005, Reconocimiento facial.....	80
Tabla 38: Requerimiento funcional 005, Registro de asistencia	80
Tabla 39. Requerimientos de hardware.....	81
Tabla 40. Herramientas utilizadas en el desarrollo	82
Tabla 41. Costos de hardware	82
Tabla 42. Costos de software	83
Tabla 43. Costos de personal.....	83
Tabla 44: Costos operativos.....	83
Tabla 45. Costo total	84
Tabla 46. Tensor Flow vs OpenCV.....	84
Tabla 47. Restricciones técnicas	86
Tabla 48. Tablas de la base de datos.....	96
Tabla 49. Clase tipo de usuario	112
Tabla 50. Clase Usuario	112
Tabla 51. Clase estudiante	113
Tabla 52. Clase asistencia	113
Tabla 53. Descripción de la tabla Tipo de Usuario	114
Tabla 54. Descripción de la tabla Usuario	114
Tabla 55: Descripción de la tabla Estudiante.....	115
Tabla 56. Descripción de la tabla Asistencia-Estudiante	115

Tabla 57. Resultados de las pruebas	132
Tabla 58. Plan de pruebas de aceptación	133
Tabla 59: Ejecución del plan de pruebas	134
Tabla 60. Caso de prueba - Registrar estudiante	135
Tabla 61. Caso de prueba - Capturar rostro	136
Tabla 62. Caso de prueba - Capturar rostro con lentes	138
Tabla 63. Caso de prueba - Capturar rostro con lentes	139
Tabla 64. Caso de prueba realizar reconocimiento	140
Tabla 65. Caso de prueba – Registrar asistencia	141

ÍNDICE DE FIGURAS

Figura 1. Denuncias recibidas en la DIVINDAT de enero a abril 2021 -----	19
Figura 2. Tipos de aprendizaje automático -----	35
Figura 3. Red Neuronal Convolucional-----	39
Figura 4. Redes Neuronales convolucionales, -----	40
Figura 5. Aprendizaje profundo -----	41
Figura 6. Reconocimiento facial -----	41
Figura 7. Fases del ciclo de desarrollo-----	45
Figura 8. Actividades que se desarrollaran -----	46
Figura 9. Actividades en proceso-----	47
Figura 10. Actividades finalizadas-----	47
Figura 11. Fechas de entrega de cada una de las tarjetas del tablero Kanban -----	48
Figura 12. Prácticas de Core Kanban -----	49
Figura 13. Políticas explícitas-----	50
Figura 14. Fases y Etapas de la Metodología, -----	52
Figura 15. Creación del DataSet del modelo -----	52
Figura 16. Modelo de entrenamiento de la red neuronal artificial, -----	53
Figura 17. Formato de presentación de la fotografía para entrenamiento de los algoritmos-----	54
Figura 18. Especificaciones de la imagen a utilizar-----	54
Figura 19. Entrenamiento de la red neuronal en Python -1-----	55
Figura 20. Entrenamiento de la red neuronal en Python -2-----	55
Figura 21. Entrenamiento del modelo en Python utilizando el editor de código visual studio code-----	56
Figura 22. Proceso para crear un clasificador de rostros -----	57
Figura 23. Proceso para crear un clasificador de rostros empleando Haar-Cascade -----	57
Figura 24. Ejemplo de detección de rostros -----	58
Figura 25. Reconocimiento facial-----	59
Figura 26. Matriz de confusión -----	60

Figura 27. Entendiendo el verdadero positivo, el verdadero negativo, el falso positivo y el falso negativo en una matriz de confusión -----	61
Figura 28. Interpretación de la matriz de confusión-----	62
Figura 29. Requerimientos no funcionales -----	73
Figura 30. Modelo conceptual de la Solución -----	78
Figura 31. Arquitectura del software -----	87
Figura 32. Arquitectura del hardware-----	88
Figura 33. Pasos para el reconocimiento facial-----	90
Figura 34. Mockup de inicio sesión-----	91
Figura 35. Mockup de la pantalla principal -----	91
Figura 36. Mockup Pantalla administrar usuarios -----	92
Figura 37. Mockup registrar usuario -----	92
Figura 38. Mockup pantalla administrar estudiante -----	93
Figura 39. Mockup formulario registrar estudiante -----	93
Figura 40. Mockup reconocimiento facial-----	94
Figura 41. Mockup registro de asistencia -----	94
Figura 42. Diseño lógico -----	95
Figura 43. Tabla de la base de datos-----	96
Figura 44. Núcleos utilizados en Haar Cascade -----	98
Figura 45. Imagen de entrada – imagen integral -----	98
Figura 46. Imagen de entrada – imagen integral -----	99
Figura 47. Imagen integral en posición (x, y)-----	99
Figura 48. Filtros Haar rotados, trasladados y con cambios de escala -----	100
Figura 49. Convolución del filtro Haar con una imagen integral -----	100
Figura 50. Importación de módulos cv2, face_recognition y os-----	101
Figura 51. Importación de módulos numpy, sys y mysql -----	101
Figura 52. Carga de imagen -----	101
Figura 53. Carga de imagen -----	102
Figura 54. Especificaciones del módulo DetectMultiScale en OpenCV -----	102
Figura 55. Argumentos del módulo DetectMultiScale en OpenCV-----	103
Figura 56. Resultado del módulo DetectMultiScale en OpenCV -----	103
Figura 57: Estructura del directorio-----	104
Figura 58. Interfaz inicio de sesión-----	104
Figura 59. Código de inicio de sesión -1-----	105

Figura 60. Código de inicio de sesión -2-----	106
Figura 61: Clase de la tabla usuario -----	106
Figura 62. Menú principal-----	107
Figura 63. Código de menú principal -----	107
Figura 64: interfaz administrar usuarios-----	108
Figura 65. Interfaz registrar usuario-----	108
Figura 66. Interfaz administrar estudiante, construcción de la interfaz -----	109
Figura 67. interfaz registrar estudiantes -----	109
Figura 68. Código del registro estudiantes -----	110
Figura 69. Interfaz de reconocimiento facial -----	110
Figura 70. Código de reconocimiento facial en C#-----	111
Figura 71. Código de reconocimiento facial en JavaScript -----	111
Figura 72. Creación del DataSet-----	111
Figura 73. Diagrama de clases-----	112
Figura 74. Matriz de confusión -----	143
Figura 75. Aplicación de la matriz de confusión -----	143
Figura 76. Valores de la matriz de confusión -----	144
Figura 77. Formulas para calcular las métricas de la matriz de confusión-----	144
Figura 78. Pregunta N ^a 1 de la encuesta -----	158
Figura 79. Pregunta N ^a 2 de la encuesta -----	158
Figura 80. Pregunta N ^a 3 de la encuesta -----	158
Figura 81. Pregunta N ^a 4 de la encuesta -----	159
Figura 82. Pregunta N ^a 5 de la encuesta -----	159
Figura 83. Pregunta N ^a 6 de la encuesta -----	159
Figura 84. Pregunta N ^a 7 de la encuesta -----	160
Figura 85. Pregunta N ^a 8 de la encuesta -----	160

RESUMEN

En la universidad Continental, observamos que la suplantación de identidad durante las fechas de exámenes finales son frecuentes, y esto se da mayormente en las asignaturas generales, este problema se viene dando cada año, lo que genera un estado de preocupación; por ello, el objetivo de la presente investigación es desarrollar un sistema de escritorio mediante la metodología Kanban para el reconocimiento facial y la identificación de los alumnos en exámenes finales en la modalidad presencial de la universidad Continental en la ciudad de Huancayo; considerando la gran importancia desarrollar dicha investigación, ya que ayuda a disminuir el problema presentado.

La metodología que se ha usado para el desarrollo del sistema es Kanban, gestionado con la herramienta Trello, lo cual ayudó a tener un mejor control de las actividades que se desarrollaron. Para la recolección de datos se utilizó la encuesta tanto a estudiantes como docentes.

Luego de la respectiva construcción del modelo de reconocimiento facial, se hizo la ejecución de pruebas con una muestra de 5 estudiantes cada uno con 50 fotografías, se capturaron rostros objetivos en diferentes escenarios como: sin mascarilla-sin lentes, con mascarilla- sin lentes, sin mascarilla-con lentes, a partir de los cuales se construyó una base de datos de imágenes.

Los resultados hallados en la evaluación del modelo del reconocimiento facial por medio de la matriz de confusión es el 93 % ante la probabilidad de similitud.

Finalmente, se concluye que el sistema de reconocimiento facial es favorable para reducir la suplantación de identidad en los exámenes.

Palabras claves: reconocimiento facial, aprendizaje profundo, red neuronal, face recognition, openCV

ABSTRACT

At the Continental University we observe that identity theft during final exam dates is frequent, and this occurs mostly in general subjects, this problem has been occurring every year, which generates a state of concern, for them the objective of The present investigation is to develop a desktop system using the Kanban methodology for facial recognition for the identification of students in final exams in the face-to-face modality of the Continental University in the city of Huancayo; considering the great importance of developing such research, since it helps to reduce the problem presented.

The methodology that has been used for the development of the system is Kanban, managed with the Trello tool, which helped to have a better control of the activities that were developed and for the collection of data, the survey of both students and teachers was used.

After the respective construction of the facial recognition model, tests were carried out with a sample of 5 students each with 50 photographs, objective faces were captured in different scenarios such as: without mask-without glasses, with mask-without glasses, without mask-with lenses, from which an image database was built.

The results found in the evaluation of the facial recognition model through the confusion matrix is 93% before the probability of similarity.

Finally, it is concluded that the facial recognition system is favorable to reduce identity theft in exams.

Keywords: Facial Recognition, Deep Learning, Neural Network, Face Recognition, OpenCV

INTRODUCCIÓN

Durante los últimos años, se han desarrollado sistemas de reconocimiento facial para mayor seguridad informática, seguridad pública, económica, asistencia laboral, control y monitoreo en el ambiente, entre otros; debido a que en el mundo hay muchos casos de suplantación de identidad, incluso se ha duplicado los casos, aún más en la pandemia. Sin embargo, a medida que la tecnología avanza, el reconocimiento facial enfrenta varios desafíos, como las oclusiones que no permiten realizar una precisión exacta y lo otro son los cambios en la iluminación que pueden causar cambios dramáticos en los rasgos faciales.

En Perú, cada vez más, se elevan las denuncias por suplantación de identidad y la única recomendación para no ser víctima de estafas, suplantaciones o fraudes es que proteger la privacidad de la cuenta o perfil en las redes sociales.

Actualmente, la suplantación de identidad se está convirtiendo en un problema que afecta en todas las áreas. A nivel local, se evidenció que en la universidad Continental uno de los problemas principales es que existe la suplantación de identidad en la toma de exámenes finales, mayormente en las asignaturas generales y se viene dando año tras año, lo que genera un estado de preocupación ante este problema.

La presente tesis aplica herramientas y procedimientos para lograr un reconocimiento facial efectivo, donde propone un sistema de escritorio de reconocimiento facial que incluye algunos rasgos de oclusión como por ejemplo cuando el estudiante cuenta con los accesorios como mascarillas y lentes, que de cierta forma afectan el resultado en la identificación de los estudiantes de pruebas.

Las pruebas se realizaron con cinco estudiantes en diferentes escenarios, formando subcarpetas con sus respectivos nombres y apellidos de cada

estudiante con 10 imágenes respectivamente para el entrenamiento, validación del modelo.

La tesis realiza el reconocimiento facial en los siguientes casos. 1) con el uso de mascarilla, 2) con el rostro descubierto y 3) con el uso de lentes.

A continuación, se desarrollará un resumen de los cinco capítulos de investigación.

Capítulo I, se desarrollará el planteamiento del problema, formulación del problema, objetivo general, objetivos específicos, justificación práctica y justificación metodológica.

Capítulo II, se presenta el marco teórico que brindará los antecedentes internacionales, antecedentes nacionales, bases teóricas y definición de términos básicos.

Capítulo III, se desarrollará la metodología, metodología aplicada para el desarrollo de la solución.

Capítulo IV, se detalla el análisis y diseño de la solución, identificación de los requerimientos funcionales y no funcionales, análisis de la solución y el diseño.

Capítulo V, se detalla la construcción, especificando el desarrollo del sistema web, utilizando la metodología Kanban, las pruebas y finalmente los resultados.

CAPÍTULO I

PLANTEAMIENTO DEL ESTUDIO

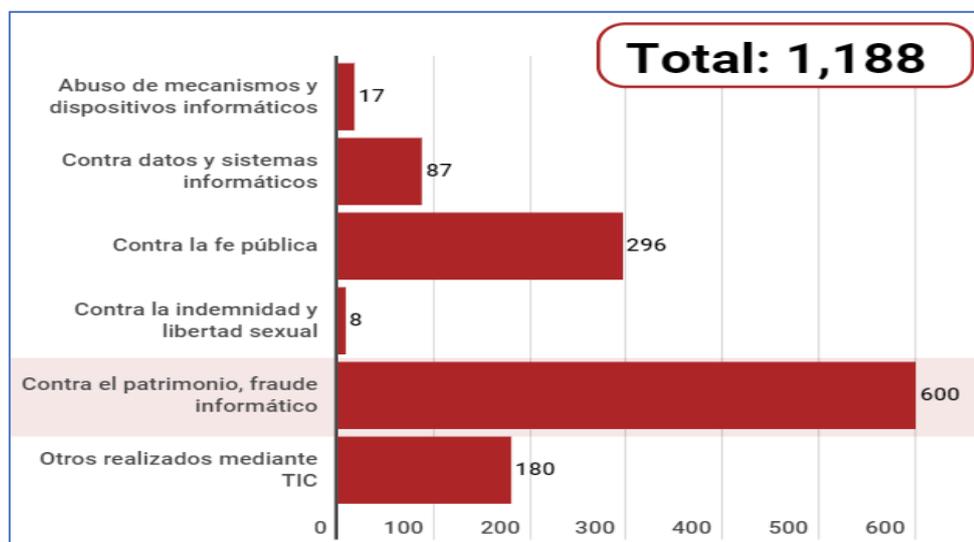
1.1. Planteamiento y formulación del problema

La inteligencia artificial aborda la simulación del comportamiento inteligente que implica que los dispositivos pueden pensar por su propia cuenta y cada vez se encuentra más involucrada en cualquier ámbito de la vida donde la tecnología se adapta a las necesidades humanas. En los últimos años, los sistemas de reconocimiento facial se han desarrollado principalmente para uso de seguridad, asistencia laboral, control y monitoreo en el ambiente, entre otros; debido que a nivel mundial hay muchos casos de suplantación de identidad y es un flagelo que está en alza en estos últimos tiempos.

En América Latina, la suplantación de identidad se ha convertido en el delito cibernético más común durante la pandemia de Covid-19, el ataque principal es el phishing donde los piratas informáticos intentan engañar a las personas para que descarguen el malware con la finalidad de obtener su información personal mediante correo o redes sociales, explotan la curiosidad, confianza y/o ansiedad. Las cifras indican que los países más atacados en Latinoamérica son Venezuela, Brasil, EE. UU, Colombia y México. (1)

En el Perú, en el año 2021 se realizó una investigación a 1188 denuncias y los más frecuentes fueron el fraude informático y la suplantación de identidad donde las formas más frecuentes fue el phishing que hacen todo lo posible para ilusionar y defraudar a las personas con paginas falsas. También suplantan la identidad en los perfiles de redes sociales para engañar a los

amigos y/o familiares con la condición de que les hagan préstamos o les depositen por temas de salud u alguna otra necesidad. En la figura 1, se muestran el número de casos frecuentes de fraude informático.



**Figura 1. Denuncias recibidas en la DIVINDAT de enero a abril 2021
Tomado de diario El Peruano (2)**

Por otro lado, la prevención de plagio es una tarea que acompañó en la educación presencial siempre; por eso, se evidenció que en la universidad Continental un problema que existe es la suplantación de identidad durante las fechas de exámenes finales, mayormente en las asignaturas generales, este problema se viene dando año tras año, lo que genera un estado de preocupación.

Este tipo de situaciones se realiza porque muchos estudiantes no toman el interés a dichos cursos y para ellos es fácil de pedir apoyo a estudiantes que son más destacados. Y la mayoría de los casos se da porque el docente no llega a conocer en su totalidad a los estudiantes y pierde el control del entorno y esto aumenta cuando el número de estudiantes es mayor. En ese punto es donde se requiere aplicar alguna estrategia para disminuir la suplantación.

Los efectos que causa este problema es la indignación de los estudiantes que se esmeran por obtener una nota aprobatoria, que a otros se les facilita suplantando con otros estudiantes, a consecuencia de eso se ha decidido

elaborar el sistema de reconocimiento facial para la identificación de los estudiantes y poder reducir la estadística de suplantación.

En la siguiente tabla 1, se evidencia según la encuesta realizada a 86 personas, el porcentaje de casos que se detectó una suplantación de identidad por semestre es el siguiente: el 22.5 % se detecta una suplantación de 1 a 3 casos, el 6.8 % es de 3 a 6 casos, el 3.5 % de 6 a 10 casos y finalmente el 67.2 % no encuentra ningún caso.

Tabla 1. Porcentaje de casos de suplantación

Porcentaje (%) de casos que se detectó una suplantación de identidad por semestre			
1-3 Casos	3-6 Casos	6-10 Casos	Ningún Caso
22.5%	6.8%	3.5%	67.2 %

Tomada de encuesta por Google Forms a personas pertenecientes a la Universidad Continental en el periodo académico 2019-II

Nota: Porcentaje (%) de casos en que se detectó una suplantación de identidad por semestre

También en la tabla 2, se evidencia la frecuencia en porcentajes sobre la suplantación de identidad por semestre donde el 4.7 % es muy frecuente, el 14.1 % es frecuente, el 25.6% es poco frecuente y el 55.6 % nunca.

Tabla 2. Frecuencia de casos de suplantación

Frecuencia en porcentaje (%) de suplantación de identidad por semestre			
Muy Frecuente	Frecuente	Poco Frecuente	Nunca
4.7%	14.1%	25,6%	55,6%

Tomada de encuesta por Google Forms a personas pertenecientes a la Universidad Continental en el periodo académico 2019-II.

Nota: Frecuencia en porcentaje (%) de suplantación de identidad por semestre

De acuerdo con una encuesta realizada a docentes de la universidad, se constató que, durante el periodo académico 2019-II, el 91,9 % de docentes no logran conocer a sus estudiantes en su totalidad, mientras que el 8,1% si logra

conocerlos, esto sucede mayormente cuando los estudiantes se encuentran entre el primero a quinto semestre. También se identificó las acciones más sospechosas de una suplantación en los exámenes finales que es el 48,8 % por nerviosismo, el 22,1 % por desconocimiento de datos personales, el 20,9 % por mirada temerosa y el 4,7 % muestra inquietud. Por lo tanto, las acciones que realiza el docente para concretar la suplantación, el 57 % de preguntan los datos personales al estudiante, el 32,1 % realiza la verificación de identidad con su DNI física y el 10,9 % pide datos generales de su familia. Seguidamente el docente después de encontrar una suplantación dentro del aula el 84,9 % anulan el examen al estudiante, el 10,5 % realizan una sanción ejemplar según el reglamento interno de la universidad Continental y el 4,6 % de docentes deciden darle la oportunidad de rendir otro examen.

Lo que el equipo de trabajo propone para contrarrestar es: desarrollar un sistema de escritorio de reconocimiento facial para la identificación de estudiantes, así identificar suplantación de identidad. Por lo expuesto nuestros problemas de investigación son los siguientes:

1.2. Formulación del problema

1.2.1. Problema general

¿Cómo identificar la suplantación de identidad de los estudiantes en los exámenes finales de la modalidad presencial en la Universidad Continental de la ciudad de la Huancayo?

1.2.2. Problemas específicos

- ¿Cómo elaborar el modelo de reconocimiento facial para la identificación de los alumnos en los exámenes finales de la universidad Continental de la ciudad de Huancayo?
- ¿Cómo evaluar el modelo de reconocimiento facial para la identificación de los alumnos en los exámenes finales de la universidad Continental de la ciudad de Huancayo?

1.3. Objetivo general y objetivos específicos

1.3.1. Objetivo general

Desarrollar el sistema de escritorio para el reconocimiento facial mediante la metodología Kanban para identificar la suplantación de identidad de los alumnos en los exámenes finales de la universidad Continental de la ciudad de Huancayo.

1.3.2. Objetivos específicos

- Aplicar el proceso típico de clasificación en máquinas de aprendizaje (machine learning) en identificación de los alumnos en los exámenes finales de la Universidad Continental de la ciudad de Huancayo.
- Evaluar el modelo de reconocimiento facial mediante la aplicación de la matriz de confusión para la identificación de los alumnos en los exámenes finales de la universidad Continental de la ciudad de Huancayo.

1.4. Justificación e importancia

1.4.1. Justificación práctica

La realización de este proyecto tiene como finalidad apoyar en la educación digital. El tema propuesto en la tesis permitirá identificar si el estudiante pertenece a la clase, y esto se realizará mediante el reconocimiento del rostro al momento de ingresar al salón de clase, así disminuir la suplantación de alumnos en exámenes finales y al mismo tiempo se registrará de manera automática su asistencia al curso. Esto permite tener una mejor interacción con los usuarios docentes; evitando llamar lista al estudiante de una manera tradicional.

1.4.2. Justificación metodológica

Para el desarrollo del proyecto se optó por usar la metodología de gestión de proyectos Kanban, debido a que permite dar seguimiento y controlar el avance, lo cual ayuda a determinar las tareas que se debe hacer y actualizar la prioridad en función de los requerimientos, el desarrollo de trabajo está a la vista del equipo, y si hay algún retraso en el proyecto, es notorio.

Comparando Kanban con Scrum, Kanban se basa en el desarrollo, define el flujo de trabajo y el tiempo de entrega, aborda los números de tareas de forma fluida y simultánea y Scrum opta por dividir tareas complejas en varias historias de usuarios y lo controlan en un flujo de trabajo más visual. (3)

Por otro lado, para el reconocimiento de imágenes se utilizará la biblioteca OpenCV, ya que es libre y tiene una librería de visión artificial, contiene más de 2500 algoritmos lo cual ayuda en el análisis y tratamientos de las imágenes capturadas, también se usará la librería face-recognition que está construido con técnicas de aprendizaje profundo para el reconocimiento de los rostros.

1.5. Limitaciones y alcance del proyecto

1.5.1. Limitaciones

En el reconocimiento facial se consideran algunas limitaciones o inconvenientes como pueden ser:

- El proyecto no abarca la etapa de implementación.
- No se hará el reconocimiento facial a más de una persona al mismo tiempo.
- El reconocimiento con mascarilla no es preciso.
- El reconocimiento se realizará en tiempo real al estudiante.
- Se utilizará como base principal la recopilación de imágenes capturadas en un Dataset, para la comparación.
- Oclusión debido a objetos o accesorios: existen artefactos en el rostro que podrían ocasionar problemas en el procesamiento y extracción de características deseadas. Tales artefactos son: anteojos, aretes, piercing, entre otros.

Para la captura de las fotografías para el entrenamiento del modelo se deben cumplir los siguientes requisitos:

- Iluminación, es importante emplear una adecuada iluminación en la toma de las imágenes para el análisis de características deseadas.
- Para la captura de la imagen se debe realizar de manera frontal.
- Las imágenes para reconocer tendrán las características similares a las anteriores.

- Para el reconocimiento facial, el ambiente debe de estar bien iluminado (preferiblemente luz natural).

CAPÍTULO II

MARCO TEÓRICO

2.1. Antecedentes del problema

2.1.1. Antecedentes internacionales

- a) Tesis titulada: “*Desarrollo de un sistema de control de acceso de personal empleando reconocimiento facial respaldado con técnicas de aprendizaje profundo*”, la investigación tuvo como objetivo desarrollar e implementar el sistema control de acceso de personal por medio de reconocimiento facial en el cual se consideró algoritmos de aprendizaje profundo (*deep learning*) aplicado en tiempo real, a través de la visión por computadora. Para detectar y alinear el rostro se utilizó algoritmos de aprendizaje profundo como MTCNN, el modelo pre entrenado *Facenet* teniendo mejor respuesta en cuanto a la métricas rendimiento, accuracy y precisión. También se agregó un algoritmo de anti-plagio, denominado algoritmo de la vida, el cual cumple la función de detectar la imitación de rostros y evitar la vulnerabilidad al momento de reconocer a una persona mediante una foto o un video, la muestra utilizada para el desarrollo de pruebas fueron 11 personas, todas ellas con 2 fotos respectivamente. Los resultados para las pruebas realizadas demuestran que el sistema de reconocimiento facial arroja un elevado rendimiento en cuanto a identificación facial, cuando el usuario registrado se coloca con vistas al frente de la cámara, cercanos al 100% y además esto se logró con tan solo 2 fotos por cada usuario registrado. (4)

Esta tesis es fundamental para realizar una introducción del aprendizaje profundo en el reconocimiento facial donde se hace énfasis en el uso de la librería OpenCV utilizado juntamente con el lenguaje de programación Python.

- b) Tesis titulada: “*Reconocimiento facial basado en redes neuronales convolucionales*”, la investigación tuvo como objetivo implementar un sistema de reconocimiento facial que haga uso de redes neuronales artificiales y luego evaluar su funcionamiento, el algoritmo utilizado para la detección ha sido el de histogramas de gradientes orientes (HOG), para ajustar la imagen se utilizó un algoritmo llamado estimación de punto de referencia que ha sido implementado en la librería *Face Recognition*, así mismo para la extracción de características se utilizó una red neuronal convolucional profunda. Este modelo presenta una precisión del 99,38 %. Y por último para comparación y clasificación se realizó haciendo uso de un Perceptrón Multicapa (MLP) que, a diferencia de las redes neuronales artificiales usadas en los pasos anteriores, esta sí que va a ser configurada y entrenada personalmente, Para realizar experimentos con la red neuronal y evaluar sus resultados, se utilizó la técnica *validación cruzada o cross-validation*. En las pruebas se ha obtenido aproximadamente un total del 99,40% de aciertos, por lo que se podría afirmar que el sistema, en general, reconoce rostros con una considerable efectividad. (5)

Esta tesis es relevante para la investigación para dar una introducción de la librería *Face Recognition* que es necesario para el reconocimiento facial que utiliza inteligencia artificial.

- c) Artículo científico titulado: “*Reconocimiento facial en video usando Deep learning*”, la investigación tuvo como objetivo desarrollar una aplicación que consiste en la implementación de *deep learning* para el reconocimiento facial en vídeos, la metodología utilizada se basa en tres enfoques: en el primer enfoque se simula una base de datos, la cual tendrá almacenados los vectores de 128 posiciones de cada imagen del *dataset* de entrenamiento junto con una etiquetas(nombre de la persona a quien corresponde la foto). En el segundo enfoque se creó un Perceptrón Multicapa que hará el trabajo de la base de datos del primer enfoque. La motivación principal para la creación de un modelo de red neuronal es la estructura de los datos, muy familiares a los *datasets* de una red neuronal. Por último, Para el tercer enfoque se creó una red convolucional

personalizada. Esto se lo realizo utilizando la librería de *Tensor Flow* y *Keras*. El modelo consta de la primera convolución con una matriz de 94 x 94. Como resultados el modelo *FaceNet*, permitió obtener vectores de 128 posiciones que caracterizan a una imagen, con el que se obtuvo lo mejores resultados. (6)

d) Tesis titulada: “*Sistema de reconocimiento facial con redes neuronales para la toma de asistencia en aulas de clase*”, la investigación tuvo como objetivo desarrollar un prototipo que utilice *software* de reconocimiento facial, redes neuronales artificiales y *hardware* de bajo costo y consumo para la toma de asistencia en aulas de clase. Caso de estudio Universidad Autónoma de Bucaramanga, utilizo métodos holísticos basados en PCA/LDA, PCA es un método que transforma una determinada cantidad de variables correladas a un pequeño número de variables incorreladas llamadas componentes principales, LDA es un método que permite reducir un problema de identificación o ubicación de alta dimensión mediante la proyección de datos en un espacio vectorial más pequeño, Para que el modelo se considere útil, debe tener una precisión 44 de mínimo el 85% y una pérdida de información por debajo del 50% durante su entrenamiento. Como conclusión, es posible hacer un despliegue del sistema, pero sería un plan para implementar a largo plazo debido a la inversión y los cambios que se deben hacer en la infraestructura de la universidad (servidores, bases de datos, personal). (7)

e) Tesis titulada: “*Evaluación de algoritmos aplicados a la extracción de características para el reconocimiento facial en base a la ISO/IEC 25010*”, planteo su objetivo en evaluar los algoritmos más conocidos para la extracción de características faciales en sistemas de reconocimiento facial. Los algoritmos son: *EigenFace*, *FisherFace*, *Viola_Jones*. La evaluación se realizó utilizando los parámetros de la ISO/IEC25010, en esta tesis se implementó estos algoritmos utilizando algoritmo PCA (análisis de componentes principales), género como resultado una valoración de 56/70 para el algoritmo de *FisherFace*, seguido por *EigenFace* con un 54/70 y por último el algoritmo *Viola-Jones* con un 53/70 (8).

Esta tesis es muy importante para nuestra investigación ya que se toma como referencia la aplicación de los algoritmos y su porcentaje de efectividad de cada uno para el reconocimiento facial.

2.1.2. Antecedentes nacionales

a) Tesis titulada: *“Redes neuronales artificiales de aprendizaje profundo para el reconocimiento facial y control de acceso de estudiantes a un laboratorio”*, esta investigación tuvo como objetivo la implementación de una red neuronal convolucional de aprendizaje profundo para el reconocimiento facial y el control de acceso de estudiantes de la carrera de Ingeniería Mecatrónica; para esto, la metodología empleada consistió en el entrenamiento de la red neuronal, con el propósito de extraer los datos relevantes de los rasgos faciales en las fotografías tomadas al grupo de estudiantes. La muestra utilizada fue de 426 fotografías correspondiente a 14 alumnos que utilizaron el Laboratorio de Control en el semestre 2019-I; igualmente, el software empleado para el entrenamiento de la red fue el MATLAB y su *Toolbox Deep Learning*. También, se realizaron pruebas para la selección de la red, iniciando con una capa de convolución, luego dos, y finalmente tres capas, las cuales dieron como resultados los siguientes porcentajes de precisión 15.63 %, 94.00 % y 67.13 %, respectivamente. De esta manera, se optó por elegir la red neuronal con dos capas de convolución, de 16 y 32 filtros, para realizar el reconocimiento facial. (9)

b) Artículo científico titulado: *“Aplicación del deep learning para el reconocimiento facial con la presencia de oclusiones en el contexto de la pandemia COVID 2021”*, la investigación tuvo como objetivo aplicar el *Deep learning* para el reconocimiento facial inclusive con el uso mascarillas y/o lentes. La investigación usa *Python* y la librería de la red convolucional pre entrenada VGG16. Con una muestra total de 2400 imágenes para las pruebas, también se realiza el reconocimiento facial con la presencia de oclusiones en el rostro; utilizando una cámara web. La precisión alcanzada en el entrenamiento de la red neuronal es de 0.2 y el porcentaje de acierto obtenido de la aplicación en la segunda fase es de 71%, concluyendo aso

que el proyecto logra detectar a personas cuando estas utilizan mascarilla, lentes, ambos o sin el uso de estos; con el porcentaje de acierto. (10)

De esta tesis se toma referencia las técnicas de aprendizaje profundo para la clasificación de imágenes y reconocimiento facial.

c) Artículo científico titulado: “*Aplicación de inteligencia artificial para monitorear el uso de mascarillas de protección*”, tuvo como objetivo crear una aplicación web que permite monitorear el uso de la mascarilla protectora en ambientes públicos. Utilizando el *framework Flask*, en el lenguaje de *Python*, la aplicación cuenta con un panel de control que ayuda a visualizar los datos obtenidos. El proceso de detección utiliza el algoritmo Haar Cascade para clasificar rostros con y sin mascarillas protectoras. Como resultado, la aplicación web es liviana y permite detectar y almacenar en la nube las imágenes capturadas y la posibilidad de un mayor análisis de datos. El clasificador presenta precisión, revocación y f-score de 63%, 93% y 75%, respectivamente. Aunque la precisión fue satisfactoria, se realizarán nuevos experimentos para explorar nuevas técnicas de visión por computadora, como el uso del aprendizaje profundo. (11)

d) Artículo científico titulada: “*Sistema de reconocimiento facial para el Control de la trata de personas en Perú*”, la investigación tiene como objetivo desarrollar un sistema que contribuye al control de la trata de persona mediante el reconocimiento facial utilizando el algoritmo LBHP, con la metodología de desarrollo XP (programación extrema) ya que es la más apropiada para el trabajo en parejas y para realizar los roles de programación y testeo. La investigación es de tipo aplicada, con una muestra de 50 rostros de personas desaparecidas o denunciadas por trata de personas, obtenidos del sistema de recompensas del Ministerio del Interior del Perú y del Registro de latinoamericanos desaparecidos de la Organización de Estados Americanos únicamente, se tiene como resultado un reconocimiento facial de 68.75% al realizar las pruebas. (12)

e) Tesis titulada: “*Sistema Inteligente Basado en Redes Neuronales para mejorar la identificación de rostros de delincuentes en el distrito de Laredo*” Este sistema inteligente basado en RNA para mejorar la identificación de rostros de delincuentes en el distrito de Laredo, para tal efecto se desarrolló la metodología *Jhon Durkin*. Tuvo como objetivo mejorar la identificación de rostros delictivos en el distrito de Laredo en apoyo a la policía nacional; Mediante la instalación de un sistema inteligente basado en redes neuronales artificiales, realizó las pruebas para determinar la normalidad en datos estadísticos a través de Shapiro-Wilk, aplicando T de estudiantes muestras bajo 30 para los indicadores, la población es de 2553 delincuentes; tomando 334. Como muestra, es concluyó en el primer indicador que el tiempo promedio en la identificación de rostros de delincuentes se redujo en un 91,66% con una disminución de 414,85 segundos, en el segundo indicador el número de identificaciones de delincuentes se incrementó recientemente el número de delincuentes identificados en un 68.82 y en el tercer indicador el tiempo promedio en alerta sobre los delincuentes identificados ante la policía se reducen en un 77,31%. (13)

2.2. Bases teóricas

2.2.1. Suplantación de identidad

Es cuando una persona entra a un perfil que no es de él y se hace pasar por el verdadero titular o crearse una cuenta con el nombre de una persona falsa y esto genera errores en terceras personas. En la actualidad la suplantación de identidad es muy frecuente. (14)

2.2.2. Tipos de suplantación

- **Suplantación digital:** los ciberdelincuentes toman posesión de las plataformas digitales e ingresan a sitios privados, haciéndose pasar por otras personas.
- **Usurpación de identidad en redes sociales:** es una de las suplantaciones más comunes en los últimos años. Se trata de crear una cuenta falsa en las redes sociales o hackear alguna cuenta de otras personas.

2.2.3. Identidad

Identidad es una palabra de origen latino (Identitas) que puede utilizarse para referirse a un conjunto de características de un sujeto o de una comunidad. Distinguir a un individuo o grupo de los demás. La identidad también está relacionada con la percepción que una persona tiene de sí misma. (15)

2.2.4. Identificación

El concepto más sencillo es de reconocer o comprobar que la persona es la misma que se busca. La única manera de comprobar que sea la persona es con el DNI. (15)

2.2.5. Casos de suplantación de identidad

En la Universidad Nacional Federico Villarreal (UNFV) fueron detenidos por la policía nacional 11 falsos postulantes para un examen de admisión. El comandante general José Luis Lavalle explicó que la banda “Los cerebros de Cono Norte”, utilizaban masas de alginato (impresiones dentales) para sacar las huellas dactilares, y cobraban entre 2 mil a 5 mil dólares según la facultad. En los últimos años se ve en las noticias muchos casos como esto de suplantación de identidad. (16)

2.2.6. Inteligencia artificial

a) Definición

Existen distintas definiciones acerca de la inteligencia artificial, ya que es un tema amplio y complejo, se describe las definiciones más resaltantes a continuación:

Rouhiainen, define la IA como “La habilidad de los ordenadores para hacer actividades que normalmente requieren inteligencia humana”. Pero para brindar una definición más detallada dice que la IA “Es la capacidad de las máquinas para usar algoritmos, aprender de los datos y utilizar lo aprendido en la toma de decisiones tal y como lo haría un ser humano”. (17)

También Escolano, Cazorla, Alfonso, Colomina y Lozano definen de manera similar que: “LA inteligencia artificial es la ciencia de construir máquinas para que hagan cosas que, si las hicieran los humanos, requerirían inteligencia”. (18)

Por otro lado, Benítez, Escudero, Kanaan y Masip dicen: “La inteligencia artificial es una disciplina académica relacionada con la teoría de la computación cuyo objetivo es emular algunas de las facultades intelectuales humanas en sistemas artificiales” (19)

b) Objetivos de la inteligencia artificial

Boden, presenta dos objetivos principales de la IA: uno es tecnológico, consiste en usar los ordenadores para hacer cosas útiles. El otro es científico, usar conceptos y modelos de IA que ayuden a resolver cuestiones sobre humanos y demás seres vivos. (20)

2.2.7. Áreas de aplicación de la IA

Los ámbitos de la aplicación de la inteligencia artificial se dan en diversos campos como describen Benítez, Escudero, Kanann y Rodo. (19)

Campos de la robótica, el análisis de imágenes, tratamiento automático de textos, neurofisiológicas con el objetivo de asistir a personas discapacitadas. Otra de las ramas con mayor proyección son los sistemas expertos, en los que el objetivo es diseñar un sistema que permite analizar un conjunto de datos y realizar típicamente asociadas a la figura de un profesional experto como el diagnóstico, detección de fallas o la toma de decisiones.

El análisis de textos es otro ejemplo interesante en el que se desarrollan numerosos sistemas de IA. En la tabla 3, se describen los principales ámbitos de aplicación de los sistemas de inteligencia artificial.

Tabla 3. Principales ámbitos de aplicación de los sistemas de inteligencia artificial

Área	Aplicaciones
Medicina	Ayuda al diagnóstico Análisis de imágenes biomédicas Procesado de señales fisiológicas
Ingeniería	Organización de la producción Optimización de procesos Cálculo de estructuras Planificación y logística Diagnóstico de fallas Toma de decisiones
Economía	Análisis financiero y bursátil Análisis de riesgo Estimación de precios en productos derivados Minería de datos Marketing y fidelización de clientes
Biología	Análisis de estructura biológicas Genética médica y molecular
Informática	Procesado de lenguaje natural Criptografía Teoría de juegos Lingüística computacional
Robótica y automática	Sistemas adaptativos de rehabilitación Interfaces cerebro-computadora Sistemas de visión artificial Sistemas de navegación automática

Tomada de Benítez (19)

De igual manera (18) explican las diversas áreas donde las IA se presenta en mayor a menor medida, a continuación, se describen brevemente:

- **Tratamiento de lenguajes naturales:** este campo se puede englobar aplicaciones que realicen traducción entre idiomas, interfaces hombre-máquina que permiten interrogar una base de datos o dar órdenes a un sistema operativo, de esta manera que la comunicación sea más amigable al usuario.

- **Sistemas expertos:** se encuentran los sistemas donde la experiencia de personal calificado se incorpora a dichos sistemas para conseguir deducciones más cercanas a la realidad.
- **Robótica:** navegación de robots móviles, control de brazos de robots, ensamblaje de piezas, etc.
- **Problemas de percepción:** reconocimiento de objetos y del habla, detección de defectos de piezas por medio de visión, apoyo en diagnóstico médicos.
- **Aprendizaje:** modelización de conductas para su posterior implantación en computadoras.

Por otro lado, Rouhiainen menciona que “la IA se puede aplicar en casi todas las situaciones. Éstas son sólo algunas de las aplicaciones técnicas de la IA que están creciendo rápidamente en la actualidad: Reconocimiento de imágenes estáticas, clasificación y etiquetado: estas herramientas son útiles para una amplia gama de industrias” (17)

Además de las definiciones anteriores, Thrun, menciona en un video que: algunas veces, el término “inteligencia artificial” tiende a incomodar a la gente, así que se han propuesto algunas alternativas. Un reconocido experto en IA, (21) cree que es mejor llamarla “ciencia de datos”, una expresión menos intimidatoria que probablemente llevaría a una mayor aceptación entre el público. (17)

2.2.8. Aprendizaje automático (Machine Learning)

El aprendizaje automático (en inglés, machine learning) es uno de los enfoques principales de la inteligencia artificial. En resumen, se trata un aspecto de la informática en el que los ordenadores o las máquinas tienen la capacidad de aprender sin estar programados para ello. Un resultado típico serían las sugerencias o predicciones en una situación particular. (17)

Para Rouhiainen, (17), menciona que: “el aprendizaje automático utiliza algoritmos para aprender de los patrones de datos.”

La figura 2, muestra los tres subconjuntos del aprendizaje automático que pueden utilizarse: aprendizaje supervisado, no supervisado y de refuerzo.

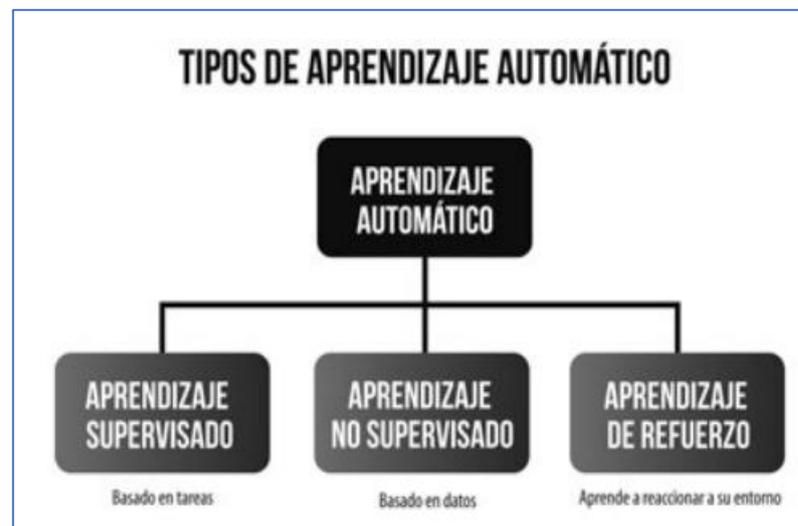


Figura 2. Tipos de aprendizaje automático
Tomada de ANALYTICS VIDHYA

Para explicar la diferencia entre estas tres clases de aprendizaje automático menciona (17) “Imaginemos que hay que organizar 20.000 imágenes y los algoritmos tienen que identificar las fotos en las que aparece un gato”. Y se explican a continuación:

- En el aprendizaje supervisado, los algoritmos usan datos que ya han sido etiquetados u organizados previamente para indicar cómo tendría que categorizarse la nueva información. Con este método, se requiere la intervención humana para proporcionar retroalimentación. Por ejemplo, enseñaríamos previamente al algoritmo fotos donde apareciera un gato para que luego pudiera identificar imágenes similares.
- En el aprendizaje no supervisado, los algoritmos no usan ningún dato etiquetado u organizado previamente para indicar cómo tendría que categorizarse la nueva información, sino que tienen que encontrar la manera de clasificarlas ellos mismos. Por tanto, este método no requiere la

intervención humana. (22) En el ejemplo, los algoritmos tendrían que clasificar ellos mismos todas las fotos en las que apareciera un gato en una categoría.

- Por último, con el aprendizaje por refuerzo, los algoritmos aprenden de la experiencia. En otras palabras, tenemos que darles “un refuerzo positivo” cada vez que aciertan (23). La forma en que estos algoritmos aprenden se puede comparar con la de los perros cuando les damos «recompensas» al aprender a sentarse, por ejemplo.

2.2.9. Aprendizaje profundo

El aprendizaje profundo (en inglés, *deep learning*) es una de aplicaciones más poderosas y de mayor crecimiento de la inteligencia artificial. Es un subcampo del aprendizaje automático que se utiliza para resolver problemas muy complejos y que normalmente implican grandes cantidades de datos.

El aprendizaje profundo se produce mediante el uso de redes neuronales artificiales, que se organizan en capas para reconocer relaciones y patrones complejos en los datos. Su aplicación requiere un enorme conjunto de información y una potente capacidad de procesamiento. Actualmente, se utiliza en el reconocimiento de voz, el procesamiento del lenguaje natural, la visión artificial y la identificación de vehículos en los sistemas de asistencia al conductor (22).

2.2.10. Reconocimiento facial

El reconocimiento facial es la autenticación biométrica que utiliza medidas del cuerpo que verifica la identidad de la persona, cuenta con subconjunto de datos biométricos que identifica al individuo a través de la mediación de la forma y estructura propias de un rostro. Permite establecer la identidad de los individuos mediante sus rasgos faciales, copiando las características de las personas, en la identificación de la imagen del rostro, tras analizar un conjunto de imágenes preestablecidas. (24)

La finalidad de un sistema de reconocimiento facial suele ser la siguiente: obtener una imagen o imagen de prueba de un rostro “desconocido”, encontrar una imagen de ese mismo rostro en un conjunto de imágenes “conocidas” o imagen de entrenamiento. La mayor dificultad añadida es que este proceso se pueda hacer en tiempo real, el sistema reconocerá automáticamente el rostro presente en la imagen o video. Por lo tanto, el reconocimiento facial es la versión más avanzada de la tecnología que un teléfono o una computadora para reconocer los amigos en las fotos.

2.2.11. Como se inició el reconocimiento facial

El reconocimiento facial en sus primeras investigaciones puede considerarse a *Woodrow Wilson Bledsoe* como un pionero de esta tecnología, ya que desde la década de 1960 ha estado investigando un sistema para clasificar los rasgos faciales humanos a través de la tabla RAND que tomaba coordenadas de diferentes puntos de una imagen para posteriormente realizar las respectivas mediciones. El sistema utiliza un lápiz óptico y coordina para colocar con precisión los ojos, la nariz o la boca de las personas. Por lo cual se desarrollaron técnicas que consistían en un sistema semiautomático donde solo el administrador tenía la autorización para localizar rasgos (ojos, orejas, nariz, boca) en fotografías.

En la década de 1970, comenzó la idea de realizar un reconocimiento facial automático. El trabajo de Kanade se basa en analizar imágenes a través de un programa encargado del procesamiento de imágenes para adquirir automáticamente rasgos faciales, entre ellos nariz, contorno de ojos, boca y mentón para medir la similitud entre rostros para después realizar el reconocimiento.

A fines de la década de 1987, gracias a Sirovich y Kirby aplicaron la técnica del álgebra lineal llamado análisis de componentes principales (PCA). Para demostrar que cualquier rostro puede ser representado por una combinación de rostros para formar una línea base conocida como “eigenpictures” con un error de 3 %.

En 1991 Kirby & Sirobich, utilizó la técnica PCA llamada EigenFace y se convirtió en el estándar para comparar el rendimiento de los algoritmos de reconocimiento facial más nuevos. Entre todos los algoritmos utilizados del reconocimiento facial 2D, tres son los mejor estudiados y utilizados: Análisis de Componentes Principales (PCA), Análisis Discriminante Lineal (LDA) y coincidencia de gráficos de paquetes flexibles (EBGM).

En 2001 nació la plataforma de detección de objetos el Framework Viola-Jones, que sus algoritmos es para detectar objetos en imágenes y posteriormente se utilizó para la detección de rostros de una manera eficaz. En la actualidad han surgido las Redes Neuronales Convolucionales y hasta el día de hoy son la mejor manera de detectar rostros, utilizando sistemas altamente eficientes capaces de almacenar información en la nube. (25)

2.2.12. Como funciona el reconocimiento facial

La funcionalidad principal del reconocimiento facial es que los rostros de las personas pasan por una cámara especializada y realiza la comparación con imágenes de personas en una lista. Estas listas de observación pueden contener fotos de todos y estas imágenes pueden provenir de cualquier lugar, incluidas nuestras cuentas de redes sociales. Los sistemas de tecnología facial pueden variar, pero generalmente funcionan de esta manera:

- ✓ **Reconocimiento facial:** la cámara detecta la imagen de tu rostro y la localiza, por recomendación la foto tiene que mostrar a la persona en primer plano.

- ✓ **Análisis facial:** se graba una imagen del rostro y se analiza. La mayoría de las tecnologías de reconocimiento facial se basan en imágenes 2D porque son fáciles de comparar con fotos genéricas o fotos en una base de datos. Los factores importantes son la distancia entre los ojos, la profundidad de las cuencas de los ojos, la distancia desde la frente hasta el mentón, la forma de los pómulos, los contornos de la boca, los oídos y el mentón; su objetivo es identificar puntos de referencia faciales que son importantes para distinguir entre los rostros.

- ✓ **Conversión de imagen a datos:** la captura de rostros convierte la información analógica (rostros) en un conjunto de información digital (datos) en función de los rasgos faciales de la persona. Los códigos digitales se denominan impresión de rostros.
- ✓ **Buscar coincidencia:** cualquier foto etiquetada en Facebook pasa a formar parte de la base de datos, que también se puede utilizar para el reconocimiento facial. Por lo tanto, se realiza para determinar si la huella facial coincide con la imagen en la base de datos de reconocimiento facial. Y se concluye que, de todas las medidas biométricas, el reconocimiento facial se considera la más natural. (26)

2.2.13. Redes neuronales convolucionales (CNN)

Las redes neuronales artificiales son sistemas que intentan simular el funcionamiento del cerebro humano. Las redes neuronales convolucionales son técnicas de aprendizaje profundo que se usan para el procesamiento de texto, su alto rendimiento o razón de existir es básicamente para el procesamiento de imágenes de una manera efectiva y eficiente como pueden ser fotografías y videos. Las operaciones que realiza son en 3 ejes (ancho, alto y profundo) en color de RGB (rojo, verde y azul). (27) Estas operaciones se pueden observar en la figura 3.

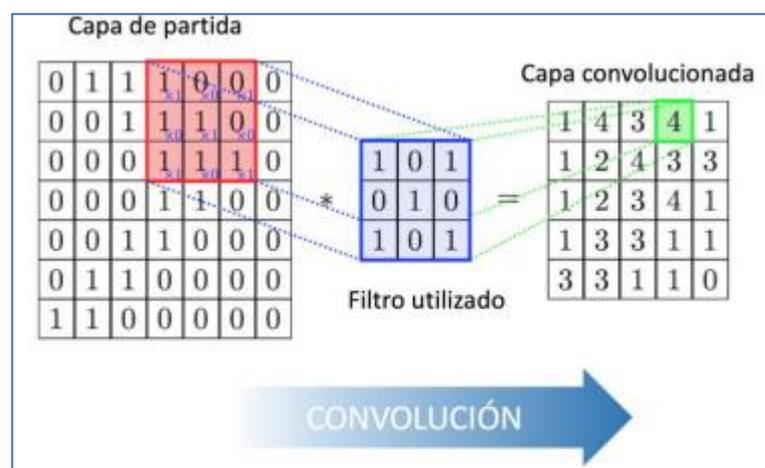


Figura 3. Red Neuronal Convolucional
Tomada de Solutions Data (29)

2.2.14. Estructura de las redes neuronales convolucionales

IBM describe las redes neuronales convolucionales cuentan con capas de entrada, capas ocultas y capas de salida. Los cuales se describen a continuación y se evidencia en la figura 4. (27)

Capa de entrada: se encarga de recibir todos los píxeles de nuestra imagen de entrada, luego pasa toda la información a la primera capa oculta.

Capa oculta: es donde se encuentra la información más importante, dentro de ella existen dos operaciones importantes: Primero es el pooling que es una operación que se encarga de reducir el tamaño de nuestra imagen, y para poder pasar a las siguientes capas, la imagen es más pequeña y menos pesado, segundo son las convoluciones, dentro de ellas pasan un montón de filtros a nuestra imagen para detectar ciertos patrones relevantes.

Capa de salida: que es la última capa oculta, su función es emitir un valor requerido y está limitado al tipo de problema que contiene el resultado.

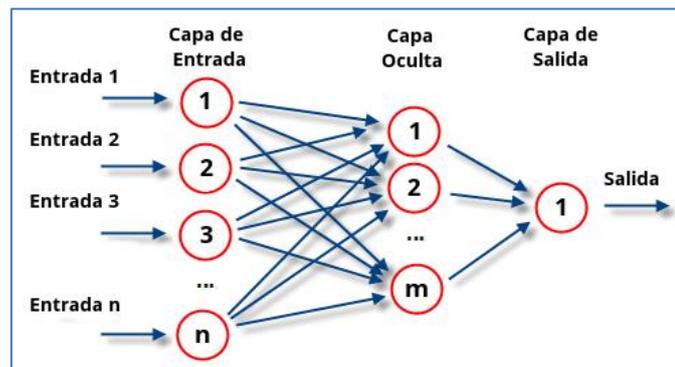


Figura 4. Redes Neuronales convolucionales, Tomada de Solutions Data (29)

2.2.15. Deep Learning o aprendizaje profundo

Aprendizaje profundo son técnicas del aprendizaje automático que se enfocan en simular el aprendizaje de los seres humanos para lograr tener conocimiento de algo. Emplea una red neuronal profunda que es una red de multicapas que contienen más de dos capas ocultas. Profundidad viene del número de capas que no intervienen los seres humanos sino el modelo mismo de aprendizaje realiza las acciones o actividades. En la figura 5, se muestra

las dos fases principales que compone el aprendizaje profundo que son fase de entrenamiento y fase de predicción.

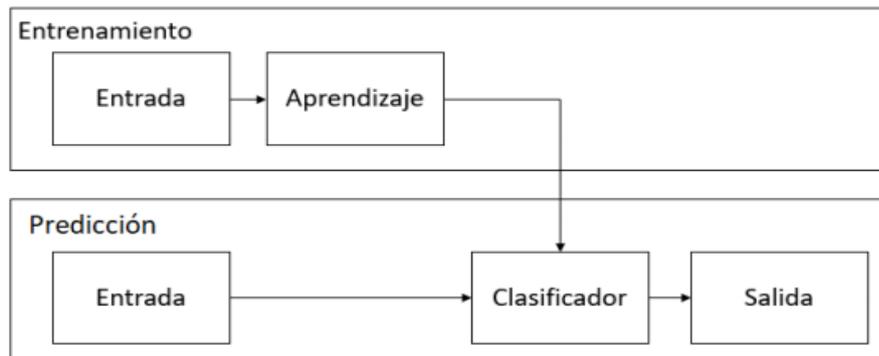


Figura 5. Aprendizaje profundo
Tomada y adaptada de Carchemilla (30)

2.2.16. Sistema de reconocimiento facial

Es un *software* que identifica automáticamente a la persona y está hecho por algoritmos que está dirigido por una computadora, lo cual realiza mediciones de los análisis de características fisiológicas del rostro como la forma y la estructura. (31)



Figura 6. Reconocimiento facial
Tomada de Narag (31)

En la figura 6, muestra que el reconocimiento facial es la tecnología que ayuda a verificar, autenticar y autorizar la identidad de una persona mediante el reconocimiento de su rostro.

2.2.17. Instalación de cámara para reconocimiento facial

Al momento de instalar la cámara es muy importante tener en cuenta el ángulo de inclinación que debe tener esta, para poder visualizar desde una mejor perspectiva del rostro el cual se quiere reconocer, y por otro lado evita los objetos obstructores que puede haber dentro del espacio o del pasillo.

2.3. Definición de términos básicos

- **Algoritmo:** en Matemáticas, Lógica, Ciencias de la Computación y disciplinas relacionadas es un conjunto de instrucciones o reglas definidas y no-ambiguas, ordenadas y finitas que permite, típicamente, solucionar un problema, procesar datos y llevar a cabo otras tareas o actividades. (32)
- **Aprendizaje máquina:** es un tipo de inteligencia artificial (AI) que brinda a las computadoras la capacidad de aprender sin tener que programarse directamente. El aprendizaje automático se centra en la creación de programas informáticos que pueden cambiar cuando se exponen a nuevos datos. (33)
- **DataSet:** conjunto de datos y es una colección de datos habitualmente tabulada.
- **Detección facial:** es una técnica que permite encontrar el rostro o cara de una o más personas en una imagen, ignorando el fondo de la imagen u otros objetos que estén presentes dentro de ella. (34)
- **Expresión facial:** una expresión facial son cambios que se producen en nuestro rostro en respuesta a estados emocionales internos. En este sentido, el análisis de las expresiones faciales incluye la identificación efectiva de las expresiones faciales y su relación directa con las emociones. (35)
- **Face-recognition:** es un algoritmo que ayuda a reconocer el rostro de una persona y está basado en características geométricas. (36)

- **Haar Cascade:** es un algoritmo en forma de cascada para la detección de rostros más antiguos, no solo se utiliza para identificar rostros, sino que también ojos, labios, etc. La detección de objetos mediante clasificadores en cascada basados en funciones Haar es un método eficaz de detección de objetos propuesto por Paul Viola y Michael Jones en su documento, *“Rapid Object Detection using a Boosted Cascade of Simple Features”* en 2001. Se trata de un enfoque basado en el aprendizaje automático en el que la función en cascada se entrena a partir de muchas imágenes positivas y negativas. Luego se utiliza para detectar objetos en otras imágenes. (37)
- **Identidad:** es el conjunto de los rasgos propios de un individuo o de una comunidad. Estos rasgos caracterizan al sujeto o a la colectividad frente a los demás. (38)
- **Imagen digital:** las imágenes digitales son fotos electrónicas tomadas de una escena o escaneadas de documentos, fotografías, manuscritos, textos impresos e ilustraciones. Se realiza una muestra de la imagen digital y se confecciona un mapa de ella en forma de cuadrícula de puntos o elementos de la figura (píxeles) (39)
- **Inteligencia artificial:** se refiere a sistemas o máquinas que imitan la inteligencia humana para realizar tareas y realizar mejoras iterativas basadas en la información recopilada. (40), también Definimos la IA como el estudio de los agentes que reciben percepciones del entorno y llevan a cabo las acciones. (41)
- **Oclusiones:** las oclusiones faciales, es decir, objetos que cubren parcialmente el rostro de una persona, hacen perder parte de la información y complicar el desempeño de las técnicas de reconocimiento de rostros. (42)
- **Procesamiento de imágenes digitales:** es una tecnología que está asociada a la ciencia de la computación y, por tanto, cabe mencionar como una proyección del término visión artificial, dentro del ámbito de la inteligencia artificial. (43)

- **Reconocimiento facial:** es una solución biométrica que utiliza algoritmos automatizados para poder verificar o identificar a una persona en función de sus características biológicas. La tecnología de reconocimiento facial funciona comparando las características faciales de la persona reconocida con características similares contenidas en una enorme base de datos.
- **Red neuronal artificial:** una red neuronal artificial está inspirado en el comportamiento del cerebro humano, que está compuesta por un conjunto de elementos que imitan los componentes neuronales biológicos, compuestas por neuronas artificiales, y tienen tres características principales: Aprendizaje, generalización y adaptación
- **Suplantación de identidad:** reemplazo o cambio por una persona o cosa que cumpla la misma función.
- **Visión por computadora:** consiste en la adquisición, procesamiento, clasificación y reconocimiento de imágenes digitales. (43) , Por otro lado, La visión artificial es una disciplina de la inteligencia artificial, así como el aprendizaje autónomo, aprendizaje profundo, procesamiento de lenguajes naturales, entre otros. (44)

CAPÍTULO III

METODOLOGÍA

3.1. Metodología aplicada para el desarrollo de la solución

Este proyecto es una Investigación Tecnológica (Desarrollo Tecnológico) que tiene un enfoque a la solución de un problema hallado, se identificó requerimientos y necesidades para el desarrollo para finalmente poner a prueba.

Para realizar este proyecto se tomó como base la metodología Kanban, en base a los WIP (trabajo en progreso) que es una de las partes fundamentales del método Kanban. Esto consistió en armar un tablero con la herramienta online Trello, en la cual se estableció que es lo que se tenía que hacer, que es lo que se está haciendo y que es lo que se hizo; esto permitió tener una vista general y detallada del flujo del proyecto.

2.3.1. Flujo de trabajo

Para el desarrollo de las actividades se trabajó con la herramienta Trello, el cual nos ayuda a tener un mejor control en el flujo del trabajo.



Figura 7. Fases del ciclo de desarrollo
Tomada y adaptada de Trello

En la figura 7, se evidencia que, en la herramienta Trello se organizó las 3 fases del proyecto las cuales son: por hacer, haciendo y hecho. Así tener una buena gestión del desarrollo del sistema.

- **Por hacer (To Do):** en esta fase se detallan las actividades que se desarrollarán y que serán clasificados cada uno por su prioridad y la necesidad de iniciar con el desarrollo.



*Figura 8. Actividades que se desarrollaran
Tomada y adaptada de Trello*

En la figura 8, se observa las actividades que se desarrollaran y con las fechas especificadas.

- **Haciendo (Doing):** en esta fase se muestra las actividades en proceso, el avance de cada uno depende del número de integrantes en el equipo y los conocimientos que se tiene sobre el tema.

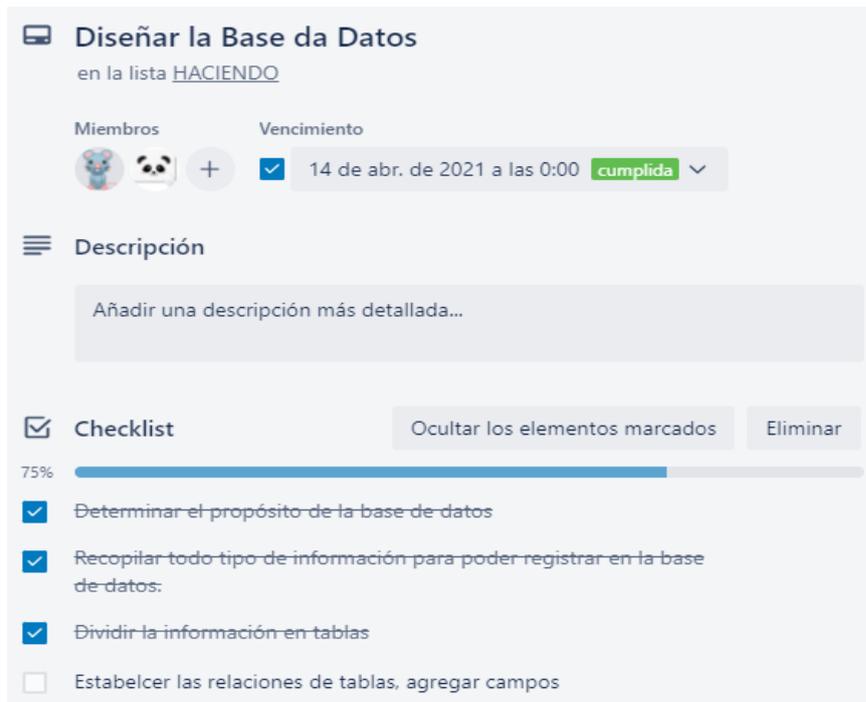


Figura 9. Actividades en proceso
Tomada y adaptada de Trello

En la figura 9, se observa que para desarrollar unas actividades de tiene que revisar que cada checklist ya sea haya terminado.

- **Hecho (Done):** Las actividades que pasaron por los 2 ciclos y que cumplirán con las funcionalidades previstas.



Figura 10. Actividades finalizadas
Tomada y adaptada de Trello

En la figura 10, se evidencia que las actividades ya se desarrollaron en su totalidad y en las fechas propuestas, y así se pueda proceder con la siguiente actividad.

Tener un control: la metodología Kanban es visual y esto nos permite tener un control del progreso del proyecto, en el desarrollo de algunas actividades se tuvo dificultades.

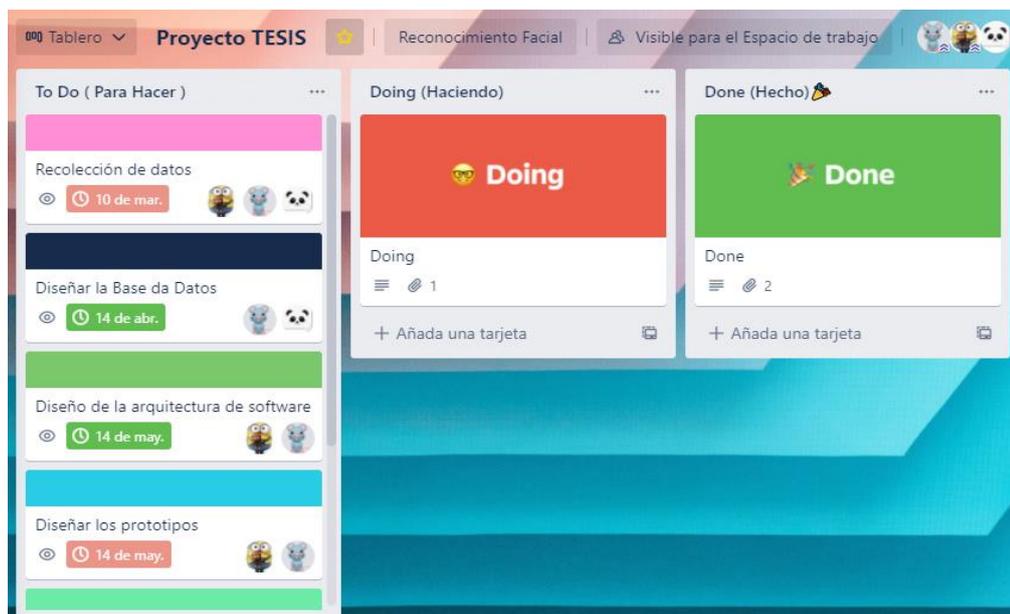


Figura 11. Fechas de entrega de cada una de las tarjetas del tablero Kanban Tomada y adaptada de Trello

En la figura 11, se evidencia los 3 procesos de Kanban organizada en la herramienta Trello, las actividades que se desarrollarán, lo que están en proceso y las finalizadas.

3.1.1. Realizar explícitas las políticas de proceso

Cada una de las tarjetas muestra etiquetas de color (verde, rojo, azul y morado); indicando lo siguiente, si una tarjeta tiene una etiqueta verde son las actividades requeridas, la etiqueta de color morado son las actividades que se terminaron de realizar, la etiqueta de color rojo significa que es urgente, la etiqueta de color azul significa que las actividades serán documentadas.

3.2. Desarrollo del sistema de escritorio mediante Kanban

Para el desarrollo del sistema se aplicó Kanban, el cual ayudó a minimizar los trabajos en progreso y por ello se siguió con las prácticas generales de Kanban:



Figura 12. Prácticas de Core Kanban
Tomada y adaptada de Trello

En la figura 12, se evidencia las prácticas Core Kanban que se tiene que seguir para un buen desarrollo del sistema.

3.2.1. Visualizar el flujo de trabajo

En este aspecto se creó un tablero de columnas en la herramienta Trello, y cada uno de los requerimientos establecidos pasará por cada fase y estará visible para el equipo. Cada que un requerimiento pasa a la siguiente fase, la nuevo se coloca en la primera columna, y lo que ya están en desarrollo se pueda evaluar y priorizar.

3.2.2. Limitar el trabajo en progreso (WIP)

Para avanzar con el desarrollo del sistema se tiene que terminar con un requerimiento y proseguir con el siguiente, para optimizar el trabajo se desarrollaron las limitaciones (1.5), esto no ayudó a avanzar con fluidez en el desarrollo del sistema, ya que se definió que no será capaz de hacer el sistema, para así no trabajar con módulos que no se necesitaran para el reconocimiento facial.

3.2.3. Gestionar el flujo

Para avanzar con fluidez el desarrollo del sistema, cada uno de los miembros de trabajo tiene que priorizar los requerimientos más importantes.

3.2.4. Hacer políticas explícitas

En el desarrollo del sistema es necesario definir las reglas del flujo, y saber en qué momento una actividad pasa al siguiente flujo y no acumularse. Para tener un mejor control se definió las políticas de DOD y DOR.

Tabla 4. Políticas DOD y DOR

Políticas DOD	Políticas DOR
En el desarrollo del reconocimiento facial el código no debe tener errores de compilación.	Los casos de prueba deben ser entendibles para cada miembro del equipo.
Cada código desarrollado debe estar documentada.	Los requerimientos deben tener una aceptación por el Product Owner
Los requerimientos para el desarrollo del sistema deben tener criterios de aceptación.	Cada requerimiento debe tener criterios de aceptación

Nota: Políticas explícitas DOD y DOR de desarrollo

En la tabla 4, se observa las políticas que se tienen que cumplir para pasar una actividad de un flujo a otro.



Figura 13. Políticas explícitas
Tomada de material de estudio de universidad Continental

En la figura 13, se observa las actividades que están en curso y las actividades que están listas para pasar al siguiente flujo cumpliendo las políticas establecidas.

3.2.5. Implementar circuitos de retroalimentación

Kanban ayudó en el desarrollo del sistema efectivamente, ya que nos dio la oportunidad de hacer las retroalimentaciones o hacer las revisiones del proceso de desarrollo y los riesgos que se generarían si no se llega a completar con un módulo a su totalidad.

En las reuniones del grupo de trabajo se hicieron priorizaciones, no empezar una actividad si lo que está en desarrollo aún no se ha terminado.

3.2.6. Mejorar colaborativamente

En el desarrollo del sistema cada integrante del equipo tenía que trabajar con una actividad hasta finalizarla.

3.2.7. Roles de Kanban

- ✓ **Product Owner:** uno de los miembros del equipo era responsable de entender la necesidad del cliente y maximizar el valor en cada proceso del desarrollo del sistema, ordenar las actividades que se deben empezar a trabajar.

- ✓ **Gestor de la entrega:** responsable del flujo de trabajo, ver si la actividad empezada está en proceso o terminado y planificar para la entrega a tiempo.

3.3. Metodología del reconocimiento facial

El reconocimiento facial es un método para identificar o confirmar la identidad de una persona a través de su rostro. (45)

A continuación, se describen las fases para el desarrollo del modelo de reconocimiento facial: La fase del entrenamiento del modelo que se encuentra compuesta por dos etapas: etapa 01 y 02, y la fase de la ejecución que contiene la etapa 03 y 04, estas fases van de acuerdo con los objetivos propuestos, se

siguen actividades cronológicamente, en la figura 14, se evidencia las fases que se realizarán.

	ETAPAS	DESCRIPCIÓN
FASE DE ENTRENAMIENTO	1	Elaboración del DataSet
	2	Entrenamiento de la red neuronal artificial
FASE DE EJECUCIÓN	3	Reconocimiento Facial
	4	Análisis de los resultados

Figura 14. Fases y Etapas de la Metodología, Tomada y adaptada de Kaspersky (45)

3.3.1. Fase del entrenamiento

- **Etapa 01:** Elaboración del DataSet

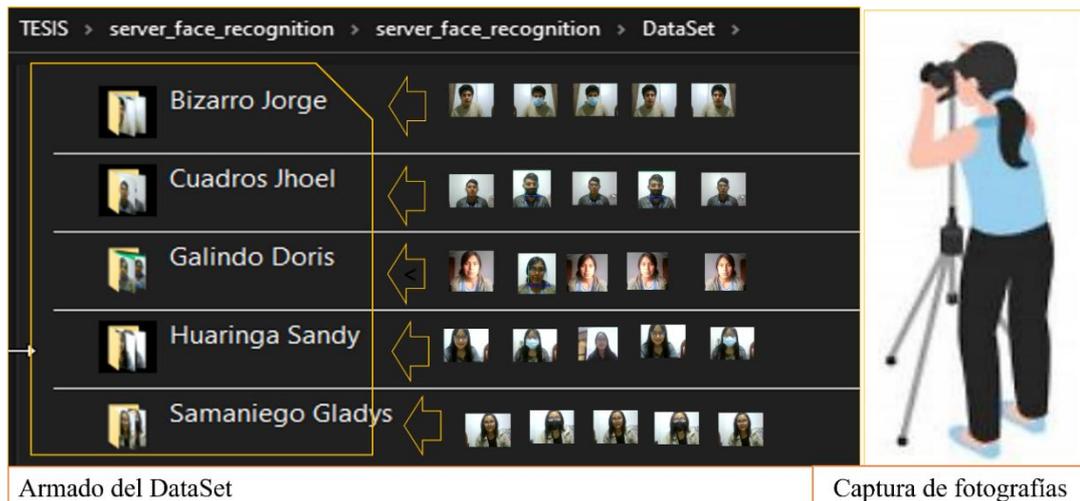


Figura 15. Creación del DataSet del modelo Tomada y adaptada de Mucha (46)

La figura 15, muestra la creación del *dataset*, con 50 imágenes por estudiante, para este caso siendo un total de 300 fotografías a colores como muestra, tomadas de forma consecutivas, dentro de cada carpeta personal de estudiante se cuenta con fotografías enfocando al rostro en diferentes ángulos sin movimiento corporal, en algunas fotografías se incluye la presencia de

oclusiones como lentes y mascarilla, sin perder los rasgos que caracteriza a cada persona, como los ojos, nariz y boca.

- **Etapa 02:** Entrenamiento de la red neuronal artificial

La figura 16, muestra la manera en cómo se realizará el entrenamiento de la red neuronal artificial y es importante ajustar a cada uno de los pesos de las entradas de todas las neuronas que forman parte de la red neuronal, así las respuestas de la capa de salida se ajusten lo más posible a los datos que tenemos.

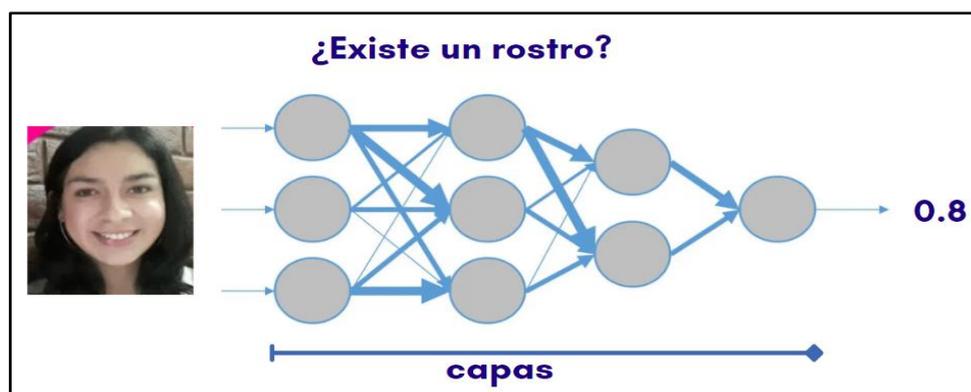


Figura 16. Modelo de entrenamiento de la red neuronal artificial, Tomada y adaptada de Solano (33)

Para que una red neuronal artificial sea capaz de identificar un rostro de una persona en cualquier imagen, es importante utilizar un elevado número de imágenes durante el entrenamiento, tanto con imágenes positivas que son rostros y con imágenes negativas que no sean rostros, con diferentes escenarios y variabilidad. Con esto la red neuronal será capaz de ajustar sus parámetros para extraer de manera precisa las características que identifican la presencia de un rostro en una imagen. (47)

Las imágenes para el entrenamiento de los algoritmos serán las que se almacenaron en la etapa 01, La figura 17, muestra la estructura de las imágenes como se encuentran contenidas.

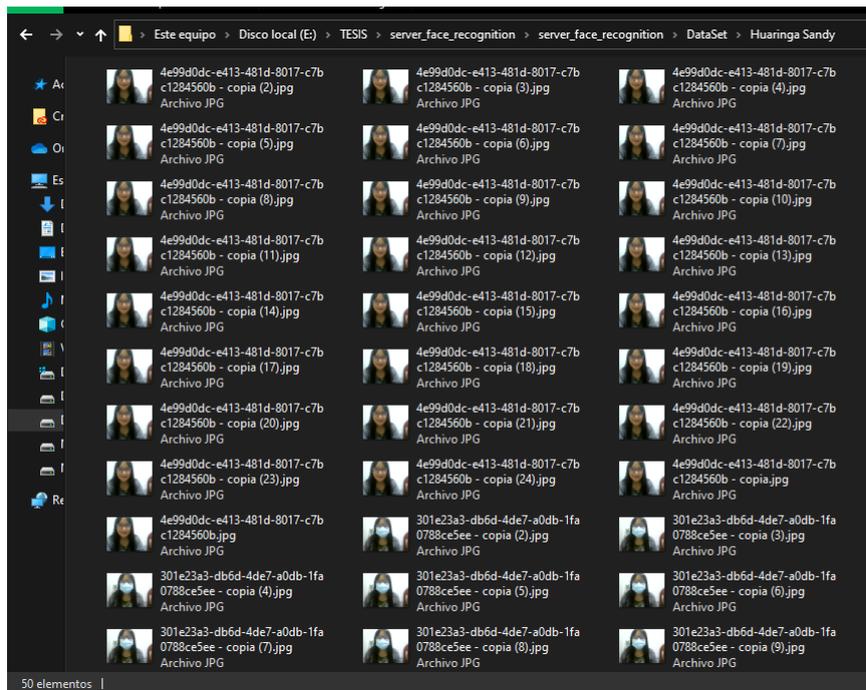


Figura 17. Formato de presentación de la fotografía para entrenamiento de los algoritmos

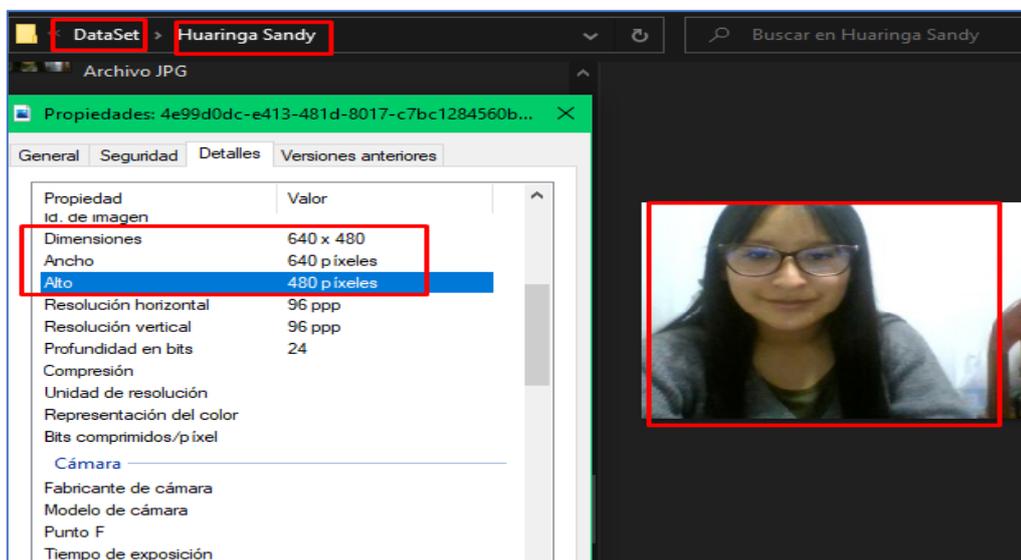


Figura 18. Especificaciones de la imagen a utilizar Tomada y adaptada de escrito de almacenamiento

Tal y como se muestra en la figura 18, las especificaciones de la imagen a utilizar son, el directorio donde se almacena tiene el nombre del estudiante a identificar, 640x480 son las dimensiones establecidas para el algoritmo y la foto a utilizar en formato JPG.

```
1 import cv2
2 import os
3 import numpy as np
4
5 dataPath = 'E:\TESIS\server_face_recognition\server_face_recognition\DataSet'
6 peopleList = os.listdir(dataPath)
7 print('Lista de personas: ', peopleList)
8
9 labels = []
10 facesData = []
11 label = 0
12
13 for nameDir in peopleList:
14     personPath = dataPath + '/' + nameDir
15     print('Leyendo las imágenes')
16
17     for fileName in os.listdir(personPath):
18         print('Rostros: ', nameDir + '/' + fileName)
19         labels.append(label)
20         facesData.append(cv2.imread(personPath+'/'+fileName,0))
21         image = cv2.imread(personPath+'/'+fileName,0)
22         cv2.imshow('image',image)
23         cv2.waitKey(10)
24         label = label + 1
25
26 print('labels= ',labels)
27 print('Número de etiquetas 0: ',np.count_nonzero(np.array(labels)==0))
28 print('Número de etiquetas 1: ',np.count_nonzero(np.array(labels)==1))
```

Figura 19. Entrenamiento de la red neuronal en Python -1

```
29
30 # Métodos para entrenar el reconocedor
31 face_recognizer = cv2.face.EigenFaceRecognizer_create()
32 face_recognizer = cv2.face.FisherFaceRecognizer_create()
33 face_recognizer = cv2.face.LBPHFaceRecognizer_create()
34
35 # Entrenando el reconocedor de rostros
36 print("Entrenando...")
37 face_recognizer.train(facesData, np.array(labels))
38
39 # Almacenando el modelo obtenido
40 face_recognizer.write('modeloEigenFace.xml')
41 face_recognizer.write('modeloFisherFace.xml')
42 face_recognizer.write('modeloLBPHFace.xml')
43 print("Modelo almacenado...")
```

Figura 20. Entrenamiento de la red neuronal en Python -2

La figura 19 y 20, muestra las líneas de código para el entrenamiento de la red, los cuales se explican a continuación:

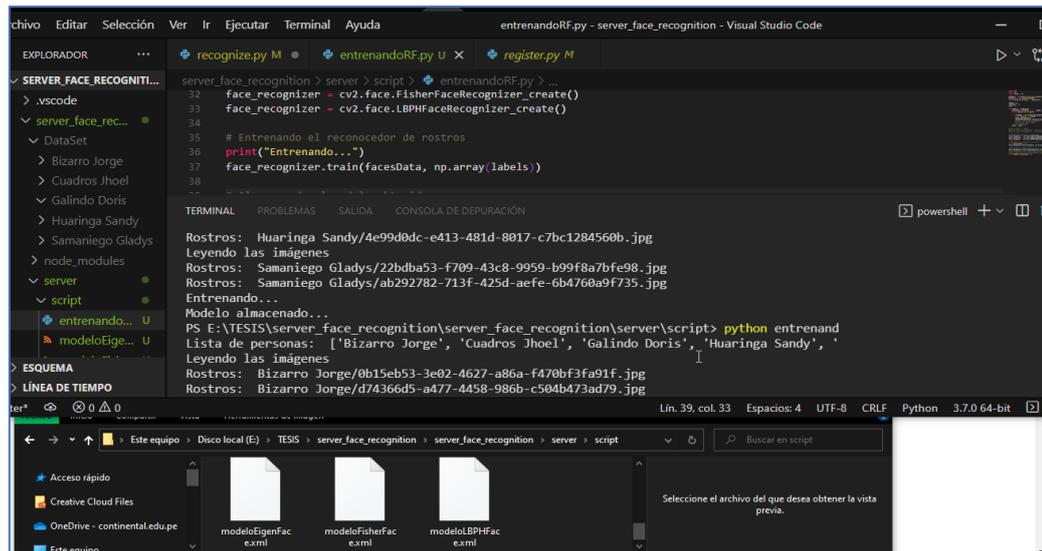


Figura 21. Entrenamiento del modelo en Python utilizando el editor de código visual studio code

Se realizó entrenamiento a 300 rostros, al finalizar el aprendizaje se obtuvo el modelo entrenado, la cual se almacena con el nombre de modelo "EigenFace.xml" como se evidencia en la figura 21, con los modelos almacenados.

3.3.2. La fase de ejecución

En la fase de ejecución se encuentra la etapa de reconocimiento y la etapa de análisis de las pruebas. Para la ejecución del reconocimiento es importante realizar el proceso de detección y clasificación de rostros usando Haar Cascade, el cual se describe a continuación:

- **Detección de rostros usando Haar Cascade**

La detección de un rostro consiste en encontrar las áreas dentro de una imagen o un video que contienen un rostro de una persona, para obtener la ubicación y tamaño exacto de la cara.

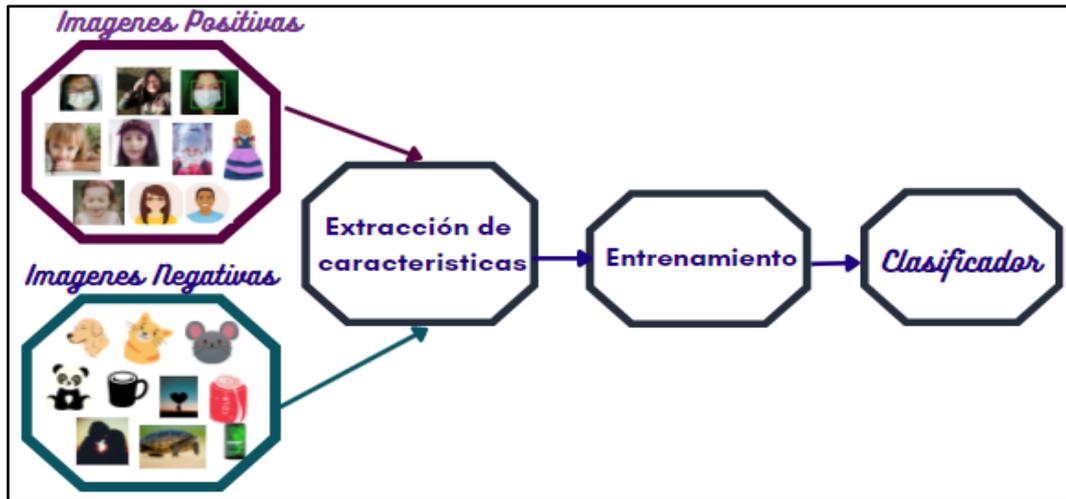


Figura 22. Proceso para crear un clasificador de rostros

En la figura 22, en un principio se necesita de una gran cantidad de imágenes para entrenar el clasificador, para que pueda diferenciar la presencia de un objeto y la no presencia del mismo. Para realizar un detector de rostros se requieren imágenes positivas (imágenes de rostro) e imágenes negativas (imágenes que no contengan rostro).

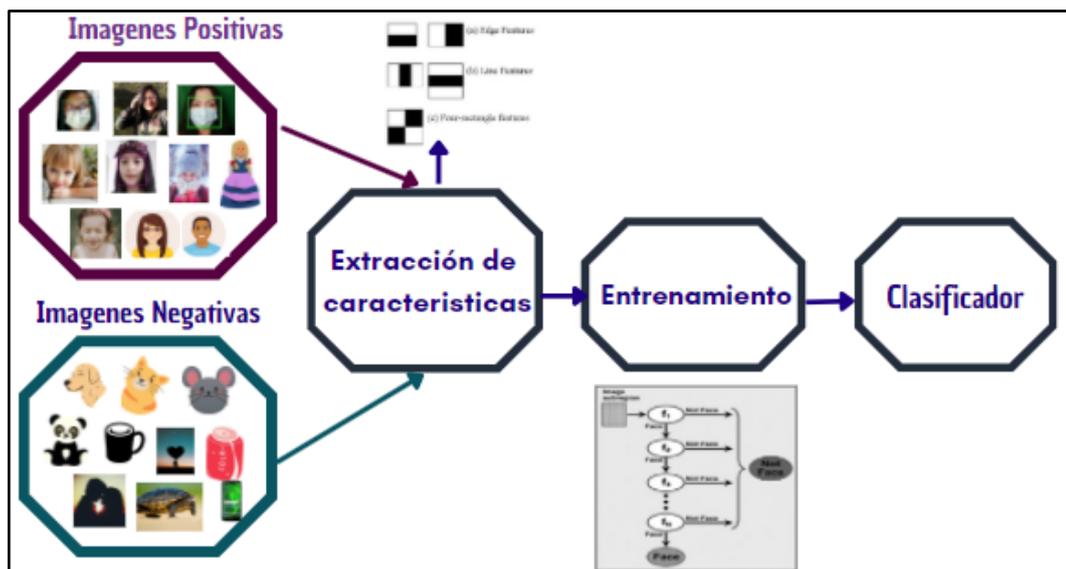


Figura 23. Proceso para crear un clasificador de rostros empleando Haar-Cascade Tomada y adaptada de Solano (34)

En la figura 23, tenemos imágenes positivas y negativas como datos de entrada. Para este caso la extracción de características es realizada con características de Haar, mientras que para entrenar el clasificador se usa una cascada de clasificadores que descartará las áreas que no sean considerados

rostros, y seguirá analizando los posibles candidatos a ser un rostro. Obteniendo finalmente el detector de rostros.

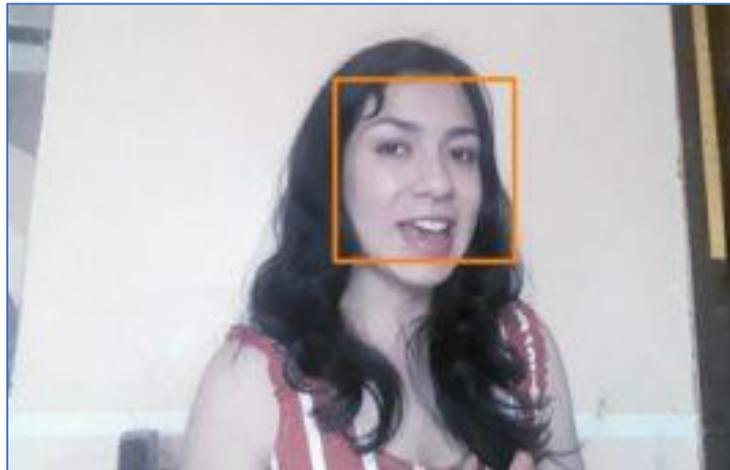


Figura 24. Ejemplo de detección de rostros
Tomada y adaptada de Solano (34)

En la figura 24, se evidencia como openCV logra detectar un rostro, mientras que ignora el fondo de la imagen u otros objetos que estén presentes dentro de ella.

- **Etapa 03: Reconocimiento facial**

En esta etapa es esencial la utilización de las librerías *OpenCV* y *haarcascade_frontalface*, juntamente con las librerías *Numpy* y *Python* los cuales hacen el trabajo de identificación y la verificación de los patrones y sus diversas características extraídas del rostro detectado para el reconocimiento en tiempo real al momento de la activación de la cámara.

Para el reconocimiento, previamente se realizó el entrenamiento lo cual se aloja en la carpeta *<script>* y se sigue el siguiente proceso que se muestra en la figura 25.

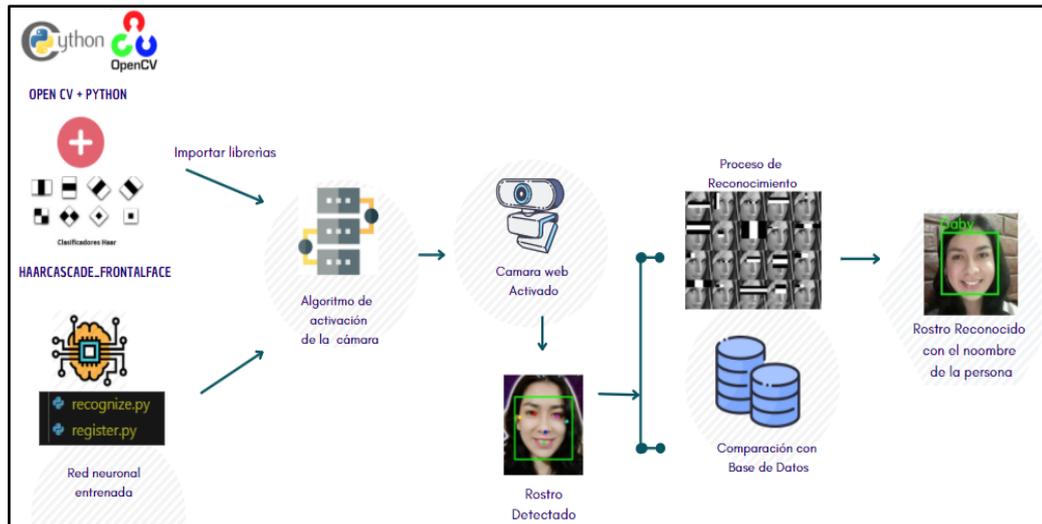


Figura 25. Reconocimiento facial

Estos rostros detectados se comparan con la base de datos, en la comparación de rostros, se analiza matemáticamente y sin margen de error la imagen entrante y se verifica que los datos correspondan con el estudiante que esté solicitando el ingreso al aula.

- **Etapa 04:** Análisis de los resultados

Se realiza diferentes pruebas en relación a los tipos planteados que son: rostro sin mascarilla y sin lentes, rostro con mascarilla sin lentes, rostro con lentes y mascarilla, rostro con lentes sin mascarilla. Estas pruebas se clasifican por tipo para analizar el porcentaje de reconocimiento de cada uno de ellos, de manera que a través de la aplicación del *Deep Learning* cumpla un porcentaje de precisión aceptable al reconocer e identificar el rostro de las personas, incluyendo las restricciones que sean causados por mascarillas o lentes que reduce los rasgos faciales para un mejor reconocimiento.

3.4. La matriz de confusión

La matriz de confusión nos ayuda a comprobar si el modelo de clasificación basado en aprendizaje automático es bueno. Sirve para mostrar de forma explícita cuando una clase es confundida con otra.

La matriz realiza una comparación de los valores reales con los predichos por el modelo de aprendizaje, esto también nos ayuda a comprender que tan bien está funcionando el modelo.

En la figura 26, se muestra la matriz de confusión de manera general.

		ACTUAL VALUES	
		POSITIVE	NEGATIVE
PREDICTED VALUES	POSITIVE	TP	FP
	NEGATIVE	FN	TN

Figura 26. Matriz de confusión Tomada de TechTarget (33)

La matriz tal y como se muestra está compuesta por:

- La variable de destino tiene dos valores: positivo o negativo.
- Las columnas los valores reales de la variable objetivo
- Las filas representan los valores predichos de la variable objetivo.

Para entender la matriz de confusión, se tiene que entender el verdadero positivo, el verdadero negativo, el falso positivo y el falso negativo. Las cuales se muestran en la figura 27.

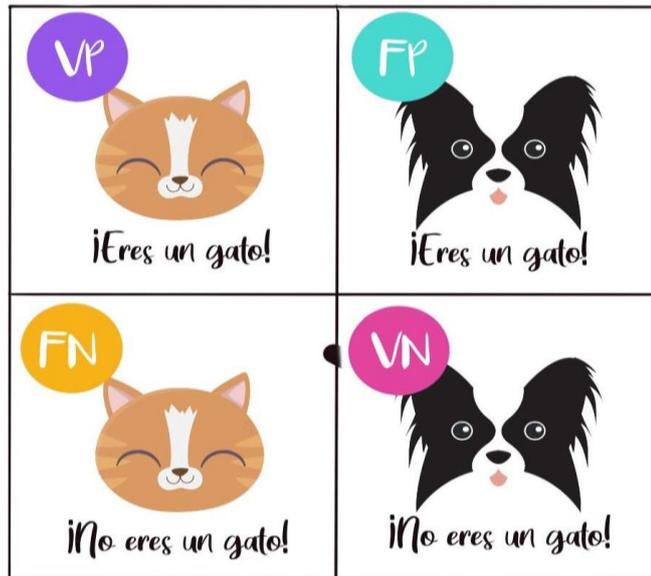


Figura 27. Entendiendo el verdadero positivo, el verdadero negativo, el falso positivo y el falso negativo en una matriz de confusión
 Tomada de TechTarget (33)

3.4.1. Verdadero positivo (TP)

- El valor predicho coincide con el valor real
- El valor real fue positivo y el modelo predijo un valor positivo

3.4.2. Verdadero negativo (TN)

- El valor predicho coincide con el valor real
- El valor real fue negativo y el modelo predijo un valor negativo

3.4.3. Falso positivo (FP): error de tipo 1

- El valor predicho se predijo falsamente
- El valor real fue negativo pero el modelo predijo un valor positivo
- También conocido como error tipo 1

3.4.4. Falso negativo (FN): error de tipo 2

- El valor predicho se predijo falsamente
- El valor real fue positivo, pero el modelo predijo un valor negativo
- También conocido como error tipo 2

De forma resumida se explica e interpreta en la figura 28, de la siguiente manera:

VP	Verdadero positivo
Interpretación	Predijo positivo y es verdad
VN	Verdadero negativo
Interpretación	Predijo negativo y es verdad
FP	Falso positivo
Interpretación	Predijo positivo pero es falso
FN	Falso negativo
Interpretación	Predijo negativo pero es positivo

*Figura 28. Interpretación de la matriz de confusión
Tomada de TechTarget (33)*

3.4.5. Métricas de la matriz de confusión

Para la evaluación el sistema puede devolver una identidad clasificando la imagen a reconocer, o por otro lado devolver un mensaje indicando que la imagen no puede ser reconocida, por lo que, en este caso, se consideran las siguientes medidas:

- Verdaderos positivos
- Verdaderos Negativos
- Falsos positivos
- Falsos Negativos

3.5. Alcance del proyecto

- La investigación se aplicará en la universidad Continental
- Se basará en los problemas encontrados en la universidad Continental
- Control eficiente en el ingreso de estudiantes a exámenes
- Manejo de asistencia.

CAPÍTULO IV

ANÁLISIS Y DISEÑO DE LA SOLUCIÓN

4.1. Identificación de los requerimientos

En esta fase de la investigación, el usuario cumple un rol importante, ya que dará a conocer las características necesarias para el desarrollo del sistema, por consiguiente, se busca satisfacer las necesidades del usuario al momento del desarrollo.

La identificación de los requerimientos es tarea del usuario, debido a que se prioriza en orden del valor que aporta el requerimiento a la solución final teniendo en cuenta el contexto de las limitaciones del proyecto.

Al tratarse de un proyecto en un plazo determinado, algunos requerimientos se catalogarán con una prioridad baja y que tal vez no sea posible desarrollarlos dentro de esta versión funcional del sistema, en el cual solo se va a abarcar los requerimientos que tengan prioridad alta y que abarcan las principales funcionalidades del sistema. Para identificar los requerimientos funcionales se realizó una encuesta en el año 2019-II a 86 personas (ver anexo A).

4.1.1. Usuarios que intervienen en el sistema

Los usuarios son parte esencial, ya que estos son los que hacen uso del sistema de reconocimiento facial para un mejor control de identificación de los

estudiantes, es por ello por lo que se define los principales usuarios que intervienen en el sistema, a continuación, se detalla en la tabla 5:

Tabla 5. Usuarios del sistema

Tipo de usuario	Descripción
Administrador	Control completo del sistema para agregar, modificar y visualizar estudiantes, docentes y registros académicos.
Docente	Registro de estudiantes, control de identificación de identidad para evitar la suplantación.
Estudiantes	Hacen uso del sistema de reconocimiento facial para la identificación de identidad.

Nota: Usuarios que intervienen en el uso del sistema

4.1.2. Historias de usuarios

Las historias de usuario es una explicación general e informal de una función de software escrito desde la perspectiva del usuario final. Para este proyecto se realizó en la tabla 6 la lista de historias de usuario.

Tabla 6. Historias de usuarios

ID	Historias de usuario
HU01	Como usuario administrador requiero una interfaz de inicio de sesión con el fin de ingresar con los accesos permitidos al sistema.
HU02	Como usuario administrador requiero registrar tipos usuarios con diferentes permisos con el objetivo de tener el control según el rol definido.
HU03	Como usuario final requiero registrar un nuevo estudiante con el objetivo de verificar su identidad mediante el reconocimiento facial al momento de ingresar al salón de clases.
HU04	Como usuario final requiero que el sistema realice captura de rostro para almacenar en la base de datos y realizar las comparaciones e identificación del estudiante.
HU05	Como usuario administrador requiero guardar la información de los estudiantes en la base datos con la finalidad de realizar análisis y comparaciones
HU06	Como usuario final requiero que se registre la asistencia con la identificación del rostro para obtener reportes de consolidado.
HU07	Como usuario administrador requiero almacenar las imágenes capturadas en un Dataset con la finalidad de utilizar para el entrenamiento del modelo de reconocimiento facial.

HU08	Como usuario final requiero que el sistema compare la foto de entrada con la imagen almacenada en la base de datos. Con el fin de encontrar un porcentaje de similitud.
HU09	Como usuario final requiero que el sistema realice el reconocimiento facial en tiempo real, con la finalidad de disminuir el número de suplantaciones de identidad.
HU10	Como estudiante requiero que el sistema valide mi identificación con el fin de poder ingresar al salón de clases.
HU11	Como usuario final requiero que el sistema me brinde un porcentaje de similitud del estudiante que realice la identificación de identidad.
HU12	Como administrador requiero listar los usuarios que usan el sistema.
HU13	Como docente requiero listar los estudiantes ingresados a la clase.
HU14	Como usuario final requiero un manual de uso con el objetivo de conocer las funcionalidades del sistema.

Nota: Historias de usuarios

4.1.3. Requerimientos funcionales

Los requerimientos funcionales describen el comportamiento y los datos que gestionara el sistema. En la tabla 7 se listan los requerimientos funcionales para el sistema.

Tabla 7. Requerimientos funcionales

N°	Requerimientos funcionales	Usuario
RF1	Iniciar sesión	Administrador
RF2	Registrar usuario	Administrador
RF3	Registrar un estudiante	Docente
RF4	Capturar la imagen a través de una cámara web	Estudiante
RF5	Guardar la información en la base de datos	Administrador
RF6	Registrar asistencia	Estudiante
RF7	Guardar la imagen capturada de los estudiantes en una carpeta	Administrador
RF8	Realizar el reconocimiento facial	Administrador
RF9	El reconocimiento facial permitirá reconocer en tiempo real al estudiante.	Estudiante
RF10	Verificar si el estudiante es correcto.	Administrador
RF11	Permitir el ingreso del estudiante según el porcentaje de similitud.	Docente
RF12	Listar Usuarios	Administrador
RF13	Listar Estudiantes	Docente
RF14	Generar un manual de usuario	Administrador

Asimismo, en la tabla 8, tabla 9, tabla 10, tabla 11, tabla 12, tabla 13, tabla 14, tabla 15, tabla 16, tabla 17, tabla 18, tabla 19, tabla 20 y tabla 21, se describen a detalle cada requerimiento funcional, especificando el nombre, descripción, indicador, prioridad, tipo de usuario, datos de entrada y salida.

Tabla 8. Requerimientos funcionales – Iniciar sesión

Nombre de requerimiento	Iniciar Sesión		
Descripción	El ingreso de inicio de sesión debe permitir la restricción o aprobación de acceso al sistema mediante el usuario y contraseña. Con la finalidad de garantizar la integridad de la información.		
Indicador	RF01	Prioridad	Alta
Tipo de requerimiento	Funcional	Usuario	Administrador
Datos de entrada	<ul style="list-style-type: none"> Nombre de usuario. Contraseña de usuario. 	Datos de Salida	<ul style="list-style-type: none"> Aprobación de acceso al sistema Denegación de acceso al sistema

Tabla 9. Requerimientos funcionales – Gestión de usuarios

Nombre de requerimiento	Gestión de usuarios		
Descripción	En el sistema es necesario tener los tipos de usuarios con diferentes permisos, el objetivo es llevar un control adecuado en la visualización y administración según el rol definido para la manipulación al sistema.		
Indicador	RF02	Prioridad	Alta
Tipo de requerimiento	Funcional	Usuario	Administrador
Datos de entrada	<ul style="list-style-type: none"> Módulo de administración de usuarios 	Datos de Salida	<ul style="list-style-type: none"> Creación de usuarios. Listar usuarios

Tabla 10. Requerimientos funcionales – Registrar estudiante

Nombre de requerimiento	Registrar estudiante		
Descripción	El sistema debe registrar los estudiantes con el objetivo de verificar su identidad mediante el reconocimiento facial al momento de ingresar al salón de clases y evitar cualquier tipo de suplantación.		
Indicador	RF03	Prioridad	Alta
Tipo de requerimiento	Funcional	Usuario	Docente
Datos de entrada	<ul style="list-style-type: none"> • DNI del estudiante. • Nombre del estudiante. • Apellido del estudiante 	Datos de Salida	<ul style="list-style-type: none"> • Ingreso al salón de clases. • No ingresa al salón de clases.

Tabla 11. Requerimientos funcionales – Capturar rostro

Nombre de requerimiento	Capturar la imagen a través de una cámara web		
Descripción	El sistema debe capturar el rostro del estudiante para almacenar en la base de datos y realizar las comparaciones e identificación.		
Indicador	RF04	Prioridad	Alta
Tipo de requerimiento	Funcional	Usuario	Estudiante
Datos de entrada	<ul style="list-style-type: none"> • Captura de rostro 	Datos de Salida	<ul style="list-style-type: none"> • Almacena en la Base de Datos.

Tabla 12. Requerimientos funcionales – Guardar información

Nombre de requerimiento	Guardar la información en la base de datos		
Descripción	El sistema debe guardar la información de los estudiantes en la base datos con la finalidad de realizar análisis y comparaciones con cada información del estudiante.		
Indicador	RF05	Prioridad	Alta
Tipo de requerimiento	Funcional	Usuario	Administrador
Datos de entrada	<ul style="list-style-type: none"> • Nombre del estudiante • Apellido del estudiante • Captura del rostro 	Datos de Salida	<ul style="list-style-type: none"> • Almacena en la Base de Datos.

Tabla 13. Requerimientos funcionales – Registrar asistencia

Nombre de requerimiento	Registrar asistencia		
Descripción	El sistema registra la asistencia con la identificación del rostro para obtener reportes de consolidado.		
Indicador	RF06	Prioridad	Alta
Tipo de requerimiento	Funcional	Usuario	Estudiante
Datos de entrada	<ul style="list-style-type: none"> Nombre del estudiante Apellido del estudiante Captura del rostro del estudiante. 	Datos de Salida	<ul style="list-style-type: none"> Registro de asistencia. No registra la asistencia

Tabla 14. Requerimientos funcionales – Guardar imagen

Nombre de requerimiento	Guardar la imagen capturada de los alumnos en una carpeta		
Descripción	El sistema almacena las imágenes capturadas en un Dataset con la finalidad de utilizar para el entrenamiento del modelo de reconocimiento facial.		
Indicador	RF07	Prioridad	Alta
Tipo de requerimiento	Funcional	Usuario	Administrador
Datos de entrada	<ul style="list-style-type: none"> Imagen capturada del estudiante 	Datos de Salida	<ul style="list-style-type: none"> Almacena las imágenes capturadas en un Dataset

Tabla 15: Requerimientos funcionales – Reconocimiento facial

Nombre de Requerimiento	Realizar el reconocimiento facial		
Descripción	El sistema compara la foto de entrada con la imagen almacenada en la base de datos. Con el fin de encontrar un porcentaje de similitud.		
Indicador	RF08	Prioridad	Alta
Tipo de Requerimiento	Funcional	Usuario	Administrador
Datos de entrada	<ul style="list-style-type: none"> Imágenes almacenadas del estudiante 	Datos de Salida	<ul style="list-style-type: none"> Porcentaje de similitud con las imágenes almacenadas

Tabla 16. Requerimientos funcionales – Reconocimiento facial en tiempo real

Nombre de requerimiento	El reconocimiento facial permitirá reconocer en tiempo real al estudiante		
Descripción	El sistema realiza el reconocimiento facial en tiempo real, con la finalidad de disminuir el número de suplantaciones de identidad.		
Indicador	RF09	Prioridad	Alta
Tipo de requerimiento	Funcional	Usuario	Estudiante
Datos de entrada	<ul style="list-style-type: none"> • Imágenes almacenadas del estudiante 	Datos de Salida	<ul style="list-style-type: none"> • Estudiantes suplantados

Tabla 17: Requerimientos funcionales – Verificar identidad

Nombre de requerimiento	Verificar si el estudiante es correcto		
Descripción	El sistema valida la identificación del estudiante con el fin de poder ingresar al salón de clases.		
Indicador	RF10	Prioridad	Alta
Tipo de requerimiento	Funcional	Usuario	Administrador
Datos de entrada	<ul style="list-style-type: none"> • Imágenes almacenadas del estudiante 	Datos de Salida	<ul style="list-style-type: none"> • Identificación de estudiantes.

Tabla 18. Requerimientos funcionales – Permitir ingreso

Nombre de requerimiento	Permitir el ingreso del estudiante según el porcentaje de similitud		
Descripción	El sistema brinda un porcentaje de similitud del estudiante que realice la identificación de identidad		
Indicador	RF11	Prioridad	Alta
Tipo de requerimiento	Funcional	Usuario	Docente
Datos de entrada	<ul style="list-style-type: none"> • Imágenes almacenadas del estudiante 	Datos de Salida	<ul style="list-style-type: none"> • Porcentaje de similitud del estudiante.

Tabla 19. Requerimientos funcionales – Listar usuarios

Nombre de requerimiento	Listar usuarios		
Descripción	El sistema muestra la visualización de los usuarios existentes con el objetivo de garantizar que no existan usuarios no permitidos al acceso del sistema.		
Indicador	RF12	Prioridad	Media
Tipo de requerimiento	Funcional	Usuario	Administrador
Datos de entrada	<ul style="list-style-type: none"> Lista de usuarios registrados. 	Datos de Salida	<ul style="list-style-type: none"> Rol de usuario Nombre de usuario Apellido de usuario

Tabla 20. Requerimientos Funcionales – Listar estudiantes

Nombre de requerimiento	Listar Estudiantes		
Descripción	El sistema muestra la visualización de los estudiantes existentes en el sistema para garantizar que no existan estudiantes duplicados.		
Indicador	RF13	Prioridad	Media
Tipo de requerimiento	Funcional	Usuario	Docente
Datos de entrada	<ul style="list-style-type: none"> Lista de estudiantes registrados 	Datos de Salida	<ul style="list-style-type: none"> DNI del estudiante Nombre del estudiante. Apellido del estudiante.

Tabla 21. Requerimientos funcionales – Manual de usuario

Nombre de requerimiento	Generar un manual de usuario		
Descripción	El usuario final requiere un manual de uso con el objetivo de conocer las funcionalidades del sistema.		
Indicador	RF14	Prioridad	Media
Tipo de requerimiento	Funcional	Usuario	Administrador
Datos de entrada	<ul style="list-style-type: none"> Funcionalidad del sistema 	Datos de Salida	<ul style="list-style-type: none"> Manual de uso

4.1.4. Especificaciones de los requerimientos

La especificación de requerimientos sirve como medio de comunicación entre cliente, usuario y desarrolladores. En las especificaciones se recogen tanto necesidades de clientes y usuarios, como los requisitos que debe cumplir el sistema a desarrollar para satisfacer dichas necesidades. En la tabla 22, se describen estas especificaciones de los requerimientos funcionales.

Tabla 22. Especificaciones de los requerimientos funcionales

Id Req.	Requerimiento	Descripción	Prioridad	Estimación (horas)
RF_001	Iniciar sesión	Como usuario administrador requiero una interfaz de inicio de sesión con el fin de ingresar con los accesos permitidos al sistema.	Alta	16 h
RF_002	Registrar Usuario	Como usuario administrador requiero registrar tipos usuarios con diferentes permisos con el objetivo de tener el control según el rol definido.	Alta	16h
RF_003	Registrar estudiante	Como usuario final requiero registrar un nuevo estudiante con el objetivo de verificar su identidad mediante el reconocimiento facial al momento de ingresar al salón de clases.	Alta	64h
RF_004	Capturar rostro del estudiante	Como usuario final requiero que el sistema realice captura de rostro para almacenar en la base de datos y realizar las comparaciones e identificación del estudiante.	Alta	48h
RF_005	Guardar la información en la base de datos	Como usuario administrador requiero guardar la información de los estudiantes en la base datos con la	Alta	32h

		finalidad de realizar análisis y comparaciones		
RF_006	Registrar Asistencia	Como usuario final requiero que se registre la asistencia con la identificación del rostro para obtener reportes de consolidado.	Alta	
RF_007	Guardar la imagen en una DataSet	Como usuario administrador requiero almacenar las imágenes capturadas en un Dataset con la finalidad de utilizar para el entrenamiento del modelo de reconocimiento facial.	Alta	48h
RF_008	Realizar comparaciones de rostros	Como usuario final requiero que el sistema compare la foto de entrada con la imagen almacenada en la base de datos. Con el fin de encontrar un porcentaje de similitud.	Alta	72h
RF_009	Determinar la identidad de un estudiante	Como usuario final requiero que el sistema realice el reconocimiento facial en tiempo real, con la finalidad de disminuir el número de suplantaciones de identidad.	Alta	64h
RF_010	Validar el estudiante	Como estudiante requiero que el sistema valide mi identificación con el fin de poder ingresar al salón de clases.	Alta	48h
RF_011	Permitir el ingreso del estudiante según el porcentaje de similitud.	Como usuario final requiero que el sistema me brinde un porcentaje de similitud del estudiante que realice la identificación de identidad	Alta	56h
RF_012.	Listar Estudiantes	Como usuario final requiero visualizar los estudiantes existentes en el sistema para	Media	16h

			garantizar que no existan estudiantes duplicados		
RF_013	Listar usuario		Como usuario final requiero visualizar los usuarios existentes en el sistema para garantizar que no existan usuarios no permitidos al acceso del sistema.	Media	16h
RF_014	Generar un manual de usuario	de	Como usuario final requiero un manual de uso con el objetivo de conocer las funcionalidades del sistema.	Media	8h

4.1.5. Requerimientos no funcionales

Para realizar el análisis de los requerimientos no funcionales se tomará los requisitos indicados en el libro de ingeniería de *software* de *sommerville*, que se muestran en la figura 29.

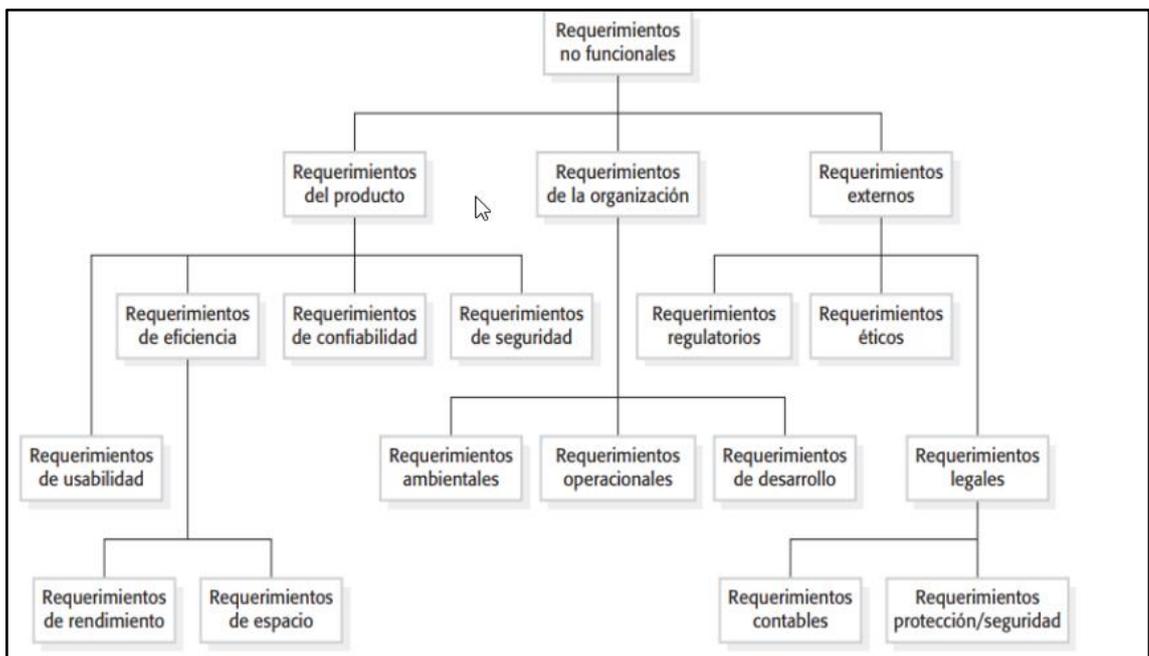


Figura 29. Requerimientos no funcionales
Tomada de *Xeridia (47)*

Los requerimientos no funcionales están relacionados con el funcionamiento y operación del *software*, describen las condiciones ambientales donde permanecerá activa la solución. Los requisitos no funcionales, están

relacionados con la capacidad, experiencia de usuario, disponibilidad, velocidad, seguridad y arquitectura de la información.

4.1.6. Requerimientos no funcionales del software

Se refieren a las cualidades, restricciones y características del software. A diferencia de los funcionales, no determinan una funcionalidad del sistema a desarrollar. A continuación, en la tabla 23, se describen los requerimientos no funcionales a nivel de *software*.

Tabla 23. Requerimientos no funcionales del software

N°	Requerimientos No Funcionales	Usuario
RNF1	Base de Datos	Administrador
RNF2	Requerimientos de usabilidad	Docente y Estudiante
RNF3	Requerimientos de rendimiento	Administrador
RNF4	Requerimientos de seguridad	Administrador
RNF5	Requerimientos de eficiencia	Docente
RNF6	Requerimientos de confidencialidad	Administrador
RNF7	Requerimientos de integridad	Administrador
RNF8	Requerimientos de disponibilidad	Administrador

De igual manera, en la tabla 24, tabla 25, tabla 26, tabla 27, tabla 28, tabla 29, tabla 30 y tabla 31, se describen a detalle cada requerimiento no funcional, especificando el nombre, descripción, indicador y prioridad.

Tabla 24. Requerimientos no funcionales – Base de datos

Nombre de requerimiento	Base de datos		
Descripción	El almacenamiento de base de datos tiene que realizarse en MySQL el cual permita el almacenamiento de información Seguro.		
Indicador	RNF01	Prioridad	Alta

Tabla 25. Requerimientos no funcionales – Requerimientos de usabilidad

Nombre de requerimiento	Requerimientos de usabilidad		
Descripción	Como usuario final requiero que el software de reconocimiento facial sea de uso fácil y sencillo.		
Indicador	RNF02	Prioridad	Alta

Tabla 26. Requerimientos no funcionales – Requerimientos de rendimiento

Nombre de requerimiento	Requerimientos de rendimiento		
Descripción	<ul style="list-style-type: none"> • Como usuario administrador requiero que el sistema tome una captura de imagen y lo guarde en el servidor local. • Como usuario administrador requiero que el sistema tenga un tiempo de respuesta de 10 segundos para realizar la captura de imagen. 		
Indicador	RNF03	Prioridad	Alta

Tabla 27. Requerimientos no funcionales – Requerimientos de seguridad

Nombre de requerimiento	Requerimientos de seguridad		
Descripción	Como usuario administrador es importante que el sistema autentique y autorice a los usuarios de acuerdo con el tipo y rol.		
Indicador	RNF04	Prioridad	Alta

Tabla 28. Requerimientos no funcionales – Requerimientos de eficiencia

Nombre de requerimiento	Requerimientos de eficiencia		
Descripción	Como usuario final requiero que el software debe realizar el reconocimiento facial en un tiempo máximo de 10 segundos.		
Indicador	RNF05	Prioridad	Alta

Tabla 29. Requerimientos no funcionales – Requerimientos de confidencialidad

Nombre de requerimiento	Requerimientos de confidencialidad		
Descripción	Como usuario administrador requiero tener el acceso seguro al sistema mediante la autenticación de usuario y contraseña, con ello garantizo que solo personas autorizadas tendrán acceso a la información.		
Indicador	RNF06	Prioridad	Alta

Tabla 30. Requerimientos no funcionales – Requerimientos de integridad

Nombre de requerimiento	Requerimientos de integridad		
Descripción	Como usuario administrador requiero que cualquier modificación o mantenimiento en el sistema, se realizara previa autorización al usuario encargado.		
Indicador	RNF07	Prioridad	Alta

Tabla 31. Requerimientos no funcionales – Requerimientos de disponibilidad

Nombre de Requerimiento	Requerimientos de disponibilidad		
Descripción	Como usuario administrador requiero que el sistema de escritorio tenga una alta disponibilidad en el acceso, para que el usuario encargado lo pueda usar en cualquier momento cuando sea necesario.		
Indicador	RNF08	Prioridad	Alta

4.1.7. Requerimientos no funcionales de la organización

En la tabla 32, se describen a detalle los requerimientos no funcionales entorno a la organización.

Tabla 32. Requerimientos no funcionales de la organización

ID	RNF	DESCRIPCIÓN	PRIORIDAD
RNF09	Requerimientos ambientales	Para un buen reconocimiento facial es necesario que se realice en un espacio iluminado.	ALTA
RNF010	Requerimientos operacionales	El reconocimiento facial debe funcionar con una cámara web estándar integrada a la PC.	ALTA
		Para el sistema es necesario un servidor en nodejs, base de datos en MySQL, librerías y el modelo de reconocimiento facial.	ALTA
RNF11	Requerimientos de desarrollo	Lenguaje de programación en Python para la parte del servidor y C# para el cliente.	ALTA
		El entorno de desarrollo de la base de datos en MySQL.	ALTA

4.2. Análisis de la solución

4.2.1. Propuesta de la solución

El presente proyecto propone una solución informática de bajo costo, donde se permitirá el reconocimiento facial para identificar a estudiantes durante los exámenes finales en clase de manera presencial, esta solución hará uso de diversas tecnologías *Open Source* como los algoritmos de *Haar-Cascade*, *Face-recognition* y las librerías *OpenCV*, *Numpy*, *Cmake* y *pip*. Estas herramientas permiten el análisis, detección y reconocimiento facial, en este sentido, cabe mencionar que de esta manera se pretende disminuir la cantidad de suplantación de identidad durante los exámenes finales, mayormente en los cursos generales.

El *software* de reconocimiento facial estará integrado a algoritmos eficientes que se implementaron en proyectos similares, este software en su primera versión tendrá las funcionalidades de registro de estudiantes, detectar el rostro y verificarlos con la base de datos para así confirmar la identidad del estudiante.

Todo este proceso se hará en una interfaz sencilla en un comienzo para facilitar su uso a los usuarios.

La solución más significativa en un comienzo es referida a la seguridad y la eficiencia en lo respecta a la identificación y reconocimiento de rostro y con esto disminuir las suplantaciones de identidad.

El *software* tiene como propósito realizar el reconocimiento facial de estudiantes para identificar suplantación de identidad en exámenes finales de manera presencial y brindar una alerta en tiempo real, así evitar suplantaciones de identidad, en la figura 30 se muestra el modelo conceptual de la solución: primero haciendo uso de las librerías ya mencionadas anteriormente y el modelo entrenado se procede a encender la cámara, se hace la detección de rostro, luego se hace el proceso de reconocimiento y la comparación con la base de datos, finalmente se hace la identificación del estudiante con su respectivo nombre.

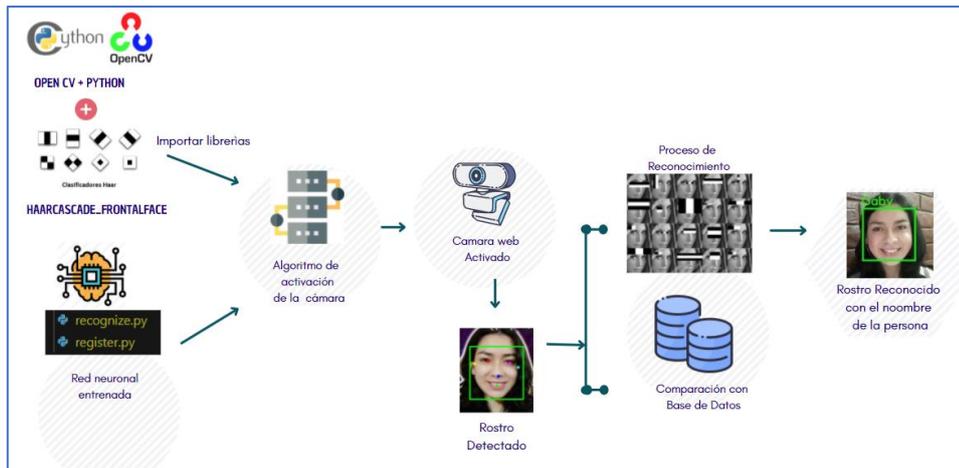


Figura 30. Modelo conceptual de la Solución Tomada de Solano (34)

4.2.2. Análisis de las especificaciones de los requerimientos funcionales

Dentro del proceso de implementación de la propuesta de la solución se encuentra la definición de los requerimientos de *software*, como son los requerimientos funcionales o requerimientos no funcionales, para el desarrollo del sistema se consideró los requerimientos que tengan una prioridad alta, por lo cual, se realiza un documento de especificación de forma estandarizado.

En la tabla 33, tabla 34, tabla 35, tabla 36, tabla 37 y tabla 38, se realiza el análisis de las especificaciones de cada uno de los requerimientos funcionales.

Tabla 33. Requerimiento funcional 001, Iniciar Sesión

Código	RF_001
Nombre	Iniciar sesión
Prioridad	Alta
Descripción	Se debe permitir iniciar sesión a un usuario que se encuentra registrado.
Usuario	Administrador
Entradas	<ul style="list-style-type: none"> • Usuario • Contraseña
Proceso	Una vez registrados los datos del usuario, se debe permitir el acceso al sistema.
Salida	Usuario Autenticado

Nota: Los requerimientos que se identificaron con una prioridad alta para el proceso del desarrollo del sistema de reconocimiento facial.

Tabla 34. Requerimiento funcional 002, Registrar usuario

Código	RF_002
Nombre	Registrar Usuario
Prioridad	Alta
Descripción	Se debe permitir registrar un usuario nuevo en la base de datos. Se debe contar con un formulario de registro de datos.
Usuario	Administrador
Entradas	<ul style="list-style-type: none"> • Nombres del usuario a registrar • Un único nombre de usuario • Contraseña
Proceso	Una vez registrados los datos del usuario y ejecutado el proceso, se debe permitir la autenticación del usuario nuevo.
Salidas	Creación de usuarios

Tabla 35. Requerimiento funcional 003, Registrar estudiante

Código	RF_003
Nombre	Registrar Estudiante
Prioridad	Alta
Descripción	Para que verificar la identidad de un estudiante es necesario permitir el registro.
Usuario	Usuario Final
Entradas	<ul style="list-style-type: none"> • DNI del estudiante. • Nombre del estudiante. • Apellido del estudiante
Proceso	Una vez registrados los datos del estudiante en la base de datos, se debe capturar la imagen del estudiante.
Salidas	Resultado de la operación de registro de estudiante

Tabla 36. Requerimiento funcional 004, Capturar rostro del estudiante

Código	RF_004
Nombre	Capturar rostro del estudiante
Prioridad	Alta
Descripción	El sistema debe permitir capturar imágenes sobre el rostro del estudiante para que sea de ayuda al momento del entrenamiento del modelo de reconocimiento facial y para la identificación de la persona.
Usuario	Usuario Final

Entradas	<ul style="list-style-type: none"> • La persona debe encontrarse ubicado al frente de la cámara web que está conectada al sistema durante 10 segundos.
Proceso	Se realiza la captura de las imágenes de forma automática.
Salidas	El sistema deberá guardar las imágenes capturadas en la carpeta que se encuentra en el disco local del ordenador.

Tabla 37. Requerimiento funcional 005, Reconocimiento facial

Código	RF_005
Nombre	Realizar el reconocimiento facial
Prioridad	Alta
Descripción	El sistema deberá identificar al estudiante si es perteneciendo o no al salón de clases.
Usuario	Usuario Final
Entradas	<ul style="list-style-type: none"> • Imágenes del estudiante objetivo mediante la cámara web en tiempo real.
Proceso	Se realiza la comparación sobre la base de datos de los estudiantes de forma identifique una suplantación de identidad.
Salidas	El sistema deberá retornar los nombres y apellidos del estudiante si pertenece a la clase, caso contrario manda un mensaje de estudiante desconocido.

Tabla 38: Requerimiento funcional 005, Registro de asistencia

Código	RF_006
Nombre	Registrar asistencia
Prioridad	Alta
Descripción	El sistema debe registrar la asistencia de los estudiantes.
Usuario	Usuario final
Entradas	<ul style="list-style-type: none"> • Imágenes del estudiante objetivo mediante la cámara web en tiempo real.
Proceso	Se realiza el registro de asistencia al momento de realizar la identificación de identidad del estudiante.
Salidas	El sistema deberá retornar los nombres y apellidos del estudiante si pertenece a la clase, caso contrario manda un mensaje de estudiante no identificado.

4.2.3. Análisis técnico y económico

4.2.3.1. Análisis técnico

Actualmente la universidad cuenta con una herramienta llamada *Proctorizer* que sirve para la supervisión remota de exámenes para sus evaluaciones digitales. Su función es eliminar el riesgo de fraudes, exámenes en grupo, plagio, el uso de internet para buscar las alternativas, entre otras faltas a la integridad académica, su enfoque es más para exámenes en línea.

A diferencia de esa herramienta, lo nuestro es un sistema de escritorio de reconocimiento facial enfocado a exámenes presenciales lo cual se cuenta con equipos que tienen características óptimas para el uso del proyecto propuesto.

A continuación, a detalle los requerimientos para la funcionalidad del sistema de escritorio de reconocimiento facial.

4.2.3.2. Requerimientos de hardware

En la tabla 39, se muestran los requerimientos de hardware que se necesitan para el desarrollo del proyecto.

Tabla 39. Requerimientos de hardware

N°	EQUIPOS
01	PC con procesador Core i5, Memoria RAM de 8GB, Almacenamiento de 1TB, SO Windows 10
02	Teclado para que los docentes puedan registrar a sus estudiantes.
03	Mouse se usará para facilitar el manejo del entorno gráfico
04	Monitor que es el principal dispositivo de salida
05	Laptop con procesador Core i5, Memoria RAM 8GB, almacenamiento de 1TB, Sistema Operativo Windows 10
06	Cámara web integrada a la pc o laptop
07	Cámara web

4.2.3.3. Herramientas para la construcción del modelo de reconocimiento facial

A continuación, en la tabla 40, se detallan los materiales y herramientas utilizados para la detección y el reconocimiento facial.

Tabla 40. Herramientas utilizadas en el desarrollo

TIPO	HERRAMIENTAS
Algoritmos	Haar Cascade, face-recognition
Lenguaje de programación	Python, JavaScript
Librerías	OpenCV, Numpy, Cmake, dlib, pip
Entorno de desarrollo	Node, C#

4.2.3.4. Análisis económico

Para el desarrollo del proyecto se realiza el análisis económico, costo de personal, costo de desarrollo, costo del software y hardware, este presupuesto se basa en una implementación piloto del simulador del sistema de escritorio de reconocimiento facial, los cuales se detallan a continuación:

Para el desarrollo del sistema es necesario contar con dos equipos de cómputo que tengan los siguientes requerimientos de hardware y software como mínimo, los cuales se especifican en la tabla 41 y tabla 42.

- **Costos de hardware**

Tabla 41. Costos de hardware

Equipo	Cant.	P. unitario	Precio total
Laptop o PC con procesador Core i5, Memoria RAM de 8GB, Almacenamiento de 1TB, SO Windows 10 que incluye cámara.	1	S/. 3,500.00	S/. 3,500.00
Total			S/. 3,500.00

- **Costos del software**

Tabla 42. Costos de software

Software	Licencia	Costo
Open CV	Libre	S/. 00.00
MySQL	Libre	S/. 00.00
Python	Libre	S/. 00.00
JavaScript	Libre	S/. 00.00
Windows 10	Libre	S/. 00.00
Total		S/.00.00

- **Costo del personal para el desarrollo**

Tabla 43. Costos de personal

Descripción	Cantidad	Precio
Desarrollador	1	360 horas
Costo por hora	1	S/. 10
Total		S/. 3,600.00

- **Costos operativos**

Tabla 44: Costos operativos

Descripción	Cantidad x meses	Costo	Costo total
Infraestructura	5	S/. 50.00	S/. 250.00
Mantenimiento	5	S/. 50.00	S/. 250.00
Total			S/. 500.00

- **Costo total del proyecto – piloto**

La tabla 45, se hace un resumen del costo total que tendría el proyecto Piloto.

Tabla 45. Costo total

Hardware	S/. 2,500.00
Costo desarrollador	S/. 3,600.00
Software	S/. 00.00
Costos Operativos	S/. 500.00
Total	S/. 6,600.00

Entonces, se puede concluir que para el desarrollo del proyecto y poder cumplir con los objetivos establecidos, se realizó el análisis de tres ítems altamente importantes que nos dan resultados efectivos y concretos para determinar dicha viabilidad, se estudió en el caso hipotético.

4.2.4. Viabilidad de la propuesta

4.2.4.1. Viabilidad técnica

Para evaluar la viabilidad técnica de la solución se realiza definición, su aplicación y las diferencias entre las librerías *TensorFlow* y *OpenCV*. En la tabla 46 se evidencia a detalle esta comparación realizada.

Tabla 46. TensorFlow vs OpenCV

TensorFlow	OpenCV
Una plataforma de código abierto que funciona con tensores, que es una generalización de vectores.	Una biblioteca multiplataforma de código abierto que funciona con visión artificial en tiempo real
Está diseñado para soluciones matemáticas mediante el uso de diagramas de flujo de datos.	Está diseñado para aumentar la eficiencia computacional de las aplicaciones en tiempo real.
Herramienta de aprendizaje automático	Herramienta de procesamiento de imágenes
Coders trabaja en TensorFlow para un alto rendimiento	OpenCV, seleccionado por los desarrolladores para "Computer Vision".

TensorFlow es un marco (base donde se desarrolla el software) para la inteligencia de las máquinas. OpenCV es una biblioteca para la visión por computadora.

Las aplicaciones son detección de patrones, cerraduras inteligentes, autos sin conductor.	Las aplicaciones son procesamiento de imágenes, detección de rostros, reconocimiento de gestos.
Admite lenguaje C, C++, Java y Python	Admite lenguaje C++ y Python

Aplicaciones	
Aplicaciones basadas en texto, que son casos de uso de redes neuronales profundas	Detección y reconocimiento de rostros
Detección de idioma, correo electrónico no deseado y detección de amenazas.	Reconocimiento de gestos
El traductor de Google - aplicaciones basadas en texto.	Detección de imágenes y objetos
	Procesamiento de imágenes

Tomado y adaptado de Code (48)

La idea primaria del proyecto era desarrollar un sistema de reconocimiento facial utilizando Tensor Flow, una herramienta muy poderosa y multiusos basada en *machine-learning* (49) la cual se encuentra disponible en su repositorio oficial (50). Por tanto, después de evaluar las alternativas se decidió usar la biblioteca de OpenCV juntamente con la librería de face recognition que ayudan en la detección y reconocimiento de rostros. También se puede decir que el proyecto es viable técnicamente ya que existe documentación y soporte suficiente respecto a estas librerías, asimismo, existe librerías gratuitas de código abierto para el reconocimiento facial con OpenCV. Y para implementar el código Visual Studio 2019, Python y para administrar la base de datos MySQL. Esto debido que hay más soporte para implantar algoritmos pre entrenados, debido al tiempo. Además del hardware propio que cumple con los requisitos mínimos para desarrollar este tipo de proyecto.

El sistema deberá desarrollarse considerando las siguientes restricciones técnicas de la tabla 47.

Tabla 47. Restricciones técnicas

Restricción	Área de desarrollo	Descripción
Lenguaje de programación - Python	Entrenamiento del modelo de reconocimiento facial	Se deberá implementar el modelo de reconocimiento facial en el lenguaje de programación Python.
Lenguaje de programación – C#	Implementación de la funcionalidad del sistema. Implementación de interfaces de usuario.	La programación de interfaces y funcionalidades del sistema em el lenguaje C# con Microsoft visual studio 2019
Servidor de Gestor de Base de datos	Sistema de gestión de base de datos - MySQL	El sistema de gestión de base de datos será MySQL, dado que es muy eficiente, permitiendo conexiones seguras y consultas eficientes.
Servidor API REST - Nodejs	Api rest – Node JS, Api del modelo de reconocimiento facial	Para la programación de la API se desarrollará en el lenguaje JavaScript con el servidor NODEJS.
Planificación - Trello	Gestión de planificación del proyecto	La planificación del proyecto se realizará en el software de trello, el cual nos permite tener un trabajo eficiente y controlado.
Edición de código	Editor de lenguaje de programación	La edición de código del modelo de reconocimiento se realizará en el software visual Code.

4.3. Diseño

Durante esta etapa se busca conseguir una estructura de software que funcione eficientemente, operando siempre en base a los requerimientos definidos anteriormente.

4.3.1. Diseño de la arquitectura del sistema

El diseño de la arquitectura se dividirá en dos partes, arquitectura de software y hardware. Los cuales se explican a continuación.

4.3.1.1. Arquitectura del software

Para tener una visión general es muy importante ver la funcionalidad del modelo de reconocimiento facial a modo general, la solución del problema se crea en un servicio con NodeJS, para la detección y reconocimiento facial se realiza usando la librería de face-recognition con el algoritmo Haarcascade, la interfaz se realiza con visual Basic en el lenguaje de programación C#, finalmente para la base de datos se usará MySQL WorkBench. En la Figura 31 se observa la arquitectura implementada.

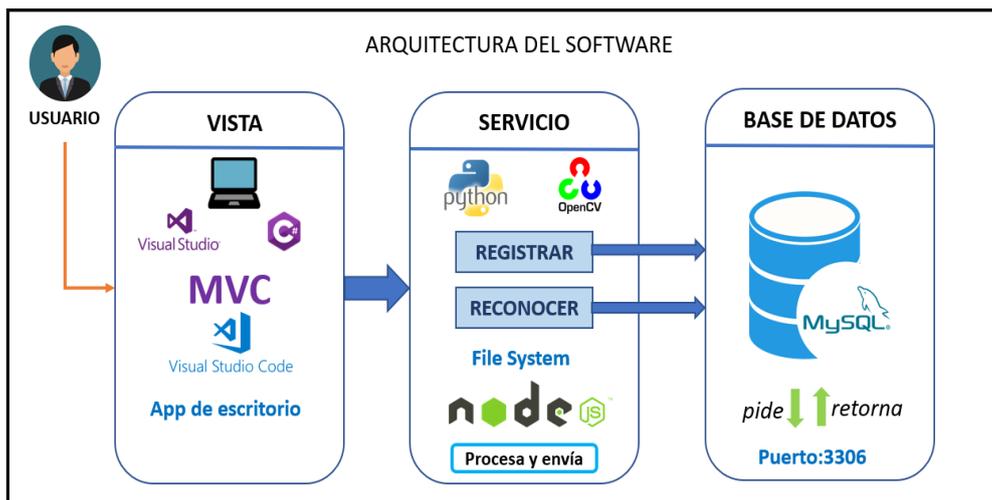


Figura 31. Arquitectura del software

4.3.1.2. Arquitectura del hardware

Para tener una visión de la funcionalidad del sistema es muy importante ver la arquitectura de hardware donde tenemos una PC para la interacción del usuario donde se muestra el ejecutable del sistema de escritorio de reconocimiento facial elaborado con el *Net Framework – C#*, donde se comunica con el servidor de *Api Rest de NodeJs* por el puerto 3000 y para el reconocimiento se ejecuta el Api de la cámara posteriormente poder hacer la captura del rostro, lo cual hace la comparación con la base de datos que se encuentra en nuestro Sistema de Gestión de Base de Datos de MySQL que se comunica por el puerto 3306. En la Figura 32 se observa la arquitectura de hardware implementada.

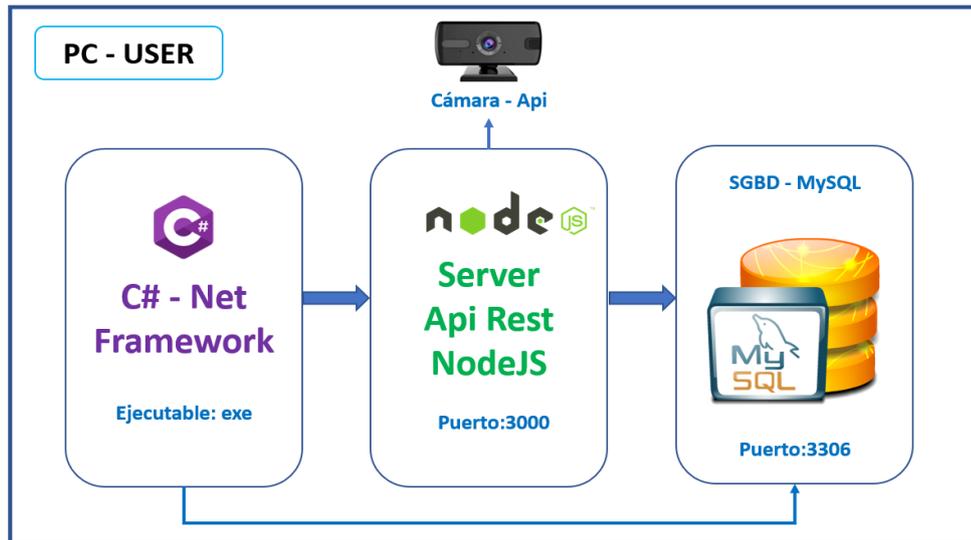


Figura 32. Arquitectura del hardware

4.3.2. Herramientas

- **Python:** es un lenguaje de programación muy usado en aprendizaje automático y aprendizaje profundo, gracias a *Python* la librería de *OpenCV* es más sencilla de funcionar.
- **Django:** es un *framework* para el desarrollo de sitios web de código abierto seguros y mantenibles.
- **Angular:** es un *framework* para JavaScript que sirven para hacer aplicaciones y páginas web está enfocado en las vistas de las páginas web (*frontend*).
- **Face-recognition:** es un algoritmo que ayuda a reconocer el rostro de una persona y está basado en características geométricas.
- **OpenCV:** es una biblioteca que está enfocado en los procesamientos de imágenes y videos en tiempo real, es de libre visión artificial que es desarrollado por Intel.
- **Base de Datos MySQL:** después de tener las imágenes vamos a preparar una base de datos de individuos que queremos que reconozca nuestro sistema. La base de datos también es conocida como una colección de información para una manera rápida de obtener los datos que se necesite.

- **Cmake:** se utilizó esta librería para controlar el proceso de compilación y administrar.
- **Dlib:** es una librería que ayuda en la extracción de los puntos del rostro y extrae 68 puntos aproximados para el reconocimiento facial.
- **Numpy:** es una librería de funciones matemáticas que ayuda a operar con vectores y matrices para tener acceso más rápido a la lectura y escritura de los elementos.
- **Node:** es un entorno de tiempo de ejecución de JavaScript que no solo es para JavaScript sino también para Python.
- **Pip:** es una herramienta útil para administrar e instalar paquetes de Python de una forma rápida.
- **Kanban:** es una metodología lo cual te ayuda a desarrollar y gestionar proyectos con constantes evaluaciones para verificar cómo se está avanzando el proyecto que es más enfocado a proyectos individuales.

4.3.3. Diseño del reconocimiento facial

Para el reconocimiento facial se debe tener en cuenta los siguientes pasos que se muestra en la figura 33, el primero se basa en detectar los rostros, luego se realiza la comparación, para definir los puntos de coincidencia y finalmente se clasifica.

El sistema de escritorio crea automáticamente un directorio, donde se almacenan las imágenes clasificadas por nombre del estudiante con el cual se realiza la comprobación del rostro para después obtener un resultado de la detección del reconocimiento facial.

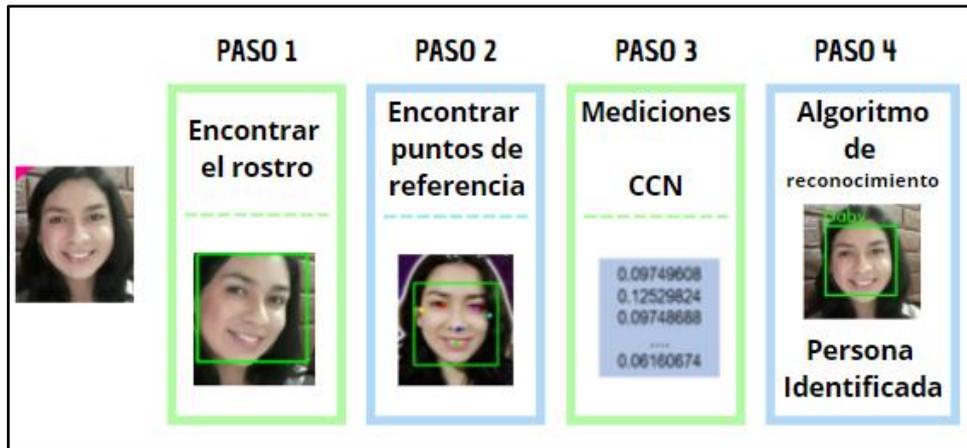


Figura 33. Pasos para el reconocimiento facial

Tal como se muestra en la figura 33, los pasos que se siguen para realizar el reconocimiento facial son: Paso 1, encontrar el rostro, paso 2, encontrar los puntos de referencia, paso 3, aplicación de red neuronal, paso 4, reconocimiento de la persona.

4.3.4. Diseño de interfaz y navegación

Las interfaces están diseñadas para proporcionar una estructura operativa eficiente que cumpla consistentemente con los requisitos enumerados en la tabla 3: que corresponde a los requerimientos funcionales.

El sistema de reconocimiento facial estará compuesto por una interfaz de login, interfaz de la pantalla principal, interfaz para realizar la captura de rostro y posteriormente realizar el reconocimiento facial. Los *mockups* están elaborados en el software de Balsamiq los cuales se detallan a continuación

4.3.5. Interfaz de login

En la figura 34, login estará compuesto básicamente por una interfaz sencilla, el cual tendrá dos campos de input para el nombre de usuario y la contraseña.



Figura 34. Mockup de inicio sesión

4.3.6. Pantalla de principal

La pantalla principal del sistema está compuesta por una cabecera, un menú principal y un pie de página. En la figura 35 se puede evidenciar a detalle.

El logo: en esta sección se encuentra el logo representativo de la universidad Continental. En la cabecera: en este apartado encontramos el título del proyecto, en el menú principal se encuentran las funcionalidades principales del sistema.

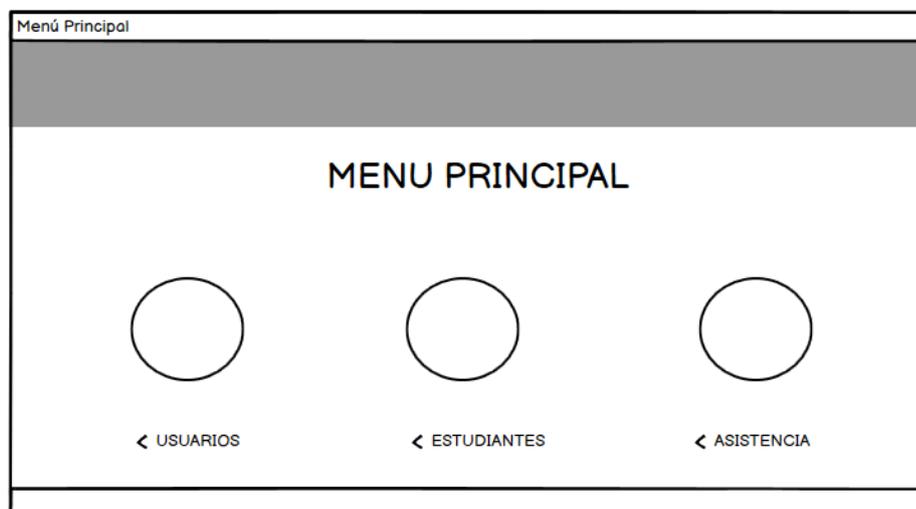


Figura 35. Mockup de la pantalla principal

4.3.7. Pantalla administrar usuarios

La pantalla administrar usuarios está compuesto por un menú superior, cuerpo y pie de página. En el menú principal se encuentra opciones básicas, donde se podrá regresar a la pantalla principal, ver la lista de usuarios y registrar uno nuevo, en el pie de página se muestra información básica acerca del sistema. En la figura 36, se evidencia diseño del mockup.

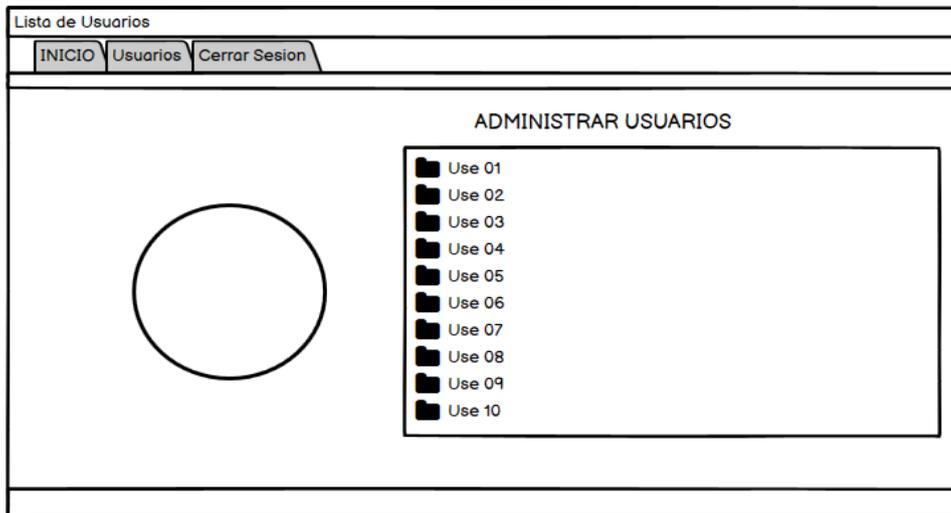


Figura 36. Mockup Pantalla administrar usuarios

4.3.8. Formulario registrar usuario

Para registrar un nuevo usuario al sistema, en donde se realizará el ingreso de datos como nombre de usuario, contraseña, nombre y apellido. En la figura 37, se plasma el mockup para registrar usuario.

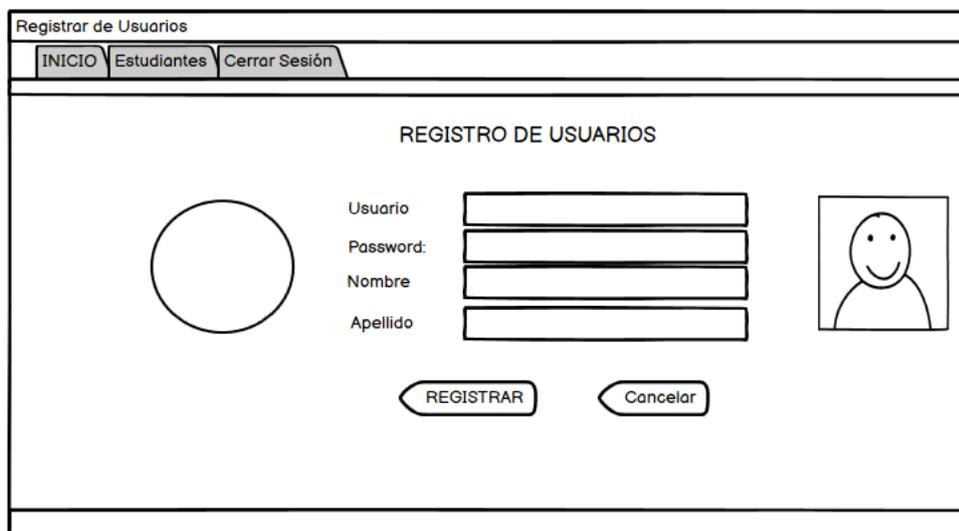


Figura 37. Mockup registrar usuario

4.3.9. Pantalla administrar estudiantes

La pantalla administrar estudiantes contiene un menú superior, cuerpo y pie de página. En el menú superior se encuentra opciones básicas, donde se podrá regresar a la pantalla principal, ver la lista de estudiantes, registrar estudiante y realizar el reconocimiento facial, en el pie de página se muestra información básica acerca del sistema. En la figura 38, se evidencia diseño del *mockup*.

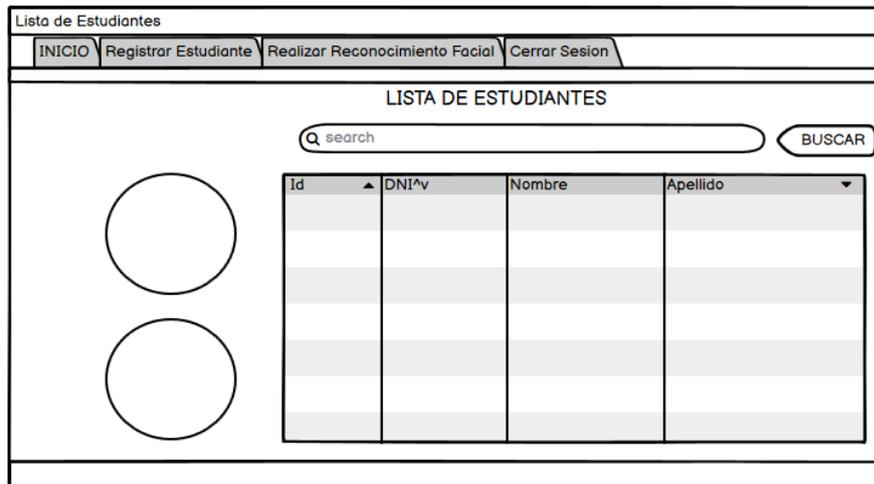


Figura 38. Mockup pantalla administrar estudiante

4.3.10. Formulario registrar estudiante

Para registrar un nuevo estudiante en el sistema, se realizará el ingreso de datos, nombre y apellido, para la captura de rostro también es importante completar estos campos. En la figura 39, se plasma el mockup para registrar estudiante.

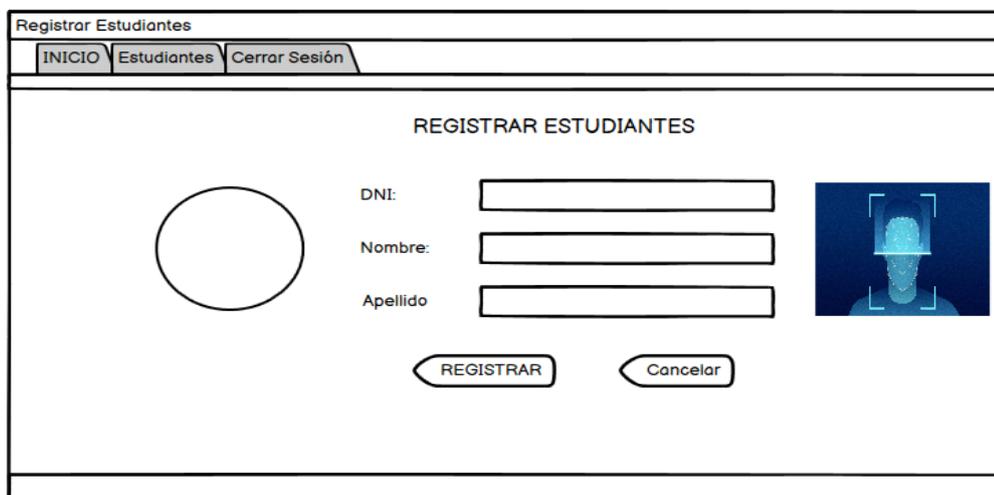


Figura 39. Mockup formulario registrar estudiante

4.3.11. Formulario reconocimiento facial

En el formulario de reconocimiento facial, tenemos el menú superior con opciones de regresar al menú principal y realizar el reconocimiento, en esta sección se pide dos campos tipo input como son nombre y apellido. Y como salida se obtiene un label con el nombre del estudiante en caso de halla coincidencias, caso contrario un label con la descripción de desconocido. En la figura 40, se plasma el *mockup*.

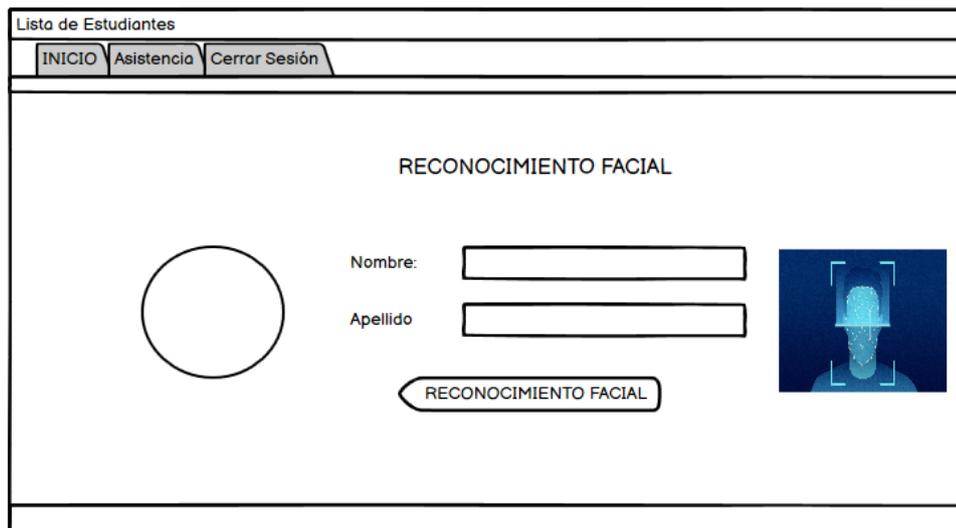


Figura 40. Mockup reconocimiento facial

4.3.12. Formulario de asistencia

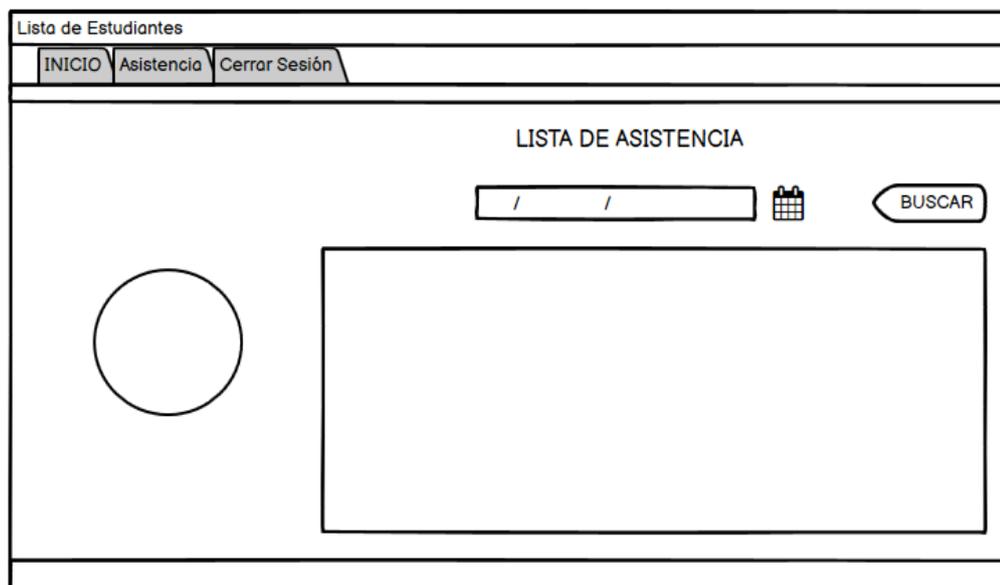


Figura 41. Mockup registro de asistencia

4.3.13. Diseño de la base de datos

El motor de la base de datos que fue elegido fue MySQL workbench, por la factibilidad y facilidad de entendimiento para ejecutar todo tipo de información. Para evaluar el algoritmo de reconocimiento facial es necesario una base de datos por esta razón se realiza el diseño de la base de datos para responder con las características de la solución propuesta.

En la figura 42, se muestra el esquema de base de datos; el diseño lógico, donde cuenta con cuatro tablas importantes: *tblusuario*, *tbltipousuario*, *tblestudiante* y *tblasistenciaestudiante*.

En la tabla *tblusuario* se registran datos de los usuarios como nombre de usuario, contraseña, apellidos, nombres, y tipo de usuario, quienes van a ser los encargados de manipular el sistema.

En la tabla *tbltipousuario* se encarga de almacenar al tipo de usuario según los roles para actualizar, eliminar y registrar. Por otro lado, la tabla *tblestudiante* se encuentra diseñada para almacenar nuevos estudiantes con sus respectivos datos como DNI, nombre, apellido, edad y foto. Finalmente tenemos a la tabla *tblasistenciaestudiante* lo cual nos permite obtener una información relativa de la asistencia del estudiante al momento de realizar el reconocimiento facial.

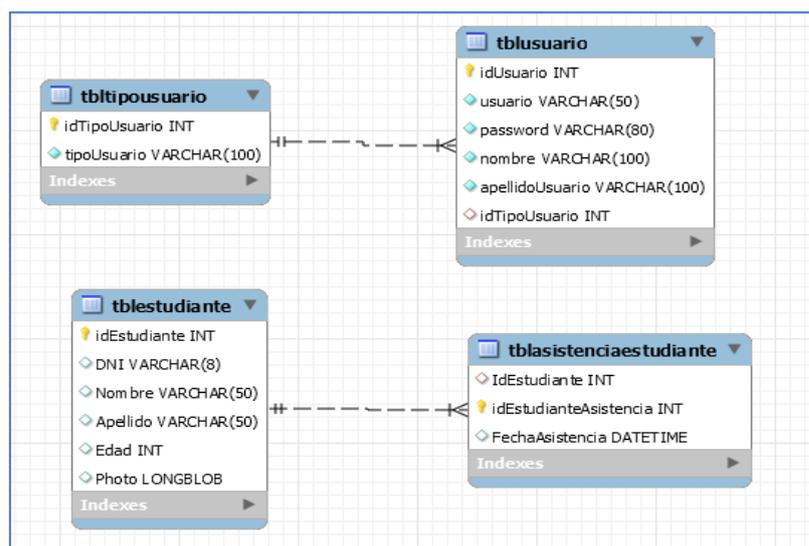


Figura 42. Diseño lógico

Con los requerimientos obtenidos y el diseño desarrollado en la figura 42 del modelo Entidad relación de la base de datos, se procedió a elaborar la base de datos en el entorno MySQL Workbench, para almacenar los registros que se necesitan en la solución del sistema. Esta base de datos está compuesta por tablas, las cuales se describen a la tabla 48.

Tabla 48. Tablas de la base de datos

ÍTEM	NOMBRE DE LA TABLA
01	TblUsuario
02	TblTipoUsuario
03	TblEstudiante
04	TblAsistenciaEstudiante

En la figura 43, se muestra cada campo utilizado en la base de datos.

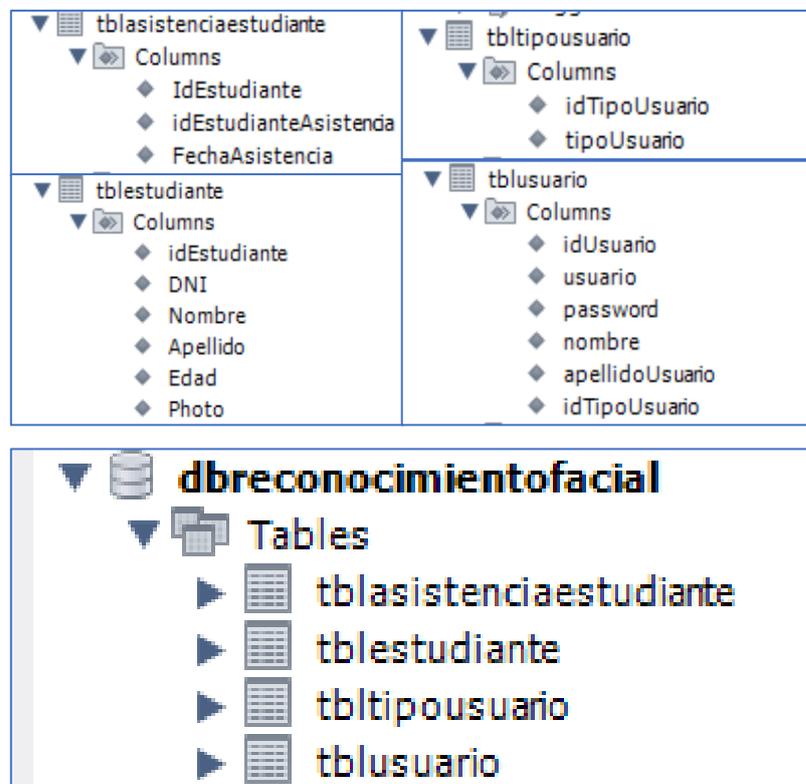


Figura 43. Tabla de la base de datos

CAPÍTULO V CONSTRUCCIÓN

5.1. Construcción

Para la etapa de construcción, todos los componentes y características se integran y desarrollan como un solo producto. También se van haciendo pruebas de validación durante la codificación.

En esta etapa se desarrollan todas las vistas (lógicos, procesos, despliegue e implementación) de todos los componentes del software, se priorizan los módulos principales y se desarrollan para obtener una versión funcional del software de reconocimiento facial.

La etapa de construcción se encuentra dividida en dos etapas: Modelamiento de la red neuronal artificial y construcción del software, las cuales se describen a continuación:

5.5.1. Modelamiento de la red neuronal artificial y el Machine Learning

Para el modelamiento de la red neuronal se utilizó el método Haar Cascade (37), que consta de tres etapas fundamentales: 1) rastrea la imagen en busca de características Haar que se asemejen a los objetos buscados; 2) utiliza un clasificador boosting para seleccionar las características más relevantes, y 3) conecta en cascada los clasificadores, con el objetivo de mejorar los resultados finales.

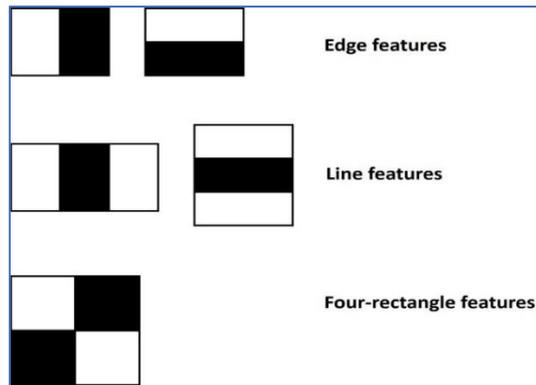


Figura 44. Núcleos utilizados en Haar Cascade
Tomado y adaptado de Mordvintsev (51)

La Figura 44 ejemplifica los tipos de núcleos manuales utilizados para la detección de Haar Cascade: la imagen (a) hace referencia a los granos para identificar las características del borde; la imagen (b) hace referencia a los granos para identificar las características de línea, y la imagen (c) hace referencia a los granos como cuatro rectángulos, usados para reconocer bordes diagonales.

5.5.2. Creando una imagen integral

Una imagen integral es una técnica que ayuda acelerar el cálculo de operaciones que incluyan la suma del valor de los píxeles de un área. Para crear la imagen integral hay que reemplazar cada píxel por la suma de todos píxeles contenidos en un rectángulo cuya esquina superior izquierda es el vértice 0,0 de la imagen. Y cuya esquina inferior derecha es el propio píxel.

Para entender partimos de una imagen con los siguientes píxeles que se muestran en la figura 45:

Input image					Integral image				
4	1	3	7	9	4	5	8	15	24
2	5	9	3	7	6	12	24	34	50
1	3	7	2	5	7	16	35	47	68
7	8	6	1	4	14	31	56	69	94
6	1	7	8	7	20	38	70	91	123

Figura 45. Imagen de entrada – imagen integral
Tomado y adaptado de Mordvintsev (51)

La figura 46 muestra que cada píxel en la imagen integral es la suma de todos los píxeles vecinos que están arriba y a la izquierda de ese píxel, incluido ese píxel mismo. Con este método podemos calcular fácil y rápidamente cualquier área de la imagen que queramos.

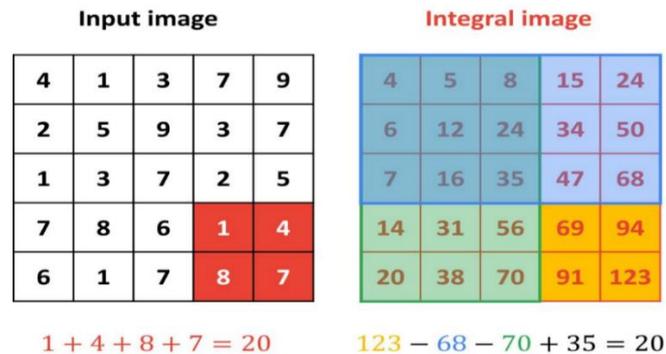


Figura 46. Imagen de entrada – imagen integral
Tomado y adaptado de Mordvintsev (51)

En la figura 47, se muestra un ejemplo de una imagen integral en posición (x, y).



Figura 47. Imagen integral en posición (x, y)
Tomado y adaptado de Mordvintsev (51)

La imagen integral en la localización x, y, contiene la suma de los píxeles de la parte superior izquierda de la imagen y se puede calcular como se indica a continuación:

$$II(x, y) = \sum_{x' \leq x, y' \leq y} Im(x', y')$$

Donde $II(x, y)$ es la imagen integral e $Im(x, y)$ es la imagen original.

5.5.3. Extracción de características

En las imágenes, las características de cada objeto son aplicando ciertas funciones que permiten la representación y descripción de los objetos de interés en la imagen.

Los filtros con bases Haar, realizan una codificación de diferencia de intensidades en la imagen, generando características de contornos, puntos y líneas, mediante la captura de contraste entre regiones como se muestra en la figura 48.

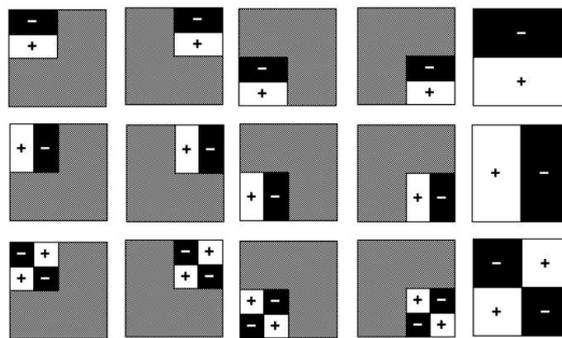


Figura 48. Filtros Haar rotados, trasladados y con cambios de escala

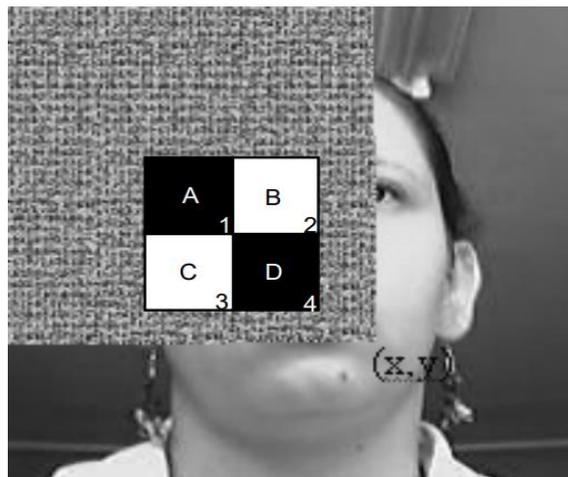


Figura 49. Convolución del filtro Haar con una imagen integral

En la figura 49, se presenta la convolución de un filtro Haar con la imagen integral. De esta forma se puede extraer una característica en un tiempo constante sobre la imagen integral adicionando y sustrayendo los valores de los vértices para cada rectángulo. La suma de los píxeles que forman el rectángulo D se puede calcular como:

$$sumD = (4 + 1) - (2 + 3)$$

Donde 1, 2, 3, 4, son los valores dados en la imagen integral en dichas localizaciones.

5.5.4. Escritura de código de red neuronal

El primer módulo que se va a importar es CV2 es opencv, que se utiliza para llamar la cámara y realizar algún procedimiento. Otro modulo que se va a importar es *face_recognition* se usa para realizar el reconocimiento facial y por último el módulo *os* que se utiliza para identificar el nombre de la cara que aparece en la cámara. En la figura 50, se evidencia la importación de estos módulos.

```
1 # coding: utf-8
2 # Your code here!
3 from cv2 import Mat_MAGIC_VAL, cv2
4 import face_recognition
5 import os
6
```

*Figura 50. Importación de módulos cv2, face_recognition y os
Tomada y adaptada de Visual Studio Code*

En la figura 51 importamos numpy con un alias np

```
import numpy as np
import sys
import MySQLdb as mysql
```

*Figura 51. Importación de módulos numpy, sys y mysql
Tomada y adaptada de Visual Studio Code*

En la figura 52, añadimos y leemos nuestra imagen con el `cv2.imread()`, en el array `IMAGES` añadiremos cada imagen (rostro).

```
for _file in LIST:
    curImg = cv2.imread(f'{PATH}/{_file}')
    IMAGES.append(curImg)
```

*Figura 52. Carga de imagen
Tomada y adaptada de Visual Studio Code*

En la figura 53, en las siguientes líneas del código llamamos `face_cascade`

```

casc_path=os.path.dirname(cv2.__file__)+'/data/haarcascade_frontalface_default.xml'
face_cascade = cv2.CascadeClassifier(casc_path)
detected_count = 0

```

Figura 53. Carga de imagen
Tomada y adaptada de Visual Studio Code

A continuación, debemos convertir nuestra imagen a escala de grises porque las cascadas de Haar solo funcionan en imágenes grises. Entonces, vamos a detectar rostros, ojos y sonrisas en imágenes en escala de grises, pero dibujaremos rectángulos alrededor de los rostros detectados en las imágenes en color.

Para emplear la detección de rostros (y otro objeto) con haar cascade en OpenCV necesitamos del módulo **detectMultiScale**. Este nos permitirá obtener un rectángulo delimitador en donde se encuentre el objeto dentro de una imagen que se muestra en la figura 54.

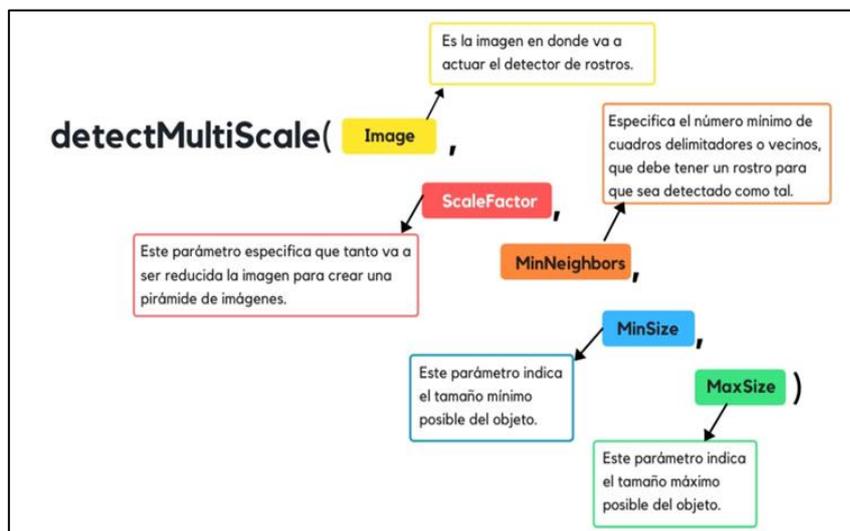


Figura 54. Especificaciones del módulo DetectMultiScale en OpenCV
Tomado de Solano (34)

En el primer paso detectaremos la cara. Para extraer las coordenadas de un rectángulo que vamos a dibujar alrededor de la cara detectada, necesitamos crear caras de objetos.

En este objeto vamos a almacenar nuestras caras detectadas. Con una función **detectMultiScale()** obtendremos una tupla de cuatro elementos: X, Y son las coordenadas de una esquina superior izquierda, y, w, y, h son el ancho y la altura del rectángulo. Este método requiere varios argumentos. La primera

es la imagen gris, la imagen de entrada en la que detectaremos rostros. El segundo argumento es el factor de escala que nos dice cuánto se reduce el tamaño de la imagen en cada escala de imagen. El tercer y último argumento es el número mínimo de vecinos. Este parámetro especifica cuántos vecinos debe tener cada rectángulo candidato para conservarlo, el código empleado usando el módulo **detectMultiScale()** se muestra en la figura 55.

```
faces = face_cascade.detectMultiScale(  
    gray,  
    scaleFactor=1.2,  
    minNeighbors=5,  
    minSize=(200, 200),  
    flags=cv2.CASCADE_SCALE_IMAGE  
)  
  
for(x , y, w, h) in faces:  
    cv2.rectangle(img, (x,y) ,(x+w, y+h), (0,255,0), 3)  
cv2_imshow(img)
```

Figura 55. Argumentos del módulo DetectMultiScale en OpenCV
Tomada y adaptada de Visual Studio Code

En la figura 56 podemos evidenciar el resultado de la ejecución de código al utilizar el módulo **DetectMultiScale**, donde nos grafica un rectángulo alrededor del rostro identificado.

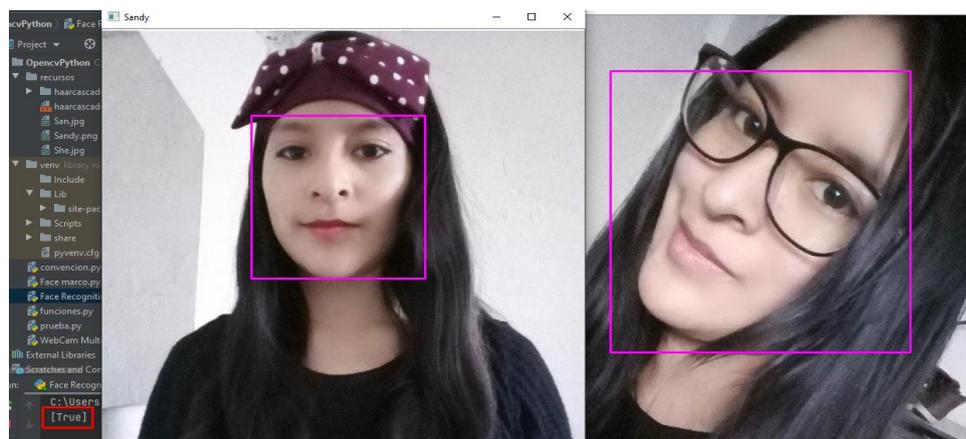


Figura 56. Resultado del módulo DetectMultiScale en OpenCV
Tomada y adaptada de Visual Studio Code

En la figura 57, es la estructura del directorio donde se guarda las caras. De forma predeterminada, el nombre de la imagen es el nombre de la persona, lo que facilita su obtención.

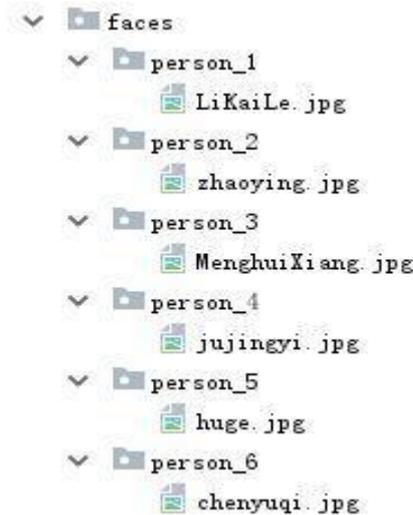


Figura 57: Estructura del directorio Tomada de Solano (34)

La arquitectura capas-neuronas de una red neuronal artificial no es nunca constante y depende en ocasiones del problema y depende de la cantidad de personas que se va a identificar es por ello por lo que se implementó usando el perceptrón simple, para comprobar en tiempo de entrenamiento, ya que al momento de procesar las imágenes se requiere de gran recurso computacional y lo que se busca es optimizar el algoritmo al máximo.

5.5.5. Interfaz de inicio de sesión

El inicio de sesión del sistema se realiza de manera tradicional donde se pide ingresar las credenciales del usuario, ver figura 58. Los resultados obtenidos en la parte de construcción.



Figura 58. Interfaz inicio de sesión

En la figura 58, esta interfaz corresponde para el inicio de sesión del usuario en el sistema, donde el usuario docente tiene la opción de ingresar su nombre de usuario y contraseña para acceder. Y en la figura 58 se evidencia la parte del código del desarrollo.

```

1 referencia
private void btnIniciar_Click(object sender, EventArgs e)
{
    string usuario = txtUsuario.Text;
    string password = txtPassword.Text;

    try
    {
        ControladorUsuario ctrl = new ControladorUsuario();
        string rpt = ctrl.ctrLogin(usuario,password);
        if (rpt.Length>0)
        {
            MessageBox.Show(rpt, "Aviso", MessageBoxButtons.OK, MessageBoxIcon.Warning);
        }
        else
        {
            frmPrincipal frmPrincipal = new frmPrincipal();
            frmPrincipal.Visible = true;
            this.Visible = false;
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "aviso", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
    }
}

```

Figura 59. Código de inicio de sesión -1
Tomada del código de programación en C#

```

/*METODO de tipo usuarios*/
1 referencia
public Usuarios porUsuario(string usuario)
{
    MySqlDataReader reader;
    MySqlConnection conexion = clsConexion.getConexion();
    conexion.Open();

    string sql = "SELECT idUsuario, password, nombre, idTipoUsuario FROM tblUsuario WHERE u
    MySqlCommand comando = new MySqlCommand(sql, conexion);
    comando.Parameters.AddWithValue("@usuario", usuario);

    reader = comando.ExecuteReader();

    Usuarios usr = null;

    while (reader.Read())
    {
        usr = new Usuarios();
        usr.Id =int.Parse(reader["idUsuario"].ToString());
        usr.Password = reader["password"].ToString();
        usr.Nombre = reader["nombre"].ToString();
        usr.IdTipoUsuario = int.Parse(reader["idTipoUsuario"].ToString());
    }

    return usr;
}

```

```

public String ctrLogin(string usuario, string password)
{
    ModeloUsuario modeloUsuario = new ModeloUsuario();
    string respuesta = "";
    Usuarios datosUsuarios = null;
    if (string.IsNullOrEmpty(usuario) || string.IsNullOrEmpty(password))
    {
        respuesta = "Debe llenar todos los campos";
    }
    else
    {
        datosUsuarios = modeloUsuario.porUsuario(usuario);
        if ( datosUsuarios==null)
        {
            respuesta = "El Usuario no existe";
        }
        else
        {
            if (datosUsuarios.Password!= password)
            {
                respuesta = "El usuario y/o contraseña no coinciden";
            }
        }
    }
    return respuesta;
}
}

```

Figura 60. Código de inicio de sesión -2
Tomada del código de programación en C#

```

namespace ReconocimientoFacial
{
    8 referencias
    class Usuarios
    {
        int id, idTipoUsuario;
        string user, password, nombre, apellido;

        4 referencias
        public string User { get => user; set => user = value; }
        7 referencias
        public string Password { get => password; set => password = value; }
        4 referencias
        public string Nombre { get => nombre; set => nombre = value; }
        3 referencias
        public string Apellido { get => apellido; set => apellido = value; }
        1 referencia
        public int Id { get => id; set => id = value; }
        1 referencia
        public int IdTipoUsuario { get => idTipoUsuario; set => idTipoUsuario = value; }
    }
}

```

Figura 61: Clase de la tabla usuario
Tomada del código de programación en C#

5.5.6. Interfaz de la pantalla principal



Figura 62. Menú principal
Tomada y adaptada de Visual Studio Code

La figura 62, corresponde a la interfaz del menú principal del sistema donde se muestran las opciones de navegación, donde se tiene las opciones de administrar usuario, estudiantes y asistencia.

```
7 using System.Text;
8 using System.Threading.Tasks;
9 using System.Windows.Forms;
10
11 namespace ReconocimientoFacial
12 {
13     19 referencias
14     public partial class frmPrincipal : Form
15     {
16         int TipoUsuario;
17         8 referencias
18         public frmPrincipal()...
19
20         1 referencia
21         private void frmPrincipal_Load(object sender, EventArgs e)...
22
23         1 referencia
24         private void frmPrincipal_FormClosed(object sender, FormClosedEventArgs e)...
25
26         1 referencia
27         private void pictureBox5_Click(object sender, EventArgs e)
28         {
29             frmListaEstudiantes frm = new frmListaEstudiantes();
30             frm.Visible = true;
31             this.Hide();
32         }
33
34         1 referencia
35         private void pictureBox1_Click(object sender, EventArgs e)
36         {
37             frmListaUsuarios frm = new frmListaUsuarios();
38             frm.Visible = true;
39             this.Hide();
40         }
41     }
42 }
```

Figura 63. Código de menú principal
Tomada del código de programación en C#

5.5.7. Interfaz administrar usuarios



Figura 64: interfaz administrar usuarios
Tomada del código de programación en C#

5.5.8. Interfaz registrar usuario

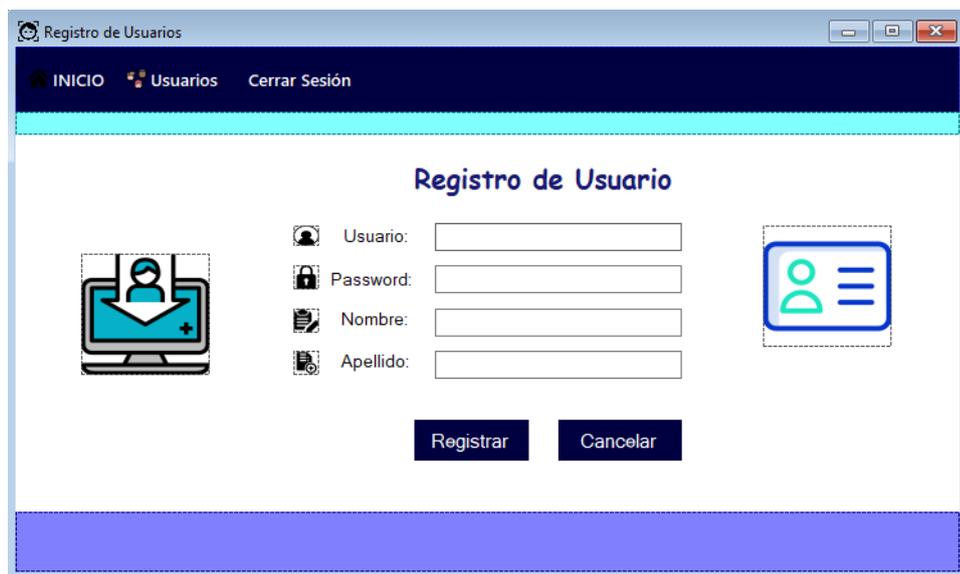


Figura 65. Interfaz registrar usuario
Tomada del código de programación en C#

Para registrar un nuevo usuario al sistema, en donde se realizará el ingreso de datos como nombre de usuario, contraseña, nombre y apellido. En la figura 65, construye la interfaz para registrar usuario.

5.5.9. Interfaz administrar estudiantes



Figura 66. Interfaz administrar estudiante, construcción de la interfaz Tomada del código de programación en C#

5.5.10. Formulario registrar estudiante

En la figura 67, se plasma el mockup para registrar estudiante. Y en la figura 68 el código empleado para el desarrollo de la interfaz.



Figura 67. interfaz registrar estudiantes Tomada del código de programación en C#

```

private void btnRegistrarEstudiante_Click(object sender, EventArgs e)
{
    string DNI = txtDNIEstudiante.Text;
    string Nombre = txtNombreEstudiante.Text;
    string Apellido = txtApellidoEstudiante.Text;
    int Edad = Convert.ToInt32(ndEdadEstudiante.Value);

    // HttpClient client = new HttpClient();
    using (var wb = new WebClient())
    {
        var data = new NameValueCollection();
        data["DNI"] = DNI;
        data["Nombre"] = Nombre;
        data["Apellido"] = Apellido;

        var response = wb.UploadValues("http://localhost:3000/api/exec/register", "POST", data);

        string responseInString = Encoding.UTF8.GetString(response);
        Hide();

        MessageBox.Show("Registro Guardado " + responseInString);
    }
}

```

**Figura 68. Código del registro estudiantes
Tomada del código de programación en C#**

5.5.11. Formulario reconocimiento facial

En la figura 69, se realiza la construcción de la interfaz de reconocimiento facial.



**Figura 69. Interfaz de reconocimiento facial
Tomada del código de programación en C#**

```

private void btnRegistrarUsuario_Click(object sender, EventArgs e)
{
    string Nombre = txtNombreEstudiante.Text;
    string Apellido = txtApellidoEstudiante.Text;

    // HttpClient client = new HttpClient();
    using (var wb = new WebClient())
    {
        var data = new NameValueCollection();
        data["Nombre"] = Nombre;
        data["Apellido"] = Apellido;

        var response = wb.UploadValues("http://localhost:3000/api/exec/recognize", "POST", data);
        string responseInString = Encoding.UTF8.GetString(response);

        MessageBox.Show(responseInString);
    }
}

```

Figura 70. Código de reconocimiento facial en C#
Tomada del código de programación en C#

```

1  app.post('/api/exec/recognize', (req, res) => {
2    const {
3      Nombre,
4      Apellido
5    } = req.body;
6
7    res.send({
8      msg: 'El reconocimiento empezara en segundos'
9    })
10
11   const programPython = executeProgramPython('recognize', [Nombre, Apellido])
12   programPython.stdout.on('data', (data) => {
13     console.log(data.toString())
14   })
15   programPython.on('close', (code) => {
16     console.log( `python program exit with code ${code}` )
17   })
18 })
19
20 app.listen(port, () => {
21   console.log(`server run on port ${port}`)
22 })
23

```

Figura 71. Código de reconocimiento facial en JavaScript
Tomada del código de programación en C#

En la figura 72, se evidencia las carpetas creadas con el nombre de cada estudiante.



Figura 72. Creación del DataSet
Figura tomada del escritorio de la elaboración del Dataset

5.5.12. Diagrama de clases

Se describen cada una de las clases consideradas en el diagrama respecto a: nombre, descripción, atributos, métodos y relaciones. Como se muestra en la figura 73.

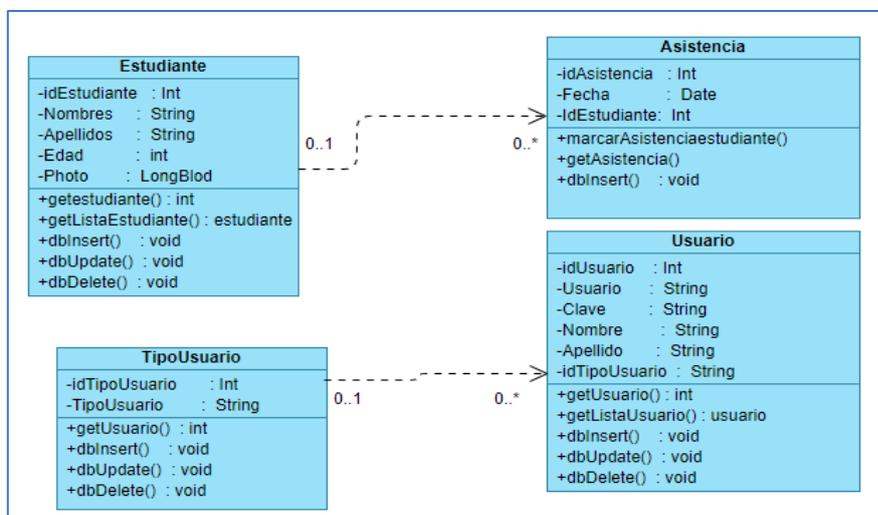


Figura 73. Diagrama de clases

Tabla 49. Clase tipo de usuario

Nombre	Tipo Usuario
Descripción	En esta clase se describe la información de los usuarios que tendrán acceso al sistema.
Atributos	los atributos son: IdUsuario, Nombre, Clave
métodos	Se usan métodos de inserción, actualización y eliminación de registros en la base de datos.

Tabla 50. Clase Usuario

Nombre	Usuario
Descripción	En esta clase se describe la información de los usuarios que tendrán acceso al sistema.
Atributos	los atributos son: IdUsuario, Nombre, Clave
métodos	Se usan métodos de inserción, actualización y eliminación de registros en la base de datos.

Tabla 51. Clase estudiante

Nombre de la clase:	Estudiante
Descripción	En esta clase se describe la información de los estudiantes
Atributos	atributos son: id, nombre, apellidos, edad, género
métodos	insertar, actualizar, eliminar

Tabla 52. Clase asistencia

Nombre de la clase:	Asistencia
Descripción	En esta clase se permite recoger información relativa de la asistencia del estudiante al momento de realizar el reconocimiento facial.
Atributos	atributos: idAsistencia, DniEstudiante, FechaAsistencia
métodos	Registrar Asistencia, Mostrar Asistencia

5.5.13. Construcción de la base de datos

La construcción de la base de datos se desarrolló en el sistema de gestión de base de datos *MySQL workbench*.

En la siguiente imagen se realiza la creación de la base de datos con el nombre *dbreconocimientofacial*, para el funcionamiento respectivo de nuestro sistema de escritorio de reconocimiento facial.

```
1 #CREACIÓN DE LA BASE DE DATOS
2 ● CREATE DATABASE dbreconocimientofacial;
3 ● USE dbreconocimientofacial;
4
```

En la siguiente tabla 53, se crea la tabla el tipo de usuario detalladamente con la siguiente información: el campo, el tamaño, el tipo de datos y la descripción de cada campo para el inicio de sesión del sistema.

Tabla 53. Descripción de la tabla Tipo de Usuario

Nombre de la tabla:		tblTipoUsuario	
Campo	Tamaño	Tipo de dato	Descripción
idTipoUsuario		Int	Clave única de tipo de usuario
tipoUsuario	100	Varchar	Nombre del tipo de usuario

```

5  #CREACIÓN DE LAS TABLAS
6  ● CREATE TABLE tblTipoUsuario
7  (
8      idTipoUsuario INT PRIMARY KEY auto_increment,
9      tipoUsuario VARCHAR(100) NOT NULL
10 );
11

```

En la tabla 54, se crea la tabla de usuario detalladamente con la siguiente información: el campo, el tamaño, el tipo de datos y la descripción de cada campo para el inicio de sesión del sistema.

Tabla 54. Descripción de la tabla Usuario

Nombre de la tabla:		tblUsuario	
Campo	Tamaño	Tipo de dato	Descripción
idUsuario		Int	Clave única del usuario
Usuario	50	Varchar	Nombre del tipo de usuario
Password	80	Varchar	Contraseña del usuario
Nombre	100	Varchar	Nombre del usuario
apellidoUsuario	100	Varchar	Apellido del usuario

RELACION: idTipoUsuario con idTipoUsuario

```

12 ● CREATE TABLE tblUsuario
13 (
14     idUsuario INT AUTO_INCREMENT PRIMARY KEY ,
15     usuario VARCHAR(50) NOT NULL,
16     password VARCHAR(80) NOT NULL,
17     nombre VARCHAR(100) NOT NULL,
18     apellidoUsuario VARCHAR(100) NOT NULL,
19     idTipoUsuario INT,
20     FOREIGN KEY(idTipoUsuario)
21     REFERENCES tblTipoUsuario(idTipoUsuario)
22 );

```

En la siguiente tabla 55, se crea la tabla estudiante detalladamente con la siguiente información: el campo, tamaño, el tipo de datos y la descripción de

cada campo para el registro de nuevos estudiantes de nuestro sistema para posterior usar esos datos para el reconocimiento facial.

Tabla 55: Descripción de la tabla Estudiante

Nombre de la tabla:		tblEstudiante	
Campo	Tamaño	Tipo de dato	Descripción
idEstudiante		Int	Clave única de tipo de usuario
DNI	8	Varchar	DNI del estudiante
Nombre	50	Varchar	Nombre del estudiante
Apellido	50	Varchar	Apellido del estudiante
Edad		Int	Edad del estudiante
Photo	+4 bytes	Longblob	Imagen del estudiante

```

5 • ○ CREATE TABLE tblestudiante (
6     idEstudiante int NOT NULL AUTO_INCREMENT,
7     DNI varchar(8) DEFAULT NULL,
8     Nombre varchar(50) DEFAULT NULL,
9     Apellido varchar(50) DEFAULT NULL,
10    Edad int DEFAULT NULL,
11    Photo LONGBLOB,
12    PRIMARY KEY (idEstudiante)
13 )

```

En la tabla 56, se crea la tabla asistencia del estudiante detalladamente con la siguiente información: el campo, tamaño, el tipo de datos y la descripción de cada campo para el registro de nuevos estudiantes de nuestro sistema con el resultado del reconocimiento facial.

Tabla 56. Descripción de la tabla Asistencia-Estudiante

Nombre de la tabla:		tblAsistenciaEstudiante	
Campo	Tamaño	Tipo de dato	Descripción
idEstudiante		Int (fore	Clave del estudiante es por defecto
idEstudianteAsistencia		Int (primary key)	Clave del estudiante para la asistencia.
FechaAsistencia		DateTime	Fecha de asistencia
Relación:	idEstudiante	con idEstudiante	de la tabla tblEstudiante

```
33 • ○ CREATE TABLE tblAsistenciaEstudiante (  
34     IdEstudiante int DEFAULT NULL,  
35     idEstudianteAsistencia int NOT NULL AUTO_INCREMENT,  
36     FechaAsistencia datetime DEFAULT NULL,  
37     PRIMARY KEY (idEstudianteAsistencia),  
38     KEY IdEstudiante (IdEstudiante),  
39     CONSTRAINT tblAsistenciaEstudiante_ibfk_1  
40     FOREIGN KEY (IdEstudiante)  
41     REFERENCES tblEstudiante (idEstudiante)  
42 );
```

Una vez concluida la implementación, se realizan las respectivas pruebas y evaluación de los módulos del sistema.

CAPÍTULO VI

PRUEBAS Y RESULTADOS

6.1. Pruebas y resultados

Las pruebas nos ayudan a comprobar que el sistema cumpla con las necesidades establecidas por el usuario, cumpliendo las garantías de calidad, Las condiciones de pruebas deben incluir tanto situaciones normales como anormales. El objetivo de las pruebas es poder encontrar un error para decidir qué hacer en el momento que ocurre algo que no debe pasar y viceversa, es decir, un sistema de pruebas está diseñado para detectar errores en los sistemas de información. Es por ello que en este proyecto se plantea pruebas para el modelo de reconocimiento facial y pruebas de aceptación del sistema.

6.1.1. Pruebas del modelo de reconocimiento facial

Para entrenar el modelo de reconocimiento facial se hicieron las pruebas con 05 personas en los siguientes escenarios: sin lentes-con mascarilla, con lentes-sin mascarilla, sin lentes - sin mascarilla. Estas pruebas se clasifican por tipo para analizar el porcentaje de reconocimiento de cada una de ellas, de manera que a través de la aplicación del *Deep Learning* que cumpla un porcentaje de precisión aceptable al reconocer e identificar el rostro de los estudiantes incluyendo en estos casos las restricciones que sean causadas por la presencia de mascarillas o lentes que reduce los rasgos faciales para un mejor reconocimiento. Estas pruebas se realizaron con *Postman*.

Postman es una aplicación que nos permite realizar pruebas API, es un cliente HTTP que nos da la posibilidad de testear “*HTTP requests*” a través de una interfaz gráfica de usuario, por medio de la cual obtendremos diferentes tipos de respuesta que posteriormente deberán validarse. (52)

6.2. Pruebas y resultados del modelo de reconocimiento facial

En esta sección, se muestran la ejecución y resultados de las pruebas realizadas para evaluar el rendimiento del modelo de reconocimiento facial que se ha implementado en este proyecto. Cada una de estas pruebas utilizó distintos escenarios, base de datos con características diferentes para verificar el funcionamiento del algoritmo frente a diferentes condiciones. El reconocimiento de cada muestra se observa a continuación de acuerdo con cada escenario planteado.

6.2.1. Prueba 01: Gladys Samaniego

Escenario 01: con mascarilla - sin lentes
Descripción: El primer caso de prueba es en el escenario de “con mascarilla - sin lentes”, el estudiante este situado de manera frontal con una mascarilla, además no hay cambio del entorno de prueba.
Persona: Gladys Samaniego
Porcentaje de Acierto: 0.67 → 67 %

3.69019508e-02	1.30873201e-02	1.28727744e-03	8.58184975e-03
2.12400779e-02	1.93091610e-03	7.72366440e-03	1.65200606e-02
4.29092470e-04	2.14546235e-04	0. 0.	2.57455488e-03
4.29092470e-04	2.57455488e-03	6.22184062e-03	9.65458062e-03
4.07637842e-03	0. 6.43638719e-04	2.78910110e-03	1.07273122e-03
8.58184940e-04	2.36000866e-03	6.00729464e-03	1.07273122e-03
2.14546235e-04	8.58184940e-04	1.69491526e-02	1.50182366e-03
4.50547086e-03	9.44003463e-03	6.43638719e-04	0. 0. 0.
6.43638719e-04	4.29092470e-04	0. 6.43638719e-04	0. 0. 0. 0.

Escenario 02: Sin mascarilla - sin lentes

Descripción: En este caso de prueba es en el escenario de “Sin mascarilla - sin lentes”, el estudiante está situado de manera frontal sin el uso de mascarilla ni lentes, además no hay cambio del entorno de prueba. Por ende, el modelo logra identificar al estudiante mostrando su nombre completo.

Persona: Gladys Samaniego

Porcentaje de Acierto: 0.91 → 89 %

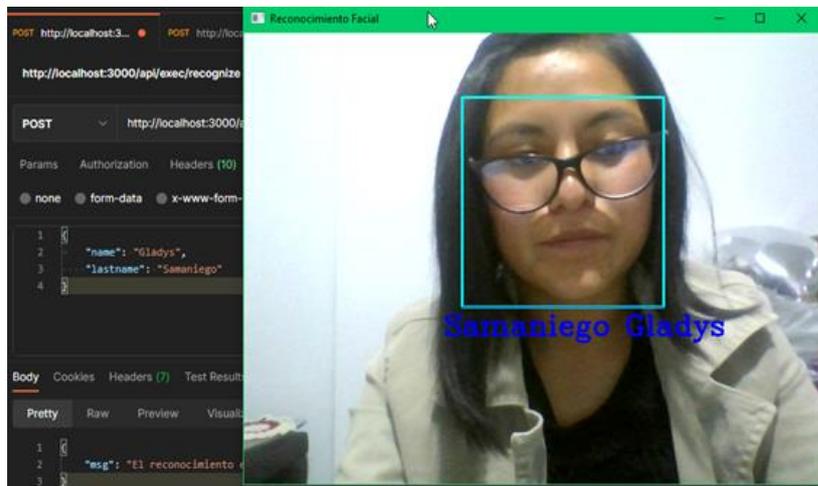
```
3.69019508e-02 1.30873201e-02 1.28727744e-03 8.58184975e-03
2.12400779e-02 1.93091610e-03 7.72366440e-03 1.65200606e-02
4.29092470e-04 2.14546235e-04 0. 0. 2.57455488e-03
4.29092470e-04 2.57455488e-03 6.22184062e-03 9.65458062e-03
4.07637842e-03 0. 6.43638719e-04 2.78910110e-03 1.07273122e-03
8.58184940e-04 2.36000866e-03 6.00729464e-03 1.07273122e-03
2.14546235e-04 8.58184940e-04 1.69491526e-02 1.50182366e-03
4.50547086e-03 9.44003463e-03 6.43638719e-04 0. 0. 0.
6.43638719e-04 4.29092470e-04 0. 6.43638719e-04 0. 0. 0. 0.
```

Escenario 03: Con lentes- sin mascarilla

Descripción: En este caso de prueba se realiza en el escenario de “Con lentes- sin mascarilla”, es estudiante está situado de forma frontal haciendo uso de lentes, en este escenario también se obtienen un resultado aceptable ya que reconoce a la persona.

Persona: Gladys Samaniego

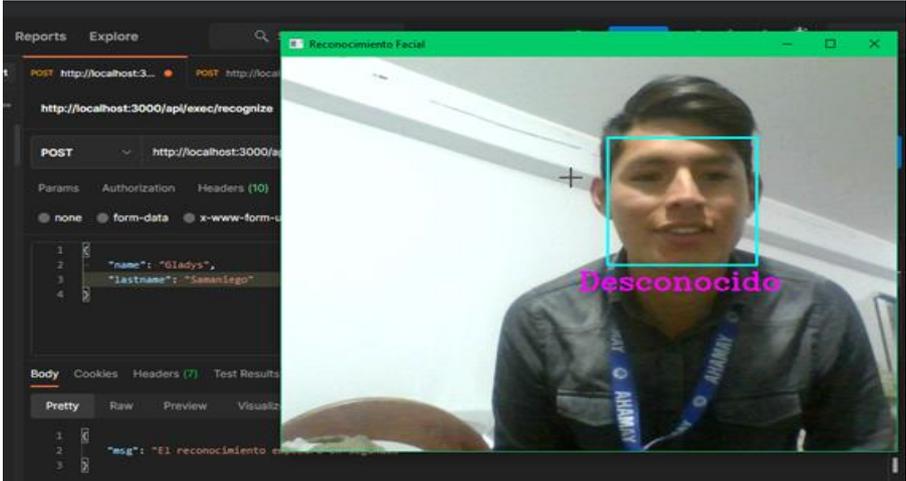
Porcentaje de Acierto: 0.88 → 78%



```
3.69019508e-02 1.30873201e-02 1.28727744e-03 8.58184975e-03
2.12400779e-02 1.93091610e-03 7.72366440e-03 1.65200606e-02
4.29092470e-04 2.14546235e-04 0. 0. 2.57455488e-03
4.29092470e-04 2.57455488e-03 6.22184062e-03 9.65458062e-03
4.07637842e-03 0. 6.43638719e-04 2.78910110e-03 1.07273122e-03
```

Escenario 04: Otra Persona - Desconocido

Descripción: Para este escenario se realizo las pruebas con otro estudiante que no pertenece al salon, en este escenario seria el caso de que se presente una suplantación de identidad. El resultado que nos muestra es un mensaje de desconocido, con los siguientes valores:



The screenshot displays a facial recognition interface. On the left, a web client shows a POST request to `http://localhost:3000/ap/recognize` with a body containing `"name": "Gladys", "lastname": "Samaniego"`. The main window, titled "Reconocimiento Facial", shows a video feed of a man with a blue bounding box around his face. The word "Desconocido" is written in pink text over the face. Below the video feed, a black box contains a list of numerical values representing confidence scores for different faces.

```
1.50182366e-03 6.65093306e-03 4.29092470e-04 1.07273122e-03  
2.57455488e-03 4.29092470e-04 2.14546235e-04 0. 0.  
1.07273122e-03 2.14546235e-04 1.07273122e-03 8.58184940e-04  
1.07273122e-03 2.14546235e-04 0. 0. 6.43638719e-04 0. 0. 0.  
1.71636988e-03 0. 0. 0. 1.07273122e-03 2.14546235e-04  
1.28727744e-03 8.58184940e-04 3.00364732e-03 4.29092470e-04  
2.14546235e-04 4.29092470e-04 1.28727744e-03 0. 4.29092470e-04
```

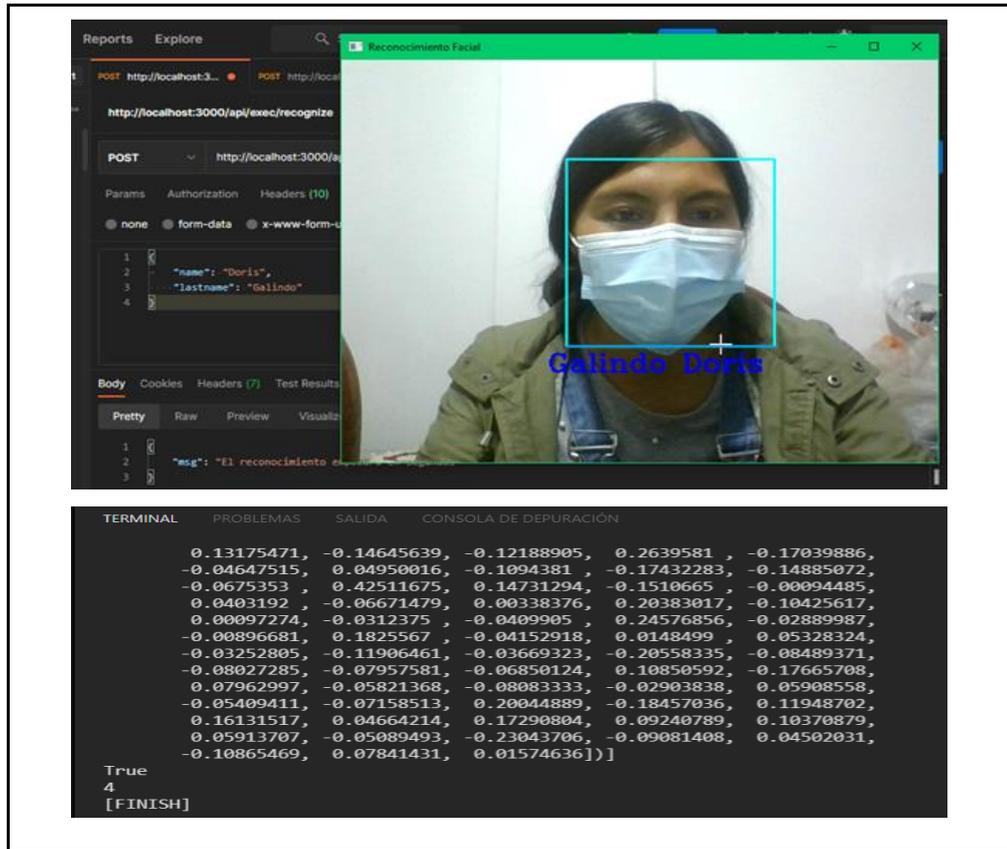
6.2.2. Prueba 02: Doris Galindo

Escenario 01: con mascarilla - sin lentes

Descripción: Este caso de prueba es en el escenario de “con mascarilla - sin lentes”, con un distinto estudiante este situado de manera frontal con una mascarilla, además no hay cambio del entorno de prueba

Persona: Doris Galindo

Porcentaje de Acierto: 0.65 → 65 %

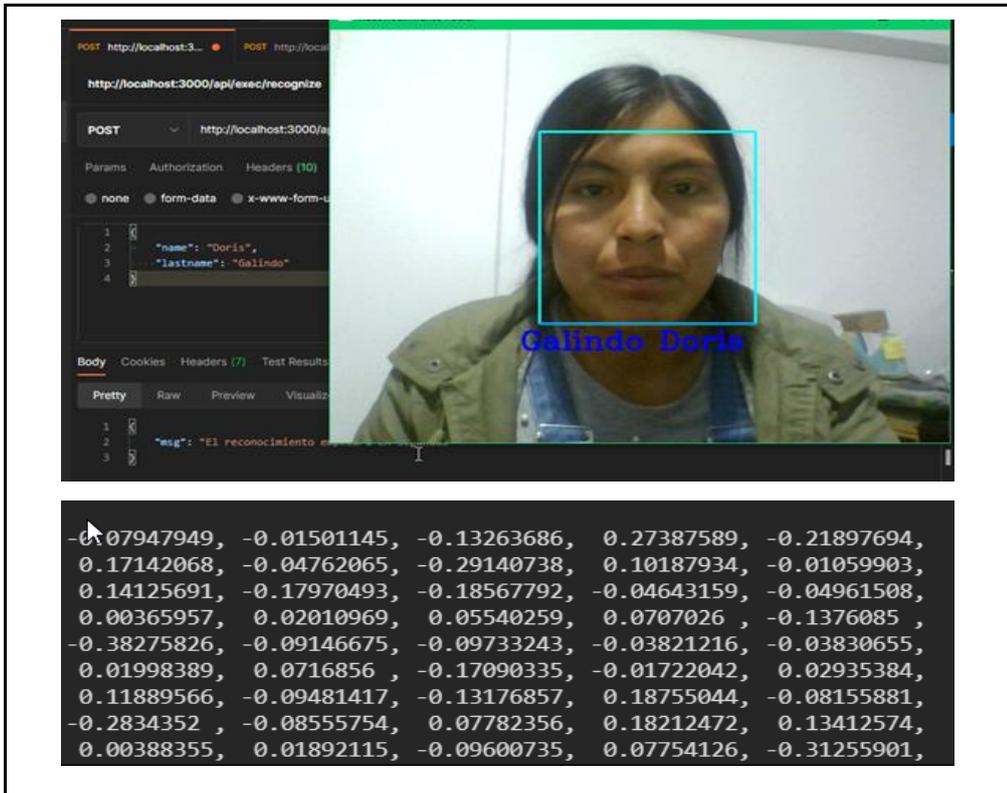


Escenario 02: sin mascarilla - sin lentes

Descripción: Esta prueba se hace con un segundo estudiante, en el escenario “sin mascarilla - sin lentes”, el estudiante se encuentra de forma frontal

Persona: Doris Galindo

Porcentaje de Acierto: 0.88 → 88 %

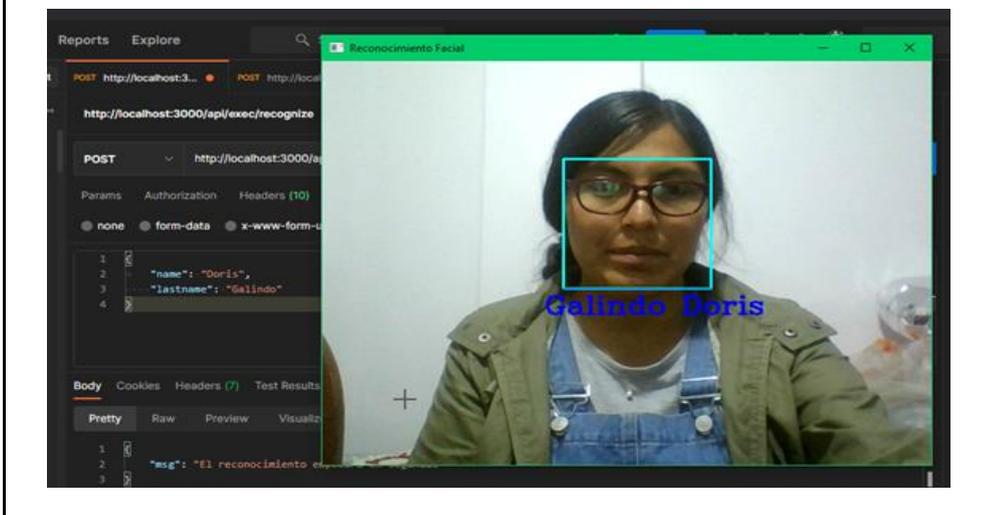


Escenario 03: Con lentes- sin mascarilla

Descripción: Se evidencia las pruebas de reconocimiento facial de la estudiante Doris Galindo donde tiene puesto lentes.

Persona: Doris Galindo

Porcentaje de Acierto: 0.86 → 82 %



```
1.50182366e-03 6.65093306e-03 4.29092470e-04 1.07273122e-03
2.57455488e-03 4.29092470e-04 2.14546235e-04 0. 0.
1.07273122e-03 2.14546235e-04 1.07273122e-03 8.58184940e-04
1.07273122e-03 2.14546235e-04 0. 0. 6.43638719e-04 0. 0. 0.
1.71636988e-03 0. 0. 0. 1.07273122e-03 2.14546235e-04
1.28727744e-03 8.58184940e-04 3.00364732e-03 4.29092470e-04
```

Escenario 04: Otra persona - Desconocido

Descripción: En este escenario la estudiante Gladys Samaniego trata de suplantar la identidad de la estudiante Doris, sin embargo el modelo muestra como desconocido.

Escenario 04: Otra persona - Desconocido



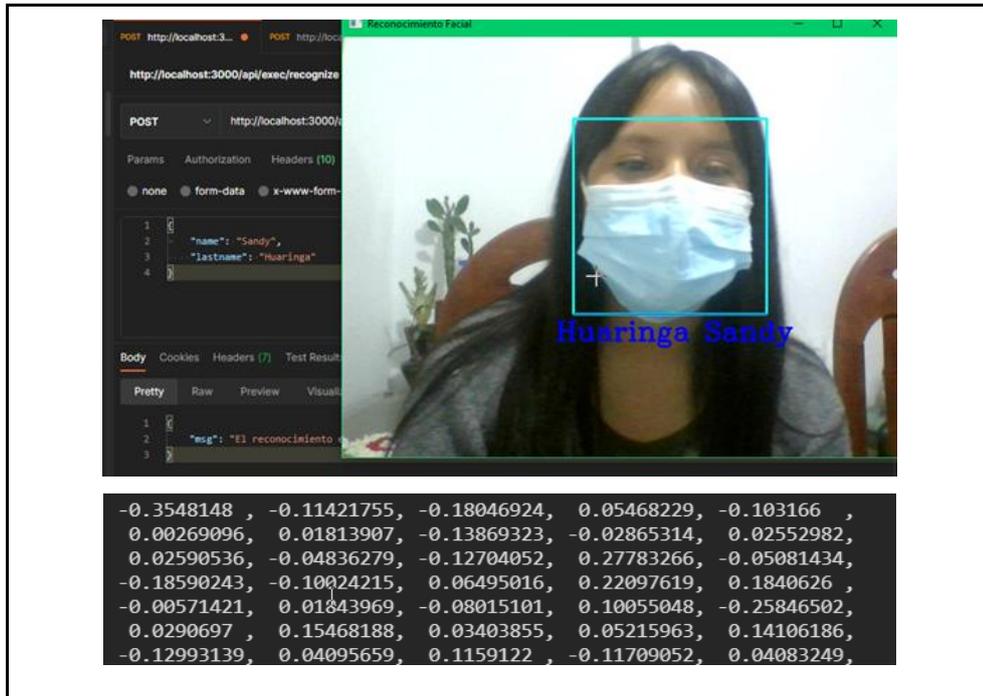
6.2.3. Prueba 03: Sandy Huaringa

Escenario 01: con mascarilla - sin lentes

Descripción: Se puede evidenciar a la estudiante Sandy Huaringa puesto una mascarilla por el contexto en el que nos encontramos. El algoritmo logra reconocer a la persona mostrando el nombre completo

Persona: Sandy Huaringa

Porcentaje de Acierto: 0.68 → 68 %

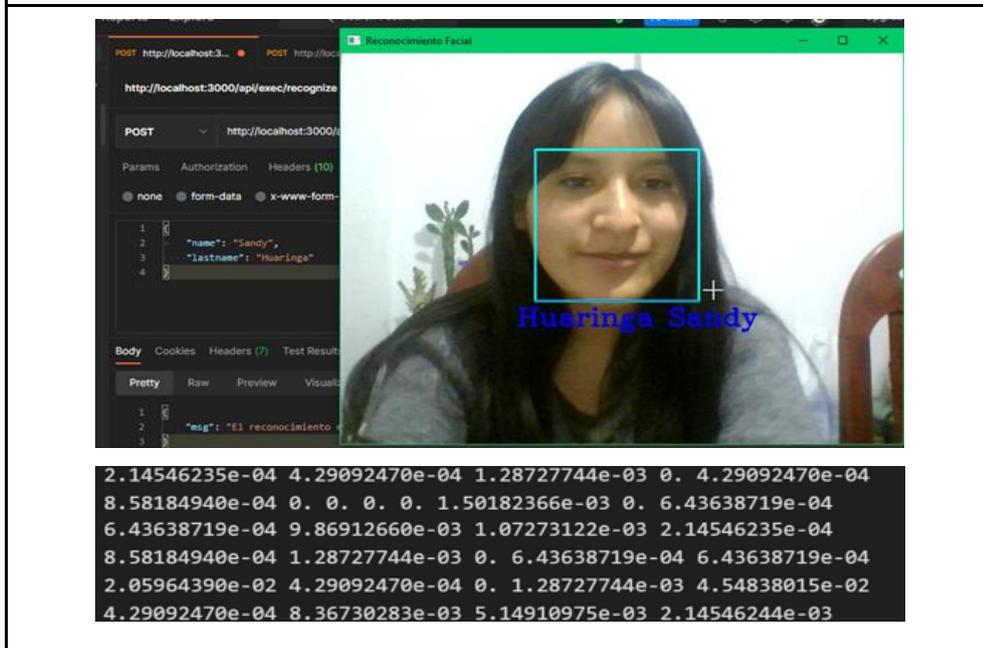


Escenario 02: sin mascarilla - sin lentes

Descripción: Para este escenario el estudiante se encuentra de manera frontal sin lentes ni mascarilla.

Persona: Sandy Huaringa

Porcentaje de acierto: 0.90 → 90 %

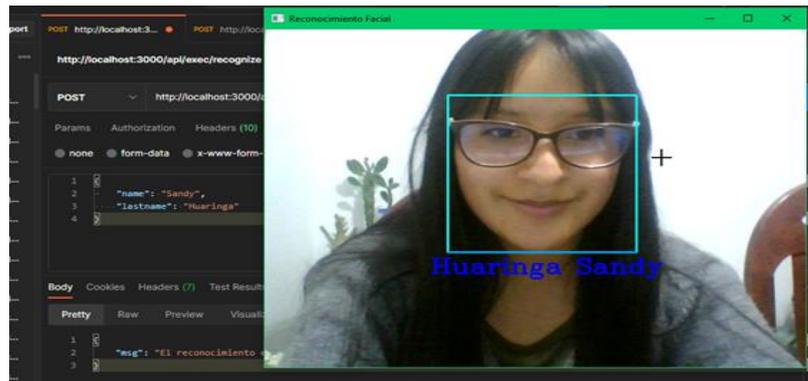


Escenario 03: Con lentes- sin mascarilla

Descripción: Para este escenario el estudiante se encuentra de manera frontal y puesto lentes, para este caso tambien el modelo logra realizar el reconocimiento.

Persona: Sandy Huaringa

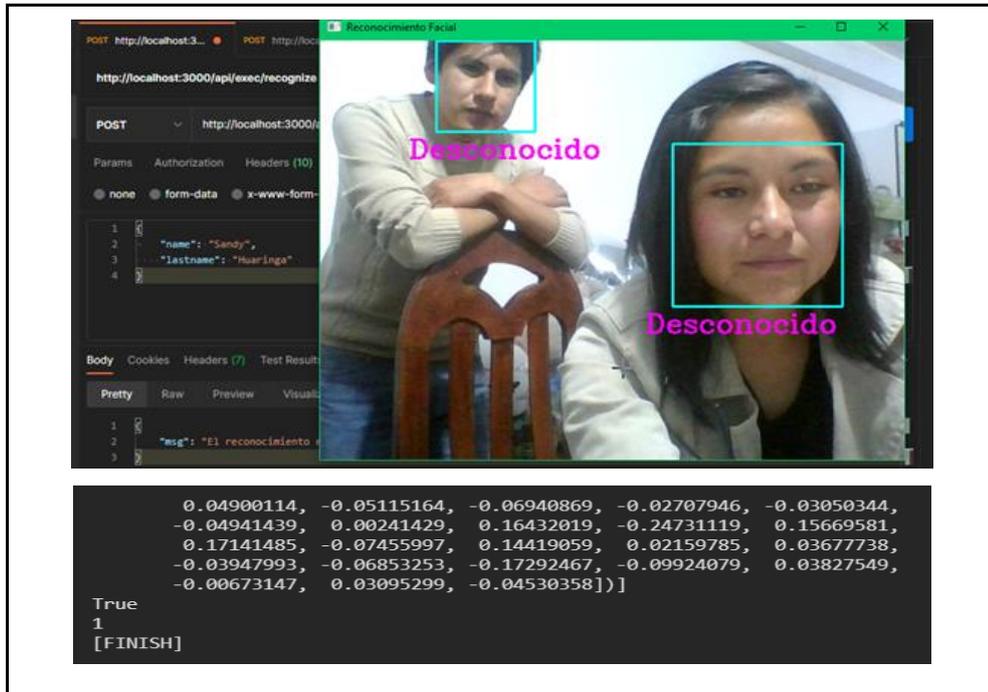
Porcentaje de acierto: 0.76 → 76%



```
3.69019508e-02 1.30873201e-02 1.28727744e-03 8.58184975e-03
2.12400779e-02 1.93091610e-03 7.72366440e-03 1.65200606e-02
4.29092470e-04 2.14546235e-04 0. 0. 2.57455488e-03
4.29092470e-04 2.57455488e-03 6.22184062e-03 9.65458062e-03
4.07637842e-03 0. 6.43638719e-04 2.78910110e-03 1.07273122e-03
8.58184940e-04 2.36000866e-03 6.00729464e-03 1.07273122e-03
2.14546235e-04 8.58184940e-04 1.69491526e-02 1.50182366e-03
```

Escenario 04: Otras personas - Desconocidos

Descripción: Para este escenario se realizo la prueba con dos estudiantes al mismo tiempo y no se reconoce a ninguno de los dos.



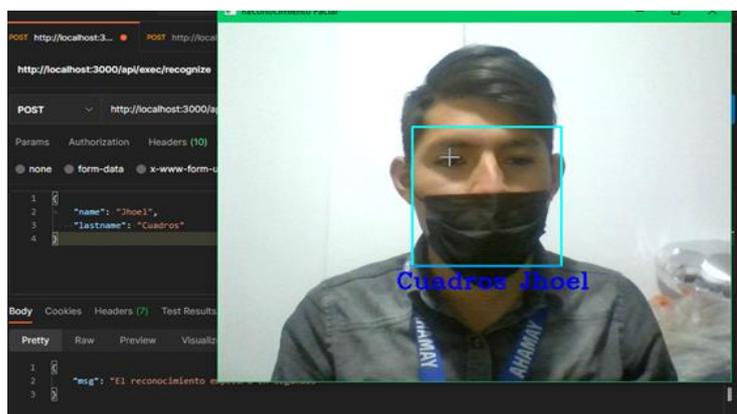
6.2.4. Prueba 04: Joel Cuadros

Escenario 01: con mascarilla - sin lentes

Descripción: Esta prueba se realizo a un estudiante con una mascarilla, impidiendo reconocer las caracteristicas de su rostro, sin embargo el modelo logra realizar el reconocimiento

Persona: Joel Cuadros

Porcentaje de Acierto: 0.72 → 72 %



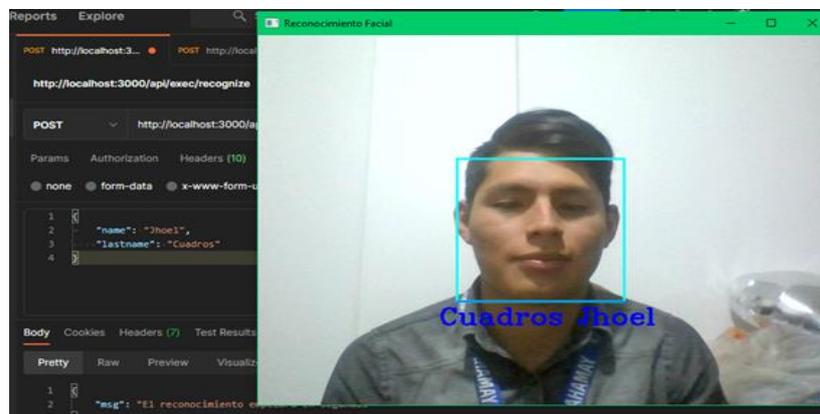
```
2.14546235e-04 6.43638719e-04 0. 2.14546235e-04 0.
4.29092470e-04 0. 8.58184940e-04 2.14546235e-04 6.43638719e-04
0. 1.71636988e-03 4.29092470e-04 8.58184940e-04 0.
1.39455050e-02 6.43638708e-03 2.57455483e-02 8.58184940e-04
7.01566190e-02 1.93091610e-03 4.50547086e-03 2.14546244e-03
6.80111572e-02 0. 2.14546235e-04 0. 4.29092470e-04
6.43638719e-04 2.14546235e-04 8.58184940e-04 1.54473288e-02
2.14546235e-04 8.58184940e-04 0. 2.14546244e-03 0.
```

Escenario 02: sin mascarilla - sin lentes

Descripcion: Se realiza el reconocimiento con la misma persona, en este escenario sin la presencia de la mascarilla.

Persona: Joel Cuadros

Porcentaje de acierto: 0.88 → 88 %



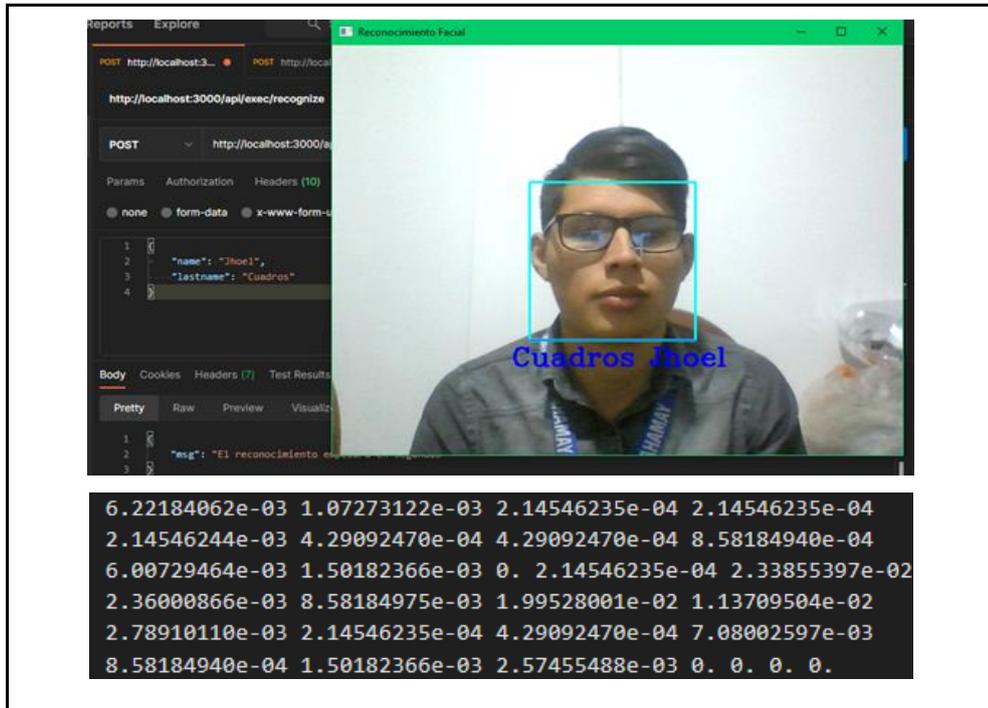
```
4.29092470e-04 2.14546235e-04 0. 0. 2.57455488e-03
4.29092470e-04 2.57455488e-03 6.22184062e-03 9.65458062e-03
4.07637842e-03 0. 6.43638719e-04 2.78910110e-03 1.07273122e-03
8.58184940e-04 2.36000866e-03 6.00729464e-03 1.07273122e-03
2.14546235e-04 8.58184940e-04 1.69491526e-02 1.50182366e-03
4.50547086e-03 9.44003463e-03 6.43638719e-04 0. 0. 0.
6.43638719e-04 4.29092470e-04 0. 6.43638719e-04 0. 0. 0. 0.
```

Escenario 03: Con lentes- sin mascarilla

Descripcion: Para este caso de prueba el estudiante se encuentra de maera forntla con la presencia de lentes.

Persona: Joel Cuadros

Porcentaje de acierto: 0.77 → 77 %



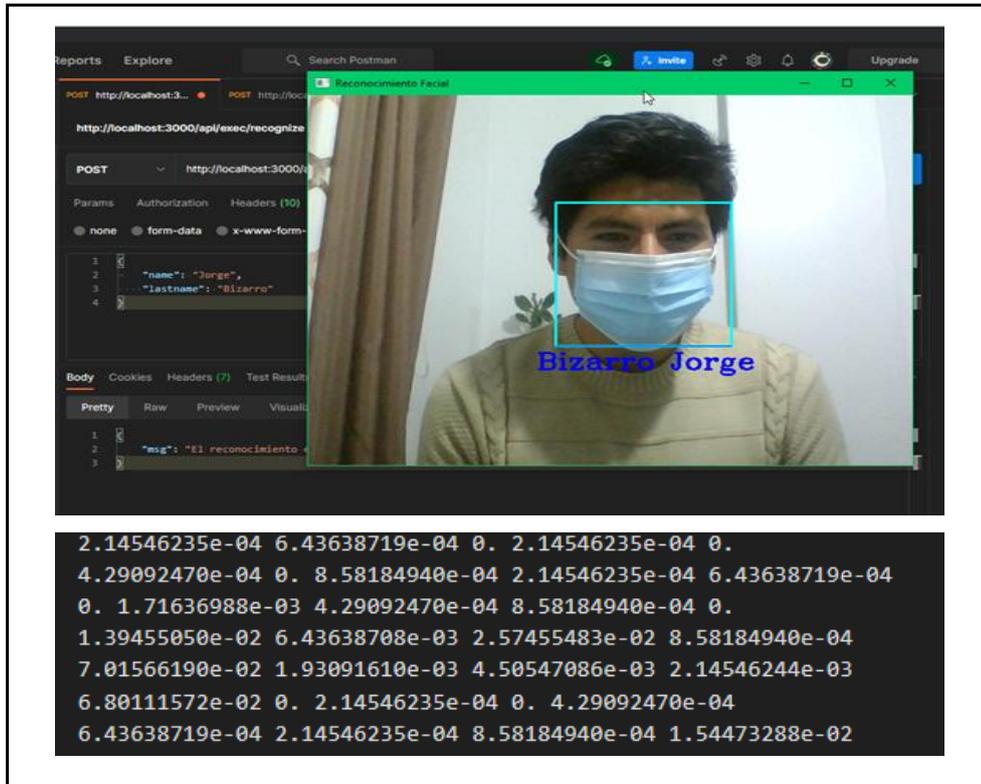
6.2.5. Prueba 05: Jorge Bizarro

Escenario 01: con mascarilla - sin lentes

Descripcion: Se puede evidenciar el estudiante Jorge Bizarro se encuentra puesto una mascarilla por el contexto en el que nos encontramos. El algoritmo logra reconocer a la persona mostrando el nombre completo en la pantalla.

Persona: Jorge Bizarro

Porcentaje de Acierto: 0.64 → 64 %



Escenario 02: sin mascarilla - sin lentes

Descripcion: Se realiza el reconocimiento con el estudiante Jorge, en este escenario sin la presencia de la mascarilla. Y el modelo obtiene un mayor porcentaje de acierto.

Persona: Jorge Bizarro

Porcentaje de acierto: 0.89 → 89 %



```

6.22184062e-03 1.07273122e-03 2.14546235e-04 2.14546235e-04
2.14546244e-03 4.29092470e-04 4.29092470e-04 8.58184940e-04
6.00729464e-03 1.50182366e-03 0. 2.14546235e-04 2.33855397e-02
2.36000866e-03 8.58184975e-03 1.99528001e-02 1.13709504e-02
2.78910110e-03 2.14546235e-04 4.29092470e-04 7.08002597e-03
8.58184940e-04 1.50182366e-03 2.57455488e-03 0. 0. 0. 0.

```

Escenario 03: Con lentes- sin mascarilla

Descripcion: Para este caso se evidencia al estudiante de forma frontal puestos lentes y cerrados los ojos, al mismo tiempo no esta en el centro de la camara.

Persona: Jorge Bizarro

Porcentaje de acierto: 0.75 → 75%



```

2.14546244e-03 4.29092470e-04 4.29092470e-04 8.58184940e-04
6.00729464e-03 1.50182366e-03 0. 2.14546235e-04 2.33855397e-02
2.36000866e-03 8.58184975e-03 1.99528001e-02 1.13709504e-02
2.78910110e-03 2.14546235e-04 4.29092470e-04 7.08002597e-03
8.58184940e-04 1.50182366e-03 2.57455488e-03 0. 0. 0. 0.

```

6.2.6. Resultados

Como resultado de las 05 pruebas que se usaron como muestras en diferentes escenarios, se obtuvo una identificación correcta según la

clasificación de estos casos como se muestra en la tabla 57, y al realizarse pruebas con personas no registradas se identifican como desconocidas.

Tabla 57. Resultados de las pruebas

Pruebas	Escenarios		Escenario 01		Escenario 0		Escenario 03	
			% con mascarilla	% con lentes	% con lentes	% sin oclusión	% con lentes	% sin oclusión
Prueba 01: Gladys Samaniego	0,67	67%	0,78	78%	0,89	89%		
Prueba 02: Doris Galindo	0,65	65%	0,82	82%	0,88	88%		
Prueba 03: Sandy Huaranga	0,68	68%	0,76	76%	0,90	90%		
Prueba 04: Joel Cuadros	0,72	72%	0,77	77%	0,88	88%		
Prueba 05: Jorge Bizarro	0,64	64%	0,75	75%	0,89	89%		

6.2.7. Pruebas de software

Las pruebas de software es un proceso de ejecución de un programa bajo condiciones de control y evaluación, registrando los resultados con la intención de descubrir un error. (53) Las pruebas pueden hacerse a diferentes niveles dependiendo del objetivo, en este caso para medir la aceptación del software se realizarán pruebas de aceptación.

6.2.8. Pruebas de aceptación

Para evaluar el *software* se realizaron pruebas de aceptación que están dirigidas a validar que el sistema cumple con los requisitos de funcionamiento esperado y satisface las necesidades del usuario.

Las pruebas de aceptación también involucran pruebas de desempeño, de stress, de seguridad, pruebas para los datos persistentes, haciendo en este caso pruebas de frecuencia de uso, restricciones de acceso, restricciones de integridad, requisitos de guarda y retención de datos, entre otras. A este nivel de pruebas se rectifican atributos de calidad como confiabilidad, usabilidad, portabilidad, etc. A continuación se describe un flujo de trabajo para la realización de pruebas de aceptación del cliente para un software de gestión dirigido y diseñado por Una empresa certificadora de calidad.

6.2.9. Plan de pruebas de aceptación

Este documento nos sirve para definir las pruebas del sistema, también nos ayuda a efectuar una trazabilidad entre los requisitos y el sistema, validando de esta forma las necesidades del usuario.

Tabla 58. Plan de pruebas de aceptación

ID	Área Funcional	Descripción	Resultado Esperado	ESTADO
PA01	Inicio de sesión	Inicio de sesión con usuario y contraseña.	Acceso a los módulos del sistema	Ejecutado
PA02	Registro de estudiante	El docente puede registrar a un nuevo estudiante con sus datos	El administrador y docente visualiza la lista de registros de estudiantes	Ejecutado
PA03	Captura de rostro	Haciendo uso de la cámara web se realiza la captura de rostro	Se guarda el rostro capturado en la base de datos.	Ejecutado
PA04	Captura de rostro de un estudiante con lentes	Haciendo uso de la cámara web se realiza la captura de rostro con más con lentes	Se muestra la imagen registrada.	Ejecutado
PA05	Captura de rostro de un estudiante con mascarilla	Haciendo uso de la cámara web se realiza la captura de rostro con mascarilla.	Se muestra la imagen registrada.	Ejecutado
PA06	Reconocimiento Facial	El reconocimiento facial se realiza en tiempo real activando la cámara web.	Muestra los datos del estudiante, caso contrario muestra un mensaje de desconocido.	Ejecutado
PA07	Registro de asistencia	Mediante el reconocimiento facial se registra la asistencia	Se registra la asistencia de los estudiantes	Ejecutado

6.2.10. Ejecución de resultados

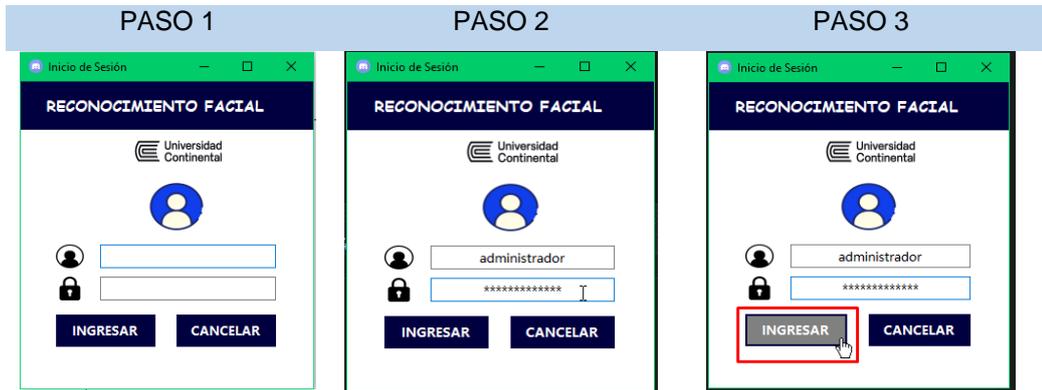
Una vez concluida la implementación se pasa a realizar la ejecución del plan de pruebas de la tabla 59. Para probar cada módulo del sistema, se identifica el caso de prueba, un responsable de la ejecución, las precondiciones, el propósito y el escenario en el que se ejecutarán las pruebas.

Tabla 59: Ejecución del plan de pruebas

Caso de Prueba	PA01 - Inicio de sesión
Responsable	Sandy Huaringa
Precondiciones	<ul style="list-style-type: none">• El usuario debe tener instalada el software de reconocimiento facial en su PC o laptop en el cual realizara la prueba.• El usuario debe estar registrado en el software.• El usuario debe tener “nombre de usuario” y “contraseña”
Propósito / Descripción	Probar el inicio de sesión en el software de reconocimiento facial de manera exitosa.
Escenario	Inicio de Sesión Exitoso.
Resultado Esperado	Al presionar el botón “INGRESAR” permite el Acceso al menú principal del sistema de reconocimiento facial.

Actividades:

01. El usuario administrador completa los campos de “usuario” y “contraseña”.
02. Presiona el botón ingresar.
03. Ingreso al Menú principal del sistema.



RESULTADO: ACEPTADO



Datos: Usuario: Administrador, Contraseña: Administrador123

Tabla 60. Caso de prueba - Registrar estudiante

Caso de Prueba	PA02 – Registrar Estudiante
Responsable	Sandy Huaranga
Precondiciones	<ul style="list-style-type: none"> • El usuario debe estar logueado en el sistema • El usuario debe tener permiso para crear un nuevo estudiante.
Propósito / Descripción	Probar el registro de estudiante de manera correcta en el sistema.
Escenario	Registro de Estudiante de manera correcta y exitoso
Resultado esperado	El administrador visualiza en la base de datos el nuevo estudiante registrado. El docente visualiza en el sistema la lista de estudiantes con el nuevo estudiante registrado.

Actividades:

PASO 1

PASO 2



RESULTADO:



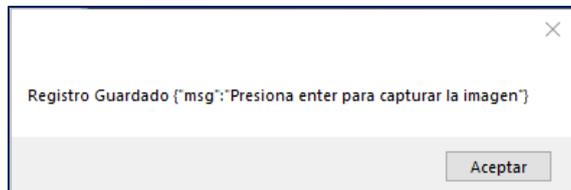
01. El usuario administrador Selecciona la opción de “Estudiantes” en el menú principal.
 02. Elige de la barra de menú la opción “Registrar Estudiante”
 03. Rellena los datos de “DNI”, “Nombres” y “Apellidos”
- Datos de prueba: DNI: 71876465, Nombres: DORIS; Apellido: GALINDO

Tabla 61. Caso de prueba - Capturar rostro

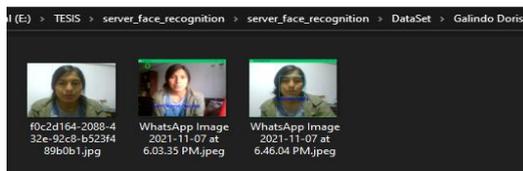
Caso de Prueba	PA03 – Capturar Rostro
Responsable	Sandy Huaringa
Precondiciones	<ul style="list-style-type: none"> • El usuario debe estar logueado en el sistema • El estudiante debe ingresar sus datos • Posición frontal del estudiante
Propósito	Probar la captura de rostro de un estudiante en el sistema.

Escenario	Haciendo uso de la cámara web se realiza la captura de rostro, sin ninguna oclusión.
Resultado esperado	Captura de Imagen de estudiante Doris Galindo. la imagen capturada se almacena en directorio DataSet con el nombre de (Doris Galindo).

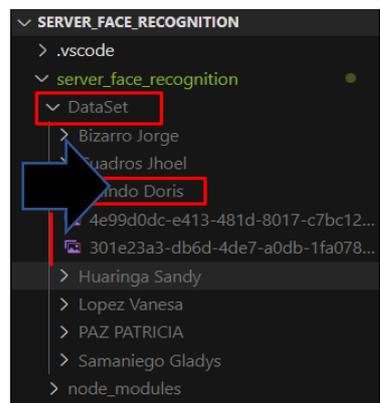
PASO 1 **PASO 2**



RESULTADO: en el directorio se almacenan los rostros capturados



Estructura Del Directorio De Almacenamiento De Rostros.



Actividades:

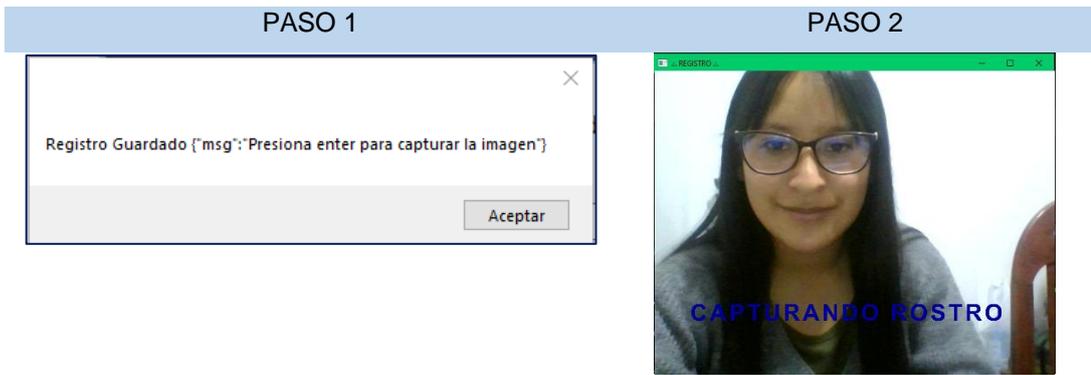
01. El usuario selecciona la opción de capturar rostro.
02. Muestra un mensaje de “**presiona ENTER para captura la imagen**”.
03. Inicio de la cámara con un tiempo de 10 segundos
04. Se realiza 3 capturas mínimas del rostro

Tabla 62. Caso de prueba - Capturar rostro con lentes

Caso de Prueba	PA04 – Capturar Rostro Con lentes
Responsable	Doris Galindo
Precondiciones	<ul style="list-style-type: none">• El usuario debe estar logueado en el sistema• El estudiante debe ingresar sus datos• Posición frontal del estudiante usando lentes
Propósito	Probar la captura de rostro de un estudiante con lentes en el sistema.
Escenario	Haciendo uso de la cámara web se realiza la captura de rostro, con la presencia de lentes.
Resultado esperado	Captura de Imagen de estudiante Sandy Huaranga. La imagen capturada se almacena en el mismo directorio DataSet con el nombre de (Sandy Huaranga).

Actividades:

01. El usuario selecciona la opción de capturar rostro.
02. Muestra un mensaje de “**presiona ENTER para captura la imagen**”.
03. Inicio de la cámara con un tiempo de 10 segundos
04. Se realiza 3 capturas mínimas con la presencia de lentes en el rostro.



RESULTADO: En el directorio creado con nombre y apellido del estudiante, para este caso (Sandy Huaringa) se almacena las imágenes capturadas.

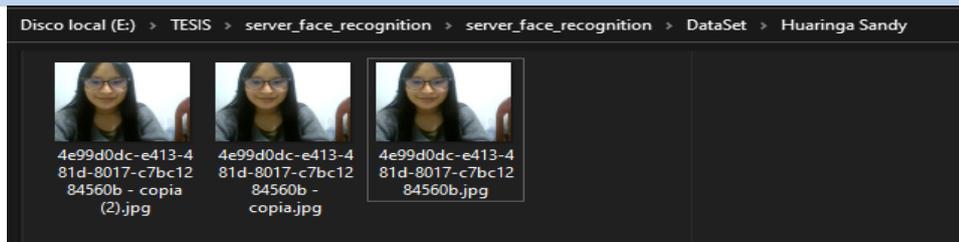


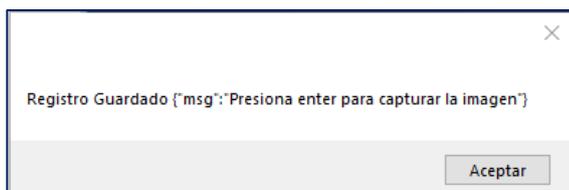
Tabla 63. Caso de prueba - Capturar rostro con lentes

Caso de Prueba	PA05– Captura de Rostro con Mascarilla
Responsable	Doris Galindo
Precondiciones	<ul style="list-style-type: none"> • El usuario debe estar logueado en el sistema • El estudiante debe ingresar sus datos • Posición frontal del estudiante usando mascarilla
Propósito	Probar la captura de rostro de un estudiante con mascarilla en el sistema.
Escenario	Haciendo uso de la cámara web se realiza la captura de rostro, con la presencia de mascarilla.
Resultado esperado	Captura de Imagen de estudiante Sandy Huaringa. La imagen capturada se almacena en el mismo directorio DataSet con el nombre de (Sandy Huaringa).

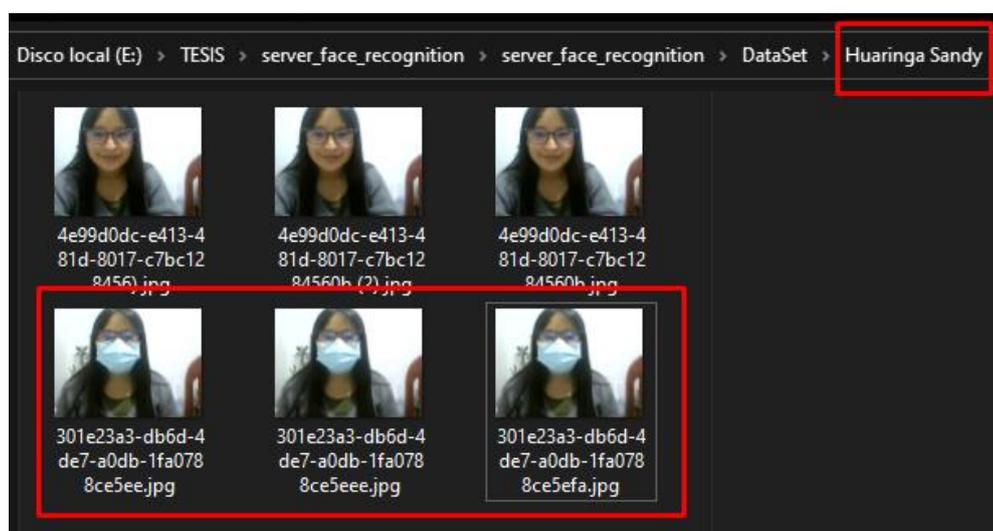
Actividades:

PASO 1

PASO 2



RESULTADO: en el directorio se almacenan los rostros capturados



05. El usuario selecciona la opción de capturar rostro.
06. Muestra un mensaje de “**presiona ENTER para captura la imagen**”.
07. Inicio de la cámara con un tiempo de 10 segundos
08. Se realiza 3 capturas mínimas del rostro

Tabla 64. Caso de prueba realizar reconocimiento

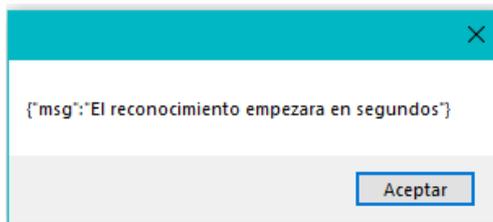
Caso de Prueba	PA06 – Realizar Reconocimiento
Responsable	Sandy Huarínga
Precondiciones	<ul style="list-style-type: none"> • El estudiante debe ingresar sus datos • Ambiente iluminado • Posición de forma frontal
Propósito	Probar el reconocimiento facial de estudiante
Escenario	Haciendo uso de la cámara web se realiza el reconocimiento facial de la estudiante Doris Galindo

Reconocimiento facial de estudiante Doris Galindo.
 Resultado esperado la imagen capturada se almacena en directorio DataSet con el nombre de (Doris Galindo).

Actividades:

01. El usuario elige el menú “Realizar reconocimiento facial”
02. El usuario debe completar los campos “nombre” y “apellido”. Al presionar el botón de reconocimiento facial.
03. Muestra un mensaje de “El reconocimiento empezara en segundos”.

PASO 1 **PASO 2**



RESULTADO: Estudiante identificado “ Galindo Doris”

04. Activación de la cámara en tiempo real.
05. Muestra el nombre completo del estudiante en un label.

Tabla 65. Caso de prueba – Registrar asistencia

Caso de Prueba	PA07 – Registrar Asistencia
Responsable	Sandy Huaranga
Precondiciones	<ul style="list-style-type: none"> • El estudiante debe ingresar sus datos • Ambiente iluminado • Posición de forma frontal
Propósito	Probar el reconocimiento facial de estudiante
Escenario	Haciendo uso de la cámara web se realiza el reconocimiento facial de la estudiante Gladys Samaniego

Resultado esperado al momento de realizar la identificación del estudiante mediante el reconocimiento facial se registra la asistencia de forma automática en el sistema

Actividades:

01. El usuario elige el menú "Realizar reconocimiento facial"
02. El usuario debe completar los campos "nombre" y "apellido". Al presionar el botón de

PASO 1



RESULTADO: Asistencia de estudiante registrado en la base de datos

reconocimiento facial.

03. Muestra un mensaje de "El reconocimiento empezara en segundos".
04. Activación de la cámara en tiempo real.
05. Muestra el nombre completo del estudiante en un label.
06. Se registra la asistencia al momento de realizar el reconocimiento.

6.2.11. Aplicación de la matriz de confusión

Para evaluar el modelo de reconocimiento facial vamos a aplicar las métricas que se muestran en la figura 74, los cuales son: exactitud ("Accuracy"), Sensibilidad y Especificidad (Recall) y Precisión.

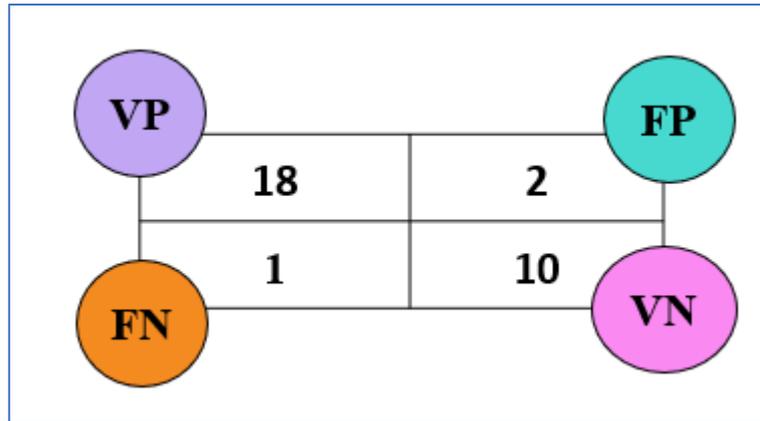


Figura 76. Valores de la matriz de confusión
Tomada y adaptada de TechTarget (33)

La figura 75 y figura 76, muestran los valores que se obtienen en la matriz a partir de los valores reales y predichos por el modelo. Ahora se hallan las métricas utilizando las fórmulas que se muestran en la figura 77.

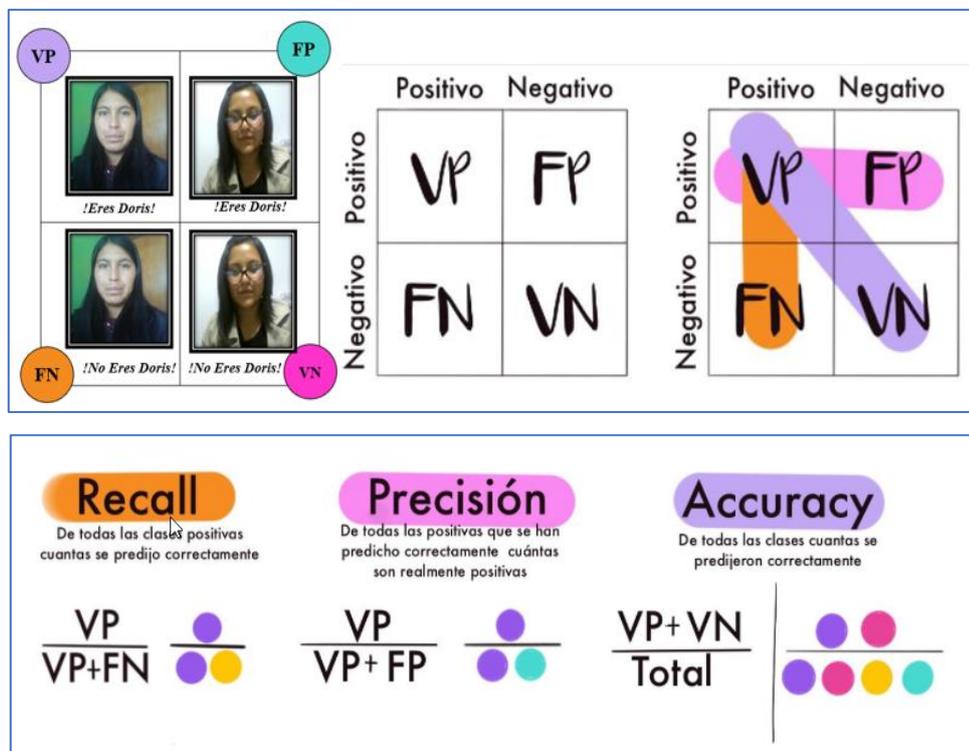


Figura 77. Formulas para calcular las métricas de la matriz de confusión
Tomada y adaptada de TechTarget (33)

- Exactitud (Accuracy): De todas las clases cuantas se predijeron correctamente.

$$Accuracy = \frac{VP + VN}{total} = \frac{18 + 10}{18 + 2 + 10 + 1} = \frac{28}{31} = 0.903 = 90\%$$

- Precisión: de todos los positivos que se han predicho con corrección cuantas son realmente positivas.

$$Precisión = \frac{VP}{VP + FP} = \frac{18}{18 + 2} = \frac{18}{20} = 0.90 = 90\%$$

- Recall: De todas las clases positivas cuantas se predijo correctamente.

$$Recall = \frac{VP}{VP + FN} = \frac{18}{18 + 1} = \frac{18}{19} = 0.94 = 94\%$$

6.3. Discusión de resultados

Como objetivo se consideró identificar la suplantación de identidad de los estudiantes en los exámenes finales de la modalidad presencial en la universidad Continental de la ciudad de la Huancayo mediante el reconocimiento facial. Para desarrollar el reconocimiento facial se consideró 4 pasos, encontrar el rostro a identificar, los puntos referencia, las mediciones y finalmente reconocer el rostro objetivo y lo cual se trabajó con el algoritmo *Face-recognition* que ayuda a reconocer el rostro de una persona.

De los resultados obtenidos en la investigación, se puede deducir que aplicando el reconocimiento facial con el algoritmo *Face-recognition* se puede reducir la suplantación de identidad de estudiantes en los exámenes finales, esto concuerda con los resultados obtenidos en el antecedente internacional de la investigación “*Desarrollo de un sistema de control de acceso de personal empleando el reconocimiento facial con técnicas de aprendizaje profundo*”, (4) ya que con la ayuda de 2 algoritmos MTCNN, modelo pre entrenado Facenet para la precisión y el algoritmo de la vida (anti-plagio) para detectar la imitación del rostro y evitar la vulnerabilidad al momento reconocer la fotografía, se demuestra que el reconocimiento facial arroja un rendimiento elevado en cuanto a la identificación facial con una cercanía al 100% cargada con tan solo 2 fotografías por cada usuario registrado.

A partir de los hallazgos encontrados con respecto al primer objetivo específico aceptamos el proceso de clasificación en *machine learning* y el reconocimiento de imágenes, donde se utilizó la biblioteca de *OpenCV* y la librería de *Face-recognition* que está construido con técnicas de aprendizaje

profundo para el reconocimiento de los rostros y para la extracción de características se utilizó una red neuronal convolucional profunda con una precisión de 99,38%. Estos resultados guardan relación con el antecedente internacional de la investigación “*Reconocimiento facial basado en redes neuronales convolucionales*” quienes señalan de como implementaron su sistema de reconocimiento facial haciendo uso de las redes neuronales artificiales y como realizaron su evaluación para su respectivo funcionamiento, ello es acorde con lo que en este estudio se halla.

Pero, en lo que no concuerda el estudio del antecedente referido con el presente es que no usamos algunas librerías como el *Tensor Flow* y *Keras*. Donde su modelo consta de la primera convolución con una matriz de 94 x 94 y como resultados obtienen el modelo *FaceNet*. Y en nuestra investigación se implementó el algoritmo dentro de un sistema de escritorio desarrollado utilizando el lenguaje de programación C# a diferencia de otras investigaciones anteriores presentadas en el antecedente nacional (46) que solo probaron el algoritmo mas no integraron este modelo a ningún otro sistema.

Por otro lado, la investigación también consideró evaluar el modelo de reconocimiento facial mediante la aplicación de la matriz de confusión. La intención de evaluar el modelo a través de la matriz de confusión utilizando las métricas como exactitud (“accuracy”), sensibilidad (“recall”) y precisión (“precisión”), en la identificación de los alumnos en los exámenes finales de la Universidad Continental de la ciudad de Huancayo. El algoritmo evaluado es el de EigenFace, donde se obtuvo un porcentaje de exactitud de 0.9 (90 %) de igual manera en la precisión un 0.9 con respecto a la especificidad se obtuvo un 0.94. Esto resultados tienen concordancia con lo hallado en el antecedente internacional “*Evaluación de algoritmos aplicados a la extracción de características para el reconocimiento facial en base a la ISO/IEC 25010*” en donde, (8) obtuvieron un porcentaje muy similar de 95% en el algoritmo. Del mismo modo que, en el antecedente nacional “*Redes neuronales artificiales de aprendizaje profundo para el reconocimiento facial y control de acceso de estudiantes a un laboratorio*” (9) los resultados que obtuvieron de los experimentos para las 70 pruebas realizadas demuestran que el sistema de

reconocimiento facial arroja un elevado rendimiento en cuanto a identificación facial se refiere, con valores aproximados al 97 % y además esto se logró con tan solo 2 fotos por cada usuario registrado. También en otro antecedente Internacional “*Sistema de reconocimiento facial con redes neuronales para la toma de asistencia en aulas de clase*” (7) obtuvieron resultados positivos al hacer uso del modelo para detectar a los estudiantes y al profesor, el sistema detectó a los 9 estudiantes y al profesor, todos con una precisión superior al 89 %.

Con respecto a la funcionalidad se usaron pruebas de eficiencia, donde se establecieron diversos escenarios presentando oclusiones como lentes y mascarillas, donde se obtuvo mayor precisión de 0.89 sin la presencia de oclusiones, siendo muy semejante al que obtuvieron (46) en el antecedente nacional en su artículo científico “*Aplicación del deep learning para el reconocimiento facial con la presencia de oclusiones en el contexto de la pandemia COVID 2021*” estos autores obtuvieron un porcentaje de 82.5 %. Así mismo la precisión de reconocimiento con lentes tiene un 78 %, y con la presencia de mascarilla un 67 %, mientras que (46) obtuvieron una precisión de 71 %, siendo estos resultados muy similares al que obtuvimos en nuestra investigación.

CONCLUSIONES

1. En esta tesis se desarrolló el sistema de escritorio para el reconocimiento facial en la identificación de los alumnos en los exámenes finales de la universidad Continental de la ciudad de Huancayo con la finalidad de reducir la suplantación de identidad. Este sistema se desarrolló utilizando la metodología Kanban que nos ayudó a gestionar el equipo y poder visualizar el flujo del trabajo y la carga de trabajo.
2. Para el entrenamiento del modelo se aplicó el proceso típico de clasificación en máquinas de aprendizaje (*machine learning*) en identificación de los alumnos en los exámenes finales de la universidad Continental de la ciudad de Huancayo, que permite poner el funcionamiento del registro y reconocimiento de los estudiantes, con una muestra de 5 personas cada uno con 50 fotografías en diferentes escenarios teniendo un total de 300 fotografías para el entrenamiento del modelo.
3. En el reconocimiento facial se utilizó la librería *Face recognition* y para evaluar este modelo se utilizó la matriz de confusión con dos personas donde se le da una etiqueta a cada uno de estos, finalmente se obtuvo un 90 % de exactitud y precisión y un 93 % de sensibilidad.
4. De acuerdo con el análisis en base a los resultados con la aplicación de la matriz de confusión, se concluyó que la elaboración del sistema de reconocimiento facial es exacto y preciso con un porcentaje del 93 % con el reconocimiento facial.

RECOMENDACIONES

1. Para trabajo futuros se recomienda actualmente trabajar en el desarrollo del proyecto de *software* ya sea de escritorio y *web* utilizar metodologías ágiles como Scrum o Kanban. Ya que estas metodologías nos ayudan a gestionar de manera mucho más eficientes el proyecto y agilizar los trabajos.
2. Se sugiere realizar más pruebas con una mayor cantidad estudiantes para tener una mayor precisión en el reconocimiento facial, ya que las pruebas que se hizo fueron con una muestra pequeña de 5 estudiantes.
3. Se sugiere a la universidad hacer una evaluación de cuanto se puede reducir la suplantación de identidad implementando el sistema de escritorio del reconocimiento facial.
4. Se recomienda realizar una comparación del algoritmo *EigenFace* con otros algoritmos de reconocimiento facial, para identificar cuál de ellos es más exacto y preciso.

REFERENCIAS BIBLIOGRÁFICAS

1. VALORA ANALITIK. Valora Analitik. [En línea] 20 de Agosto de 2021. <https://www.valoraanalitik.com/2021/08/20/colombia-cuarto-en-america-mas-amenazado-por-robo-de-identidad/#:~:text=El%20robo%20de%20identidad%20se,en%20todo%20el%20continente%20americano..>
2. PERU: Ciberdelitos en el Perú: se elevan denuncias de fraude informático y suplantación de identidad. [en línea]. Diario Oficial El Peruano, Lima 02 de julio 2021. (En sección: Ciencia y Tecnología) [fecha de consulta: 18 de enero de 2022] <https://elperuano.pe/noticia/121876-ciberdelitos-en-el-peru-se-elevan-denuncias-de-fraude-informatico-y-suplantacion-de-identidad>
3. ROCHE, Julio. *Kanban vs. Scrum, ¿tendremos un ganador? deloitte*. [En línea] 2021. [Citado el: 12 de enero de 2022.] <https://www2.deloitte.com/es/es/pages/technology/articles/kanban-vs-scrum.html..html>.
4. MUÑOZ, Edison. Desarrollo de un sistema de control de acceso de personal empleando reconocimiento facial respaldado con técnicas de aprendizaje profundo. Tesis (Título de Ingeniero en Electrónica, Automatización y Control). Ecuador: Universidad de las Fuerzas Armadas - Innovación para la Excelencia, 189 pp.
5. ILDEFONSO, Silva. Reconocimiento facial basado en redes neuronales convolucionales. Tesis (Grado en Ingeniería de las Tecnologías de Telecomunicación). España: Universidad de Sevilla, 2018, 77pp.
6. SÁNCHEZ, William, SIGUA, Edisson y VALLADARES, David. Reconocimiento facial en video usando Deep. *Universidad de Cuenca*, , Ecuador. Febrero , 2019 1- 8.
7. JURADO, Miguel y PADILLA, Andrés. Sistema de reconocimiento facial con redes neuronales para la toma de asistencia en aulas de clase. Tesis

- (Título de Ingeniero de Sistemas). Colombia: Universidad Autónoma de Bucaramanga, 2018, 67pp.
8. SÁNCHEZ, Menghy. Evaluación de algoritmos aplicados a la extracción de características para el reconocimiento facial en base a la ISO/IEC 25010. facial con redes neuronales para la toma de asistencia en aulas de clase. Tesis (Título de Ingeniero de Sistemas y Computación). Ecuador: Pontificia Universidad Católica del Ecuador, 2021,82pp.
 9. CAYLLAHUA, Nestor y SUÁREZ, Juan. Redes neuronales de aprendizaje profundo para el reconocimiento facial y control de accesos de estudiantes a un laboratorio. Tesis (Título de Ingeniero Electrónico).Lima: Universidad Ricardo Palma, 2019, 68pp.
 10. MUCHA, Elias, y otros. Aplicación del deep learning para el reconocimiento facial con la presencia de oclusiones en el contexto de la pandemia COVID 2021. *Revista ECIPerú*. Julio - 2021, 18(1), 17-24.
 11. PEREIRA, Alexandre, DONADON, Thiago y OLIVEIRA, Fabio. Aplicación de inteligencia artificial para monitorear el uso de mascarillas de protección. *Revista Científica General José María Córdova*. Diciembre 2020, 19(33), 205-222.
 12. VELARDE, Ingrid y YABARRENA, Josue. Sistema de reconocimiento facial para el control de la trata de personas en Perú. Tesis (Título de Ingeniero de Sistemas). Cusco : Universidad Andina del Cusco, 2018.
 13. MEZA, Alain y RAMOS, Maria. Sistema Inteligente Basado en Redes Neuronales para mejorar la identificación de rostros de delincuentes en el distrito de Laredo. Escuela académico - profesional de informática. Tesis (Título de Ingeniero de Sistemas). Trujillo:Universidad Cesar Vallejo, 2018, 113 pp.

14. MARTÍNEZ, Ricard. Tecnología de verificación de identidad y control en exámenes online. [En línea] 30 de Septiembre de 2020. <https://doi.org/10.1344/REYD2020.22.32357>.
15. HERNANDEZ, Daniel. *La suplantación de identidad cibernética en Ecuador*. [en línea] Bogotá : Universidad Externado de Colombia, 2019., 2019 [Fecha consulta : 10 de enero 2022]. <https://bdigital.uexternado.edu.co/handle/001/1822>
16. CÓRDOVA, Norman. UNFV: Policía detiene a 11 falsos postulantes en examen de admisión. [En línea] Andina, 8 de Abril de 2019. [Citado el: 3 de Marzo de 2022.] <https://andina.pe/agencia/noticia-unfv-policia-detiene-a-11-falsos-postulantes-examen-admision-747877.aspx>.
17. ROUHIAINEN, Lasse. *Inteligencia artificial: 101 cosas que debes saber hoy sobre nuestro futuro*. Barcelona : © Editorial Planeta, S.A., 2018, 2018. 978-84-17568-08-5.
18. ESCOLANO, Francisco, y otros. *Inteligencia artificial: modelos, técnicas y áreas de aplicación*. España : International Thomson Spain, 2003. ISBN: 8497321839.
19. BENÍTEZ, Raúl, y otros. *Inteligencia Artificial avanzada*. Barcelona : Editorial UOC, 2014. ISBN: 9788490643211.
20. BODEN, Margaret A. *Inteligencia artificial*. Madrid : Turner Publicaciones, 2016. ISBN: 9788416714223.
21. ARTIFICIAL INTELLIGENCE - Q&A with Sebastian. s.l. : Udacity, YouTube, 13 de junio de 2017.
22. GERSHGORN, Dave. The Quartz guide to artificial intelligence: What is it, why is it important, and should we be afraid? *QUARTZ*. 10 de setiembre de 2017.

23. KDnuggets. Machine Learning Key Terms, Explained. 2017.
24. LLERENA, José, y LA MADRID, Antonio. Guía de estandarización con especificaciones técnicas de las cámaras de video vigilancia, ubicación y posicionamiento para enfrentar la ineficaz identificación facial en la sección de reconocimiento facial digitalizado de la DIRCRI-PNP de Lima Metropolitana. Tesis (Magíster en Gobierno y Políticas Públicas). Lima : Pontificia Universidad Católica del Perú, 2021, 125 pp.
25. EQUIPO BEEDIGITAL. *BeeDIGITAL*. [En línea] 29 de 10 de 2019. <https://www.beedigital.es/tendencias-digitales/historia-y-evolucion-del-reconocimiento-facial/#:~:text=Los%20inicios%20del%20reconocimiento%20facial, trav%C3%A9s%20de%20la%20tabla%20RAND>.
26. AO KASPERSKY LAB. *Kaspersky*. [En línea] AO Kaspersky Lab, 2022. <https://latam.kaspersky.com/resource-center/definitions/what-is-facial-recognition>.
27. IBM CLOUD EDUCATION. IBM. [En línea] 17 de Agosto de 2020. [Citado el: 19 de enero de 2022.] <https://www.ibm.com/pe-es/cloud/learn/neural-networks>.
28. CALVO, Diego. *Red Neuronal Convolutacional CNN*. 2017.
29. SOLUTIONS, DATA. *Deep Learning, redes neuronales y visión artificial*. 2021.
30. CARCHENILLA. *Una introducción práctica al Deep Learning*. [En línea] 27 de junio de 2016. [Citado el: 19 de enero de 2022.] <https://carchenilla.wordpress.com/2016/06/27/80/>

31. NARAG, Jitin. *Facial Recognition: leverage the power of tech in your industry*. [En línea] 26 de noviembre de 2019. [Citado el: 19 de enero de 2022.] <https://www.techaheadcorp.com/blog/facial-recognition/>
32. PEREZ, Manuel. *El Algoritmo: La otra cara de Leviatán (Spanish Edition)*. Kindle Edition. B088TXQS3N : Edición Kindle, 2020. pág. 207.
33. TECHTARGET, Colaborador de. computerweekly. *computerweekly*. [En línea] 01 de enero de 2017. [Citado el: 12 de Febrero de 2022.] <https://www.computerweekly.com/es/definicion/Aprendizaje-automatico-machine-learning>.
34. SOLANO, Gabriela. OMES. [En línea] 21 de ENERO de 2020. [Citado el: 15 de Febrero de 2022.] <https://omes-va.com/deteccion-de-rostros-con-haar-cascades-python-opencv/>.
35. NOTICIAS NEUROMARKETING. [En línea] 15 de noviembre de 2021. [Citado el: 10 de Febrero de 2022.] <https://neuromarketing.la/2016/12/los-analisis-expresion-facial-funcionan/#:~:text=Las%20expresiones%20faciales%20son%20los,relaci%C3%B3n%20directa%20con%20las%20emociones>.
36. TECHVIDVAN. *Real-time Face Recognition with Python & OpenCV - TechVidvan*. 2020.
37. VIOLA, Paul y JONES, Michael. Rapid object detection using a boosted cascade of simple features. *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2001, 1-9.
38. JIMÉNEZ, Fernando. *Formados para formar*. s.l. : Editorial Cultiva Libros S.L., 2014. ISBN: 8416162271.

39. MIRANDA, Miguel. La imagen digital. Gen [online]. 2009, vol.63, 2 [citado 6 d enero 2022], pp.134-136. <http://ve.scielo.org/scielo.php?script=sci_arttext&pid=S0016-35032009000200016&lng=es&nrm=iso>. ISSN 0016-3503.
40. ORACLE. Oracle. [En línea] IA—Inteligencia artificial, 02 de Marzo de 2020. [Citado el: 14 de Febrero de 2022.] <https://www.oracle.com/mx/artificial-intelligence/what-is-ai/>.
41. RUSSELL, Stuart J. y Norvig, Peter. *Inteligencia Artificial: Un enfoque Moderno*. Madrid : Pearson Educacion, 2004. pág. 1240. Segunda Edición. 84-205-4003-X.
42. MEJDA, Chihaoul, y otros. *A Survey of 2D Face Recognition Techniques*. s.l. : Yevgeniya Kovalchuk, 2016.
43. ESQUEDA, José Jaime y Palafox Maestre, Luis Enrique. *No hay ningún eBook disponible*. California : UABC. 9707350164.
44. MUJICA, Iván, TORIBIO, Luz y CÓNDOR, Daniel F. *Inteligencia artificial como apoyo a intervenciones no farmacológicas para combatir la COVID-19*. *Revista Peruana de Medicina Experimental y Salud Pública*.2020, 37. ISSN: 1726-4642/1726-4634.
45. KASPERSKY. *Kaspersky*. [En línea] 13 de 02 de 2020. [Citado el: 19 de febrero de 2022.] <https://latam.kaspersky.com/resource-center/definitions/what-is-facial-recognition>. 03.
46. ELÍAS, Jemima, y otros. Aplicación del deep learning para el reconocimiento facial con la presencia de oclusiones en el contexto de la pandemia covid 2021. *Revista ECIPerú*.Julio-2021. 8(1),1-8. <https://doi.org/10.33017/RevECIPeru2021.0002/>.

47. XERIDIA. [En línea] 01 de septiembre de 2019. [Citado el: 12 de diciembre de 2021.]
48. CODE, Drastic. *Drastic Code*. [En línea] 2018. <https://drasticcode.com/face-recognition-opencv-or-tensorflow/>.
49. Python: Tensorflow on Windows 10 in PyCharm, 2017 [video de youtube] Melisa, Dale.(10 de abril 2017) [Citado el: 12 de diciembre de 2021.] <https://www.youtube.com/watch?v=83vR1Nz3dHA>
50. TensorFlow. 2018. [En línea]30 de marzo de 2018. [Citado el: 12 de diciembre de 2021.] <https://blog.tensorflow.org/2018/03/highlights-from-tensorflow-developer-summit-2018.html>
51. MORDVINTSEV. Face detection using Haar cascades. *OpenCV-Python Tutorials*. [En línea] 2013. [Citado el: 25 de marzo de 2022.] <https://bit.ly/3bM9aKX>.
52. ROMERO, Gustavo. *Encora*. [En línea] 29 de junio de 2021. [Citado el: 20 de enero de 2022.] <https://www.encora.com/es/blog/como-realizar-pruebas-automatizadas-con-postman>.
53. FERNÁNDEZ, Yamilis, y otros.Pruebas de aceptación para un software con la presencia de una entidad certificadora de la calidad. *Revista Cubana de Ciencias Informáticas* [ed.] Universidad de las Ciencias Informáticas. 3, Ciudad de La Habana, Cuba :, 2007, Redalyc.org, Vol. 1. 1994-1536.
54. ZELADA, Carlos. *Evaluación de modelos de clasificación*. 2017.

ANEXOS

Anexo A

Instrumento de recolección de datos

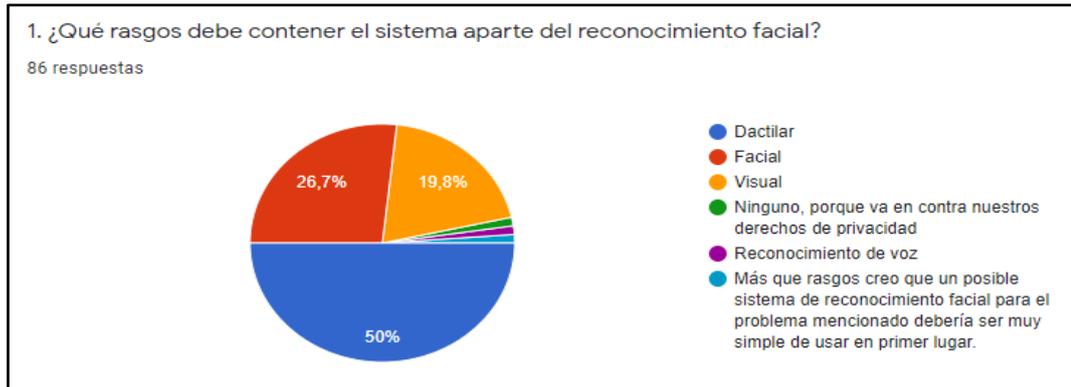


Figura 78. Pregunta Nª1 de la encuesta

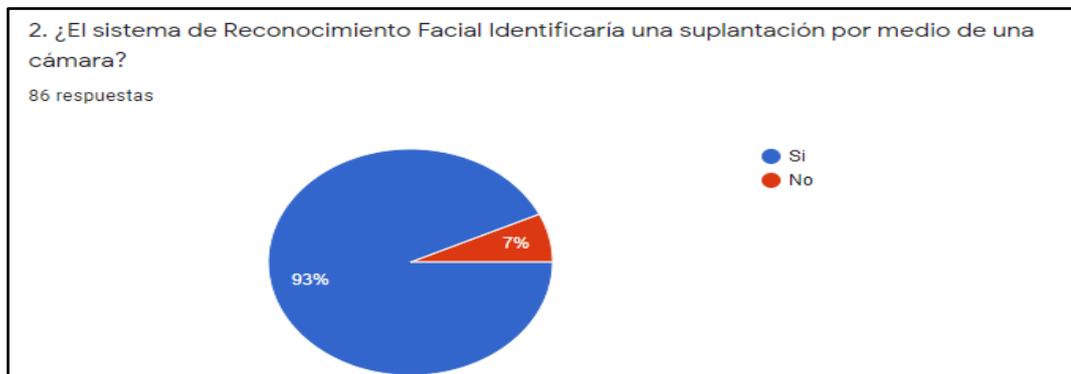


Figura 79. Pregunta Nª2 de la encuesta

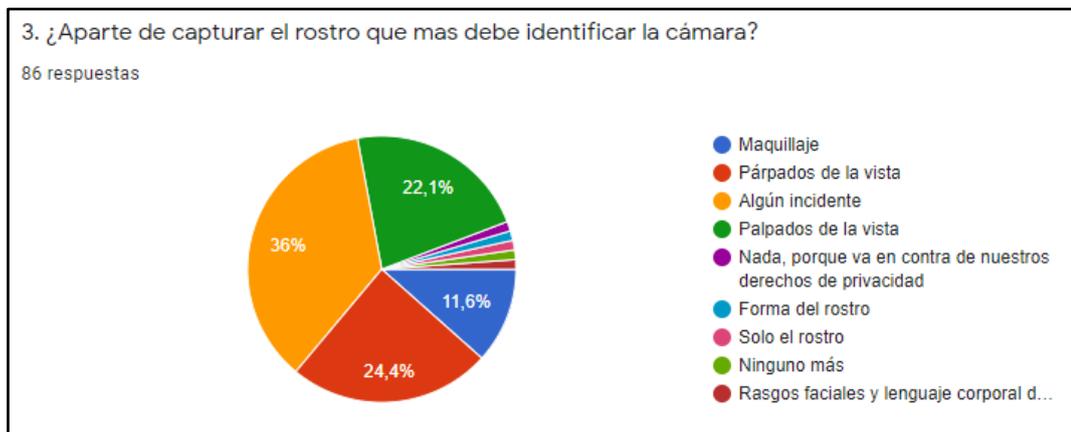


Figura 80. Pregunta Nª3 de la encuesta

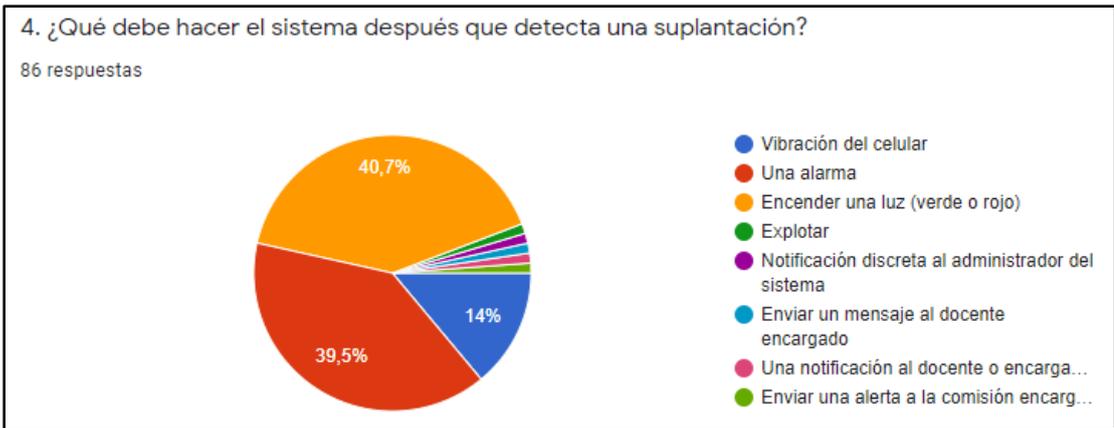


Figura 81. Pregunta Nª4 de la encuesta

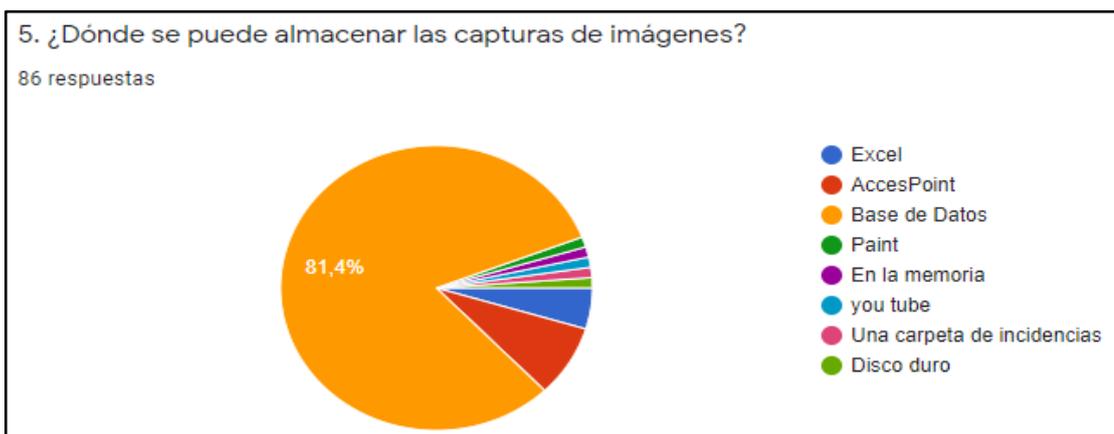


Figura 82. Pregunta Nª5 de la encuesta

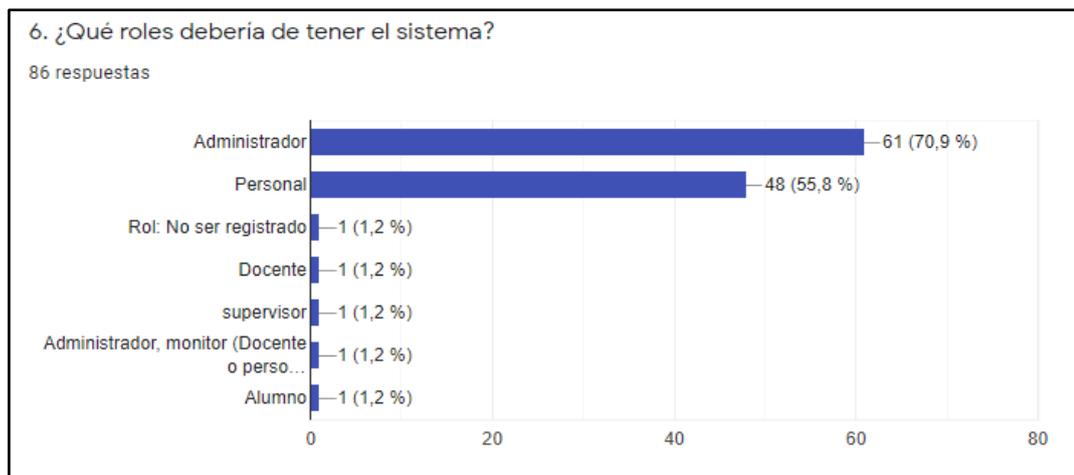


Figura 83. Pregunta Nª6 de la encuesta

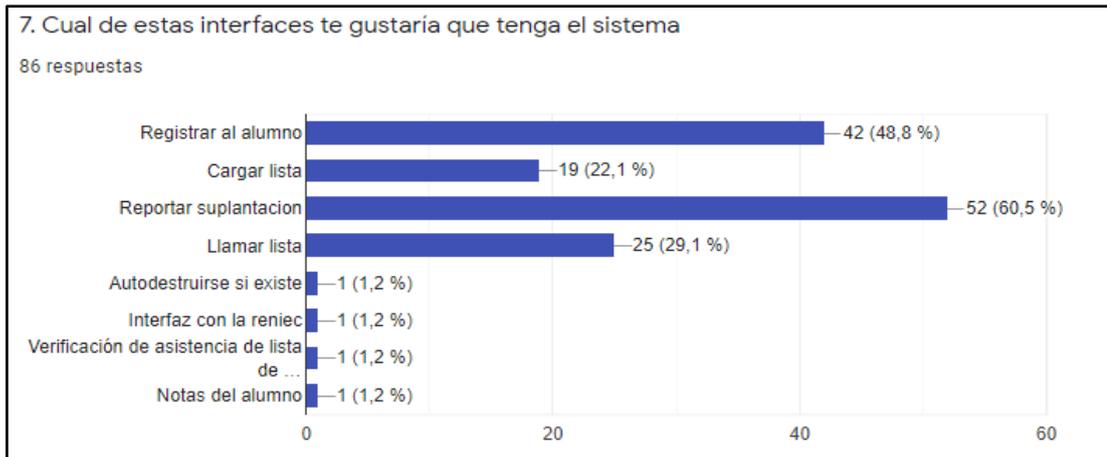


Figura 84. Pregunta Nª7 de la encuesta

8. Sugerencias sobre el sistema
Bueno debería haber una suplantacion de identidad solo al inicio en lanentrada a la u veridocar si el estudiante del carnet es ese mismo,
Aganlo sería de mucha ayuda incluso en la misma universidad
que el sistema sea de fácil uso
Se debe implementar para una examen de admisión
Mas seguro
Que sea solo tengan conocimiento los encargados de la existencia de esta
Esta bien
El sistema debería estar conectado con el celular de la persona encargada para reportar de inmediato
Implementarlo
Que antes de hacer el reconocimiento llamar lista.
Interezante
No se si es necesario tanto control. Pero es necesario q sea efectivo y no muy costoso
Sencillo y facil de usar
El problema que soluciona el sistema no me queda claro del todo, quizás detallarlo un poco más ayude.
Determinar si se hará el reconocimiento mediante fotos, videos en tiempo real u otro. De ello dependerán varios detalles más técnicos y de usabilidad.
Formulario de sorteo de lugares y encargados de los exámenes
Es un buen tema de investigación y aprendizaje, enfocarse en la tecnología disponible (hardware y software) lo más pronto posible para ver viabilidad.
LA verificación facial a través de un aplicativo.
Que permita diferenciar a los estudiantes de los que no lo son.

Figura 85. Pregunta Nª8 de la encuesta

Anexo B

Matriz de consistencia

Problema general	Objetivo general	Variables	Metodología
¿Cómo identificar la suplantación de los estudiantes en los exámenes finales de la modalidad presencial en la Universidad Continental de la ciudad de Huancayo?	Desarrollar el sistema de escritorio para el reconocimiento facial mediante la metodología Kanban para identificar la suplantación de identidad de los alumnos en los exámenes finales de la Universidad Continental de la ciudad de Huancayo.	<u>Independiente.</u> X = Reconocimiento facial para la identificación de alumnos	Kanban Población: Estudiantes de la Universidad Continental Muestra: 5 estudiantes
Problemas específicos	Objetivos específicos	<u>Dependiente.</u> Y = Modelo de reconocimiento facial	Técnicas de recolección de datos – Encuesta por Google Forms Descripción del análisis de datos
¿Cómo elaborar el modelo de reconocimiento facial para la identificación de los alumnos en los exámenes finales de la Universidad Continental de la ciudad de Huancayo?	Aplicar el proceso típico de clasificación en máquinas de aprendizaje (machine learning) en identificación de los alumnos en los exámenes finales de la Universidad Continental de la ciudad de Huancayo.		
¿Cómo evaluar el modelo de reconocimiento facial para la identificación de los alumnos en los exámenes finales de la Universidad Continental de la ciudad de Huancayo?	Evaluar el modelo de reconocimiento facial mediante la aplicación de la matriz de confusión para la identificación de los alumnos en los exámenes finales de la Universidad Continental de la ciudad de Huancayo.		

Anexo C

Código del proyecto

Código de reconocimiento

```
1     if len(faces) > 0:
2         imgS = cv2.resize(img, (0,0),None,0.25,0.25)
3         imgS = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
4
5         faceCutFrame = face_recognition.face_locations(imgS)
6         encodeCutFrame = face_recognition.face_encodings(imgS, faceCutFrame)
7
8         for encodeFace, faceLoc in zip(encodeCutFrame, faceCutFrame):
9             matches = face_recognition.compare_faces(encodeListKnow, encodeFace)
10            faceDis = face_recognition.face_distance(encodeListKnow, encodeFace)
11            matchIndex = np.argmin(faceDis) #valor minimo
12            y1,x2,y2,x1 = faceLoc
13
14            if matches[matchIndex]:
15                detected_count+=1
16                if detected_count == 10:
17                    print(True)
18                    # buscar ID del estudiante
19                    cur = database.cursor()
20
21                    cur.execute(f"select idEstudiante from tblEstudiante where Nombre = '{name_student}' and Apellido= '{lastname_student}'")
22
23                    id_student = cur.fetchone()
24                    print(id_student[0])
25
26                    # Registrar la asistencia
27                    cur.execute(f"INSERT INTO tblAsistenciaEstudiante (IdEstudiante, FechaAsistencia) values ('{id_student[0]}', now())")
28                    database.commit()
29
30                    cv2.rectangle(img, (x1,y1), (x2,y2), (255,255,0),2)
31                    cv2.rectangle(img, (x1,y2), (x2,y2), (255,20,0),cv2.FILLED)
32                    cv2.putText(img,NAME,(x1-20, y2+30),cv2.FONT_HERSHEY_COMPLEX,1,(255,0,0),2)
33
34            else:
35                cv2.putText(img,"Desconocido",(x1-30, y2+30),cv2.FONT_HERSHEY_COMPLEX,1,(250, 0, 255),2)
36                cv2.rectangle(img, (x1,y1), (x2,y2), (255,255,0),2)
37            cv2.imshow("Reconocimiento Facial", img)
38
39            if k == 27: # on [ESC]
40                print("[FINISH]")
41                break
42            cap.release()
43            cv2.destroyAllWindows()
44        else:
45            print("[CAMERA ERROR]")
46
```

```

1  from cv2 import Mat_MAGIC_VAL, cv2
2  import numpy as np
3  import face_recognition
4  import os
5  import sys
6  import MySQLdb as mysql
7  import datetime
8
9  # connect to database
10 database = mysql.connect(
11     host='localhost',
12     user='root',
13     password='root',
14     database='dbreconocimientofacial'
15 )
16 cur = database.cursor()
17
18 name_student = sys.argv[1]
19 lastname_student = sys.argv[2]
20
21 NAME = f'{lastname_student} {name_student}'
22 PATH = f'DataSet/{NAME}'
23 IMAGES = []
24 LIST = os.listdir(PATH)
25
26 casc_path = os.path.dirname(cv2.__file__) + '/data/haarcascade_frontalface_default.xml' # get path cascade from opencv
27 face_cascade = cv2.CascadeClassifier(casc_path)
28 detected_count = 0
29
30 for _file in LIST:
31     curImg = cv2.imread(f'{PATH}/{_file}')
32     IMAGES.append(curImg)
33
34 def findEncodings(images):
35     encodeList = []
36     for img in images:
37         img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
38         if len(face_recognition.face_encodings(img)) != 0:
39             encode = face_recognition.face_encodings(img)[0]
40             encodeList.append(encode)
41
42         print(face_recognition.face_encodings(img))
43     return encodeList
44
45 encodeListKnow = findEncodings(IMAGES)
46 #print('Successfully')
47
48 cap = cv2.VideoCapture(0)
49 cap.set(3, 640)
50 cap.set(4, 480)
51
52 if cap.isOpened():
53     while True:
54         success, img = cap.read()
55         img = cv2.flip(img, 1)
56
57         k = cv2.waitKey(125)
58         gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
59         faces = face_cascade.detectMultiScale(
60             gray,
61             scaleFactor=1.2,
62             minNeighbors=5,
63             minSize=(200, 200),
64             flags=cv2.CASCADE_SCALE_IMAGE
65         )
66

```

```

1 from cv2 import cv2
2 import time
3 import sys
4 import uuid
5 import os
6 import MySQLdb as mysql
7
8 # conexion a la base de datos
9 database = mysql.connect(
10     host='localhost',
11     user='root',
12     password='root',
13     database='dbreconocimientofacial'
14 )
15 cur = database.cursor()
16 DNI_student=sys.argv[1]
17 name_student = sys.argv[2]
18 lastname_student = sys.argv[3]
19
20 cur.execute(f"select idEstudiante from tblEstudiante where DNI = '{DNI_student}'")
21
22 if len(cur.fetchall()) == 0:
23     cur.execute(f"INSERT INTO tblEstudiante (DNI,Nombre,Apellido) VALUES('{DNI_student}', '{name_student}', '{lastname_student}')")
24     database.commit()
25
26 WINDOW_NAME = '.. REGISTRO ..'
27 DIRNAME_STUDENT = f'DataSet/{lastname_student} {name_student}'
28 os.makedirs(DIRNAME_STUDENT, exist_ok=True)
29
30 casc_path = os.path.dirname(cv2.__file__) + '/data/haarcascade_frontalface_default.xml' # get path cascade from opencv
31 face_cascade = cv2.CascadeClassifier(casc_path)
32
33 camera = cv2.VideoCapture(0)
34 camera.set(3, 640)
35 camera.set(4, 480)
36
37 if not camera.isOpened():
38     print('[ERROR] on start camera')
39     sys.exit()
40

```

```

1
2 while True:
3     k = cv2.waitKey(125)
4     success, frame = camera.read()
5     frame = cv2.flip(frame, 1)
6
7     if not success:
8         print('[ERROR] al abrir la camara')
9         break
10
11    cv2.imshow(WINDOW_NAME, frame)
12
13    if k == 27: # on [Esc]
14        print('[EXIT]')
15        break
16
17    if k == 13: # on [Enter]
18        prev = time.time()
19        timer = int(10)
20
21        while timer >= 0:
22            success, frame = camera.read()
23            frame = cv2.flip(frame, 1)
24            cv2.putText(frame, "CAPTURANDO IMAGEN", (100, 400), cv2.FONT_HERSHEY_COMPLEX, 1, (255, 0, 0), 2)
25            if not success:
26                print('[ERROR] al abrir la camara')
27                break
28
29            cv2.putText(frame, str(timer), (50, 50), cv2.FONT_HERSHEY_SIMPLEX, 2, (0, 255, 0), 4, cv2.LINE_AA)
30            cv2.imshow(WINDOW_NAME, frame)
31            cv2.waitKey(125)
32            cur = time.time()
33            gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
34            faces = face_cascade.detectMultiScale(
35                gray,
36                scaleFactor=1.2,
37                minNeighbors=5,
38                minSize=(200, 200),
39                flags=cv2.CASCADE_SCALE_IMAGE
40            )
41            for (x, y, w, h) in faces:
42                cv2.rectangle(frame, (x,y), (x+w, y+h), (0, 255, 0), 2)
43
44            # if cur - prev >= 1 & len(faces) == 1:
45            if cur - prev >= 1:
46                prev = cur
47                timer -= 1
48            else:
49                success, frame = camera.read()
50                frame = cv2.flip(frame, 1)
51                if not success:
52                    print('[ERROR] al abrir la camara')
53                    break
54
55                cv2.imshow(WINDOW_NAME, frame)
56                cv2.waitKey(2000)
57                cv2.imwrite(f'{DIRNAME_STUDENT}/{uuid.uuid4()}.jpg', frame)
58                print('[CAPTURED]')
59            camera.release()
60            cv2.destroyAllWindows()
61

```

```
1 app.post('/api/exec/register', (req, res) => {
2   const {
3     DNI,
4     Nombre,
5     Apellido
6   } = req.body;
7
8   res.send({
9     msg: 'Presiona enter para capturar la imagen'
10  })
11
12  const programPython = executeProgramPython('register', [DNI,Nombre, Apellido])
13
14  programPython.stdout.on('data', (data) => {
15    console.log(data.toString())
16  })
17
18  programPython.on('close', (code) => {
19    console.log(`python program exit with code ${code}`)
20  })
21 })
```